

Survey on P2P File Sharing System

Sean McCarthy

Bijit Hore

Ilya Issenin

Shannon Tauro

Songmei Han

Prepared and Presented by Songmei Han

Outline

- What is P2P?
- Advantages and Disadvantages of P2P System in Comparison with Client-Server Systems
- P2P Application Areas
- Architectures of Existing P2P File Sharing Systems
- File Discovery Mechanisms in Decentralized File Sharing Systems
- Related Issues in P2P Systems.

What is P2P ?

- In a Peer- to-Peer network, end users share resources via direct exchange between computers.
- Information is distributed among the member nodes instead of concentrated at a single server.
- A pure peer to peer system is a distributed system without any centralized control, where the software running at each node is equivalent in functionality.

Advantage and Disadvantage of P2P in Comparison with Client-Server

- Advantages:
 - Inherent Scalability
 - Availability of More Information
- Disadvantages:
 - No Guarantee about Quality of Service

P2P Application Areas

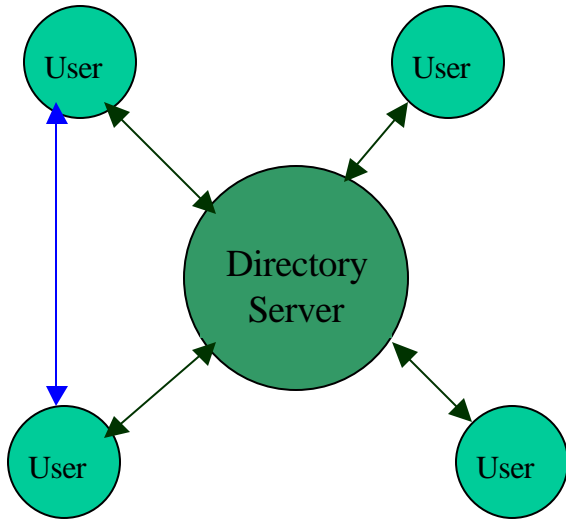
- Communication
 - AOL Instant Messenger, ICQ
- Remote Collaboration
 - Shared File Editing
 - Audio-video Conferencing.
- Distributed Computing
 - [SETI@home](#) [Folding@home](#)
- File Sharing
 - Napster
 - Gnutella, Freenet
 - KazaA, Morpheus

Architecture of Existing P2P File Sharing System (1)

- Hybrid Centralized Peer-to-Peer File Sharing System: Server facilitates the interaction between peers by maintaining directories of the shared files stored on the respective PCs of registered users of the network.
 - Napster

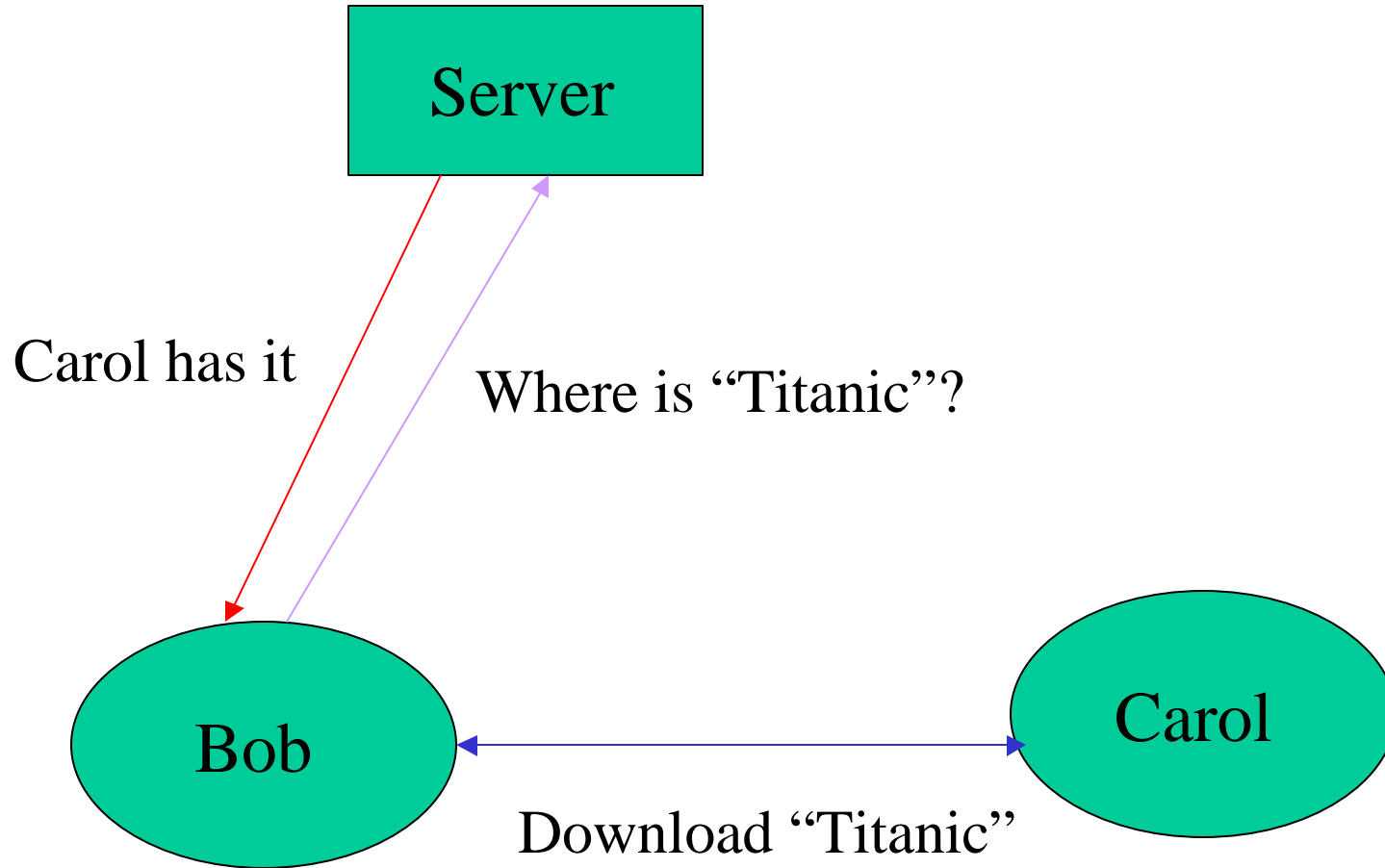
Architecture of Napster

A central directory server maintain index on the metadata of all the files in the network. The metadata might include file names, creation dates, and copyright information . The server also maintain a table of user connection information including user's IP address and line speed. A file query is sent to the server first. A query consists of a list of desired words.



When the server receives a query, it searches for matches in its index. The query results including a list of users who hold the file are sent back to the user who initiated the query. The user then opens a direct connection with the peer that has the requested file for downloading

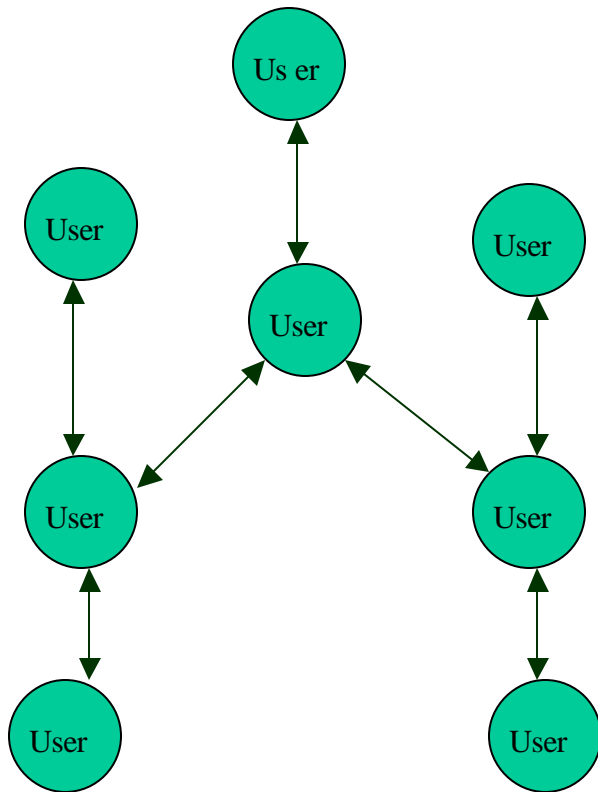
A query in Napster



Architecture of Existing P2P File Sharing System (2)

- Pure Decentralized Peer-to-Peer File Sharing System: Peers have same capability and responsibility. The communication between peers is symmetric. There is no central directory server. Index on the metadata of shared files is stored locally among all peers.
 - Gnutella
 - Freenet
 - FreeServe
 - MojoNation

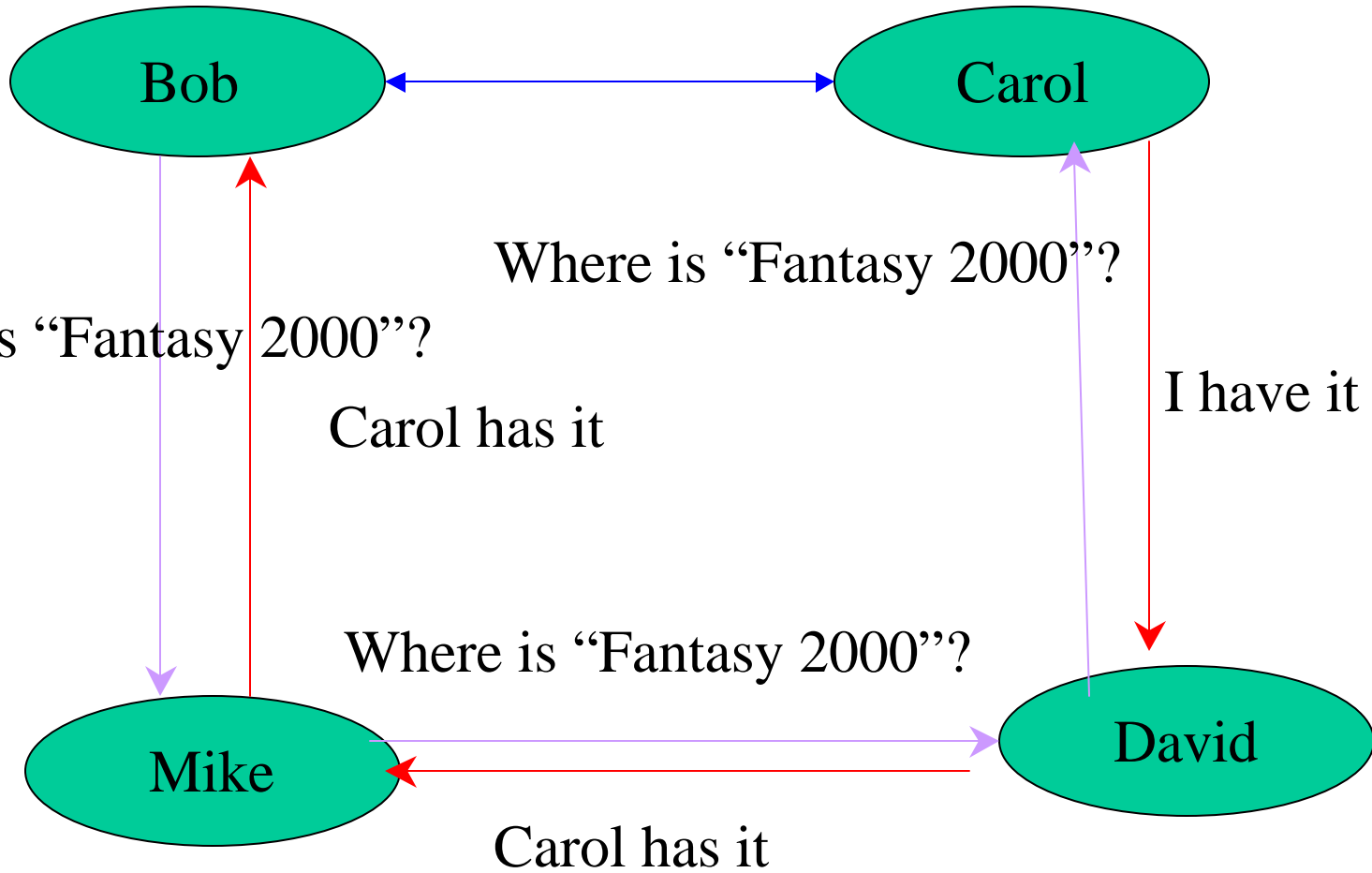
Architecture of Gnutella and Freenet



Each user acts independently.
There is no central directory server.
Peers directly change file information
among each other in a query.

A query in a decentralized file sharing system

Download "Fantasy 2000"



Advantages and Disadvantages of Centralized Indexing

- Advantages:
 - Locates files quickly and efficiently
 - Searches are as comprehensive as possible
 - All users must be registered to be on the network
- Disadvantages:
 - Vulnerable to censorship and technical failure
 - Slashdot effect: popular data become less accessible because of the load of the requests on a central server
 - Central index might be out of data because the central server's database is only refreshed periodically.

Advantages and Disadvantages of Decentralized Indexing

- Advantages:
 - Inherent scalability
 - Avoidance of “single point of litigation” problem
 - Fault Tolerance
- Disadvantages:
 - Slow information discovery
 - More query traffic on the network.

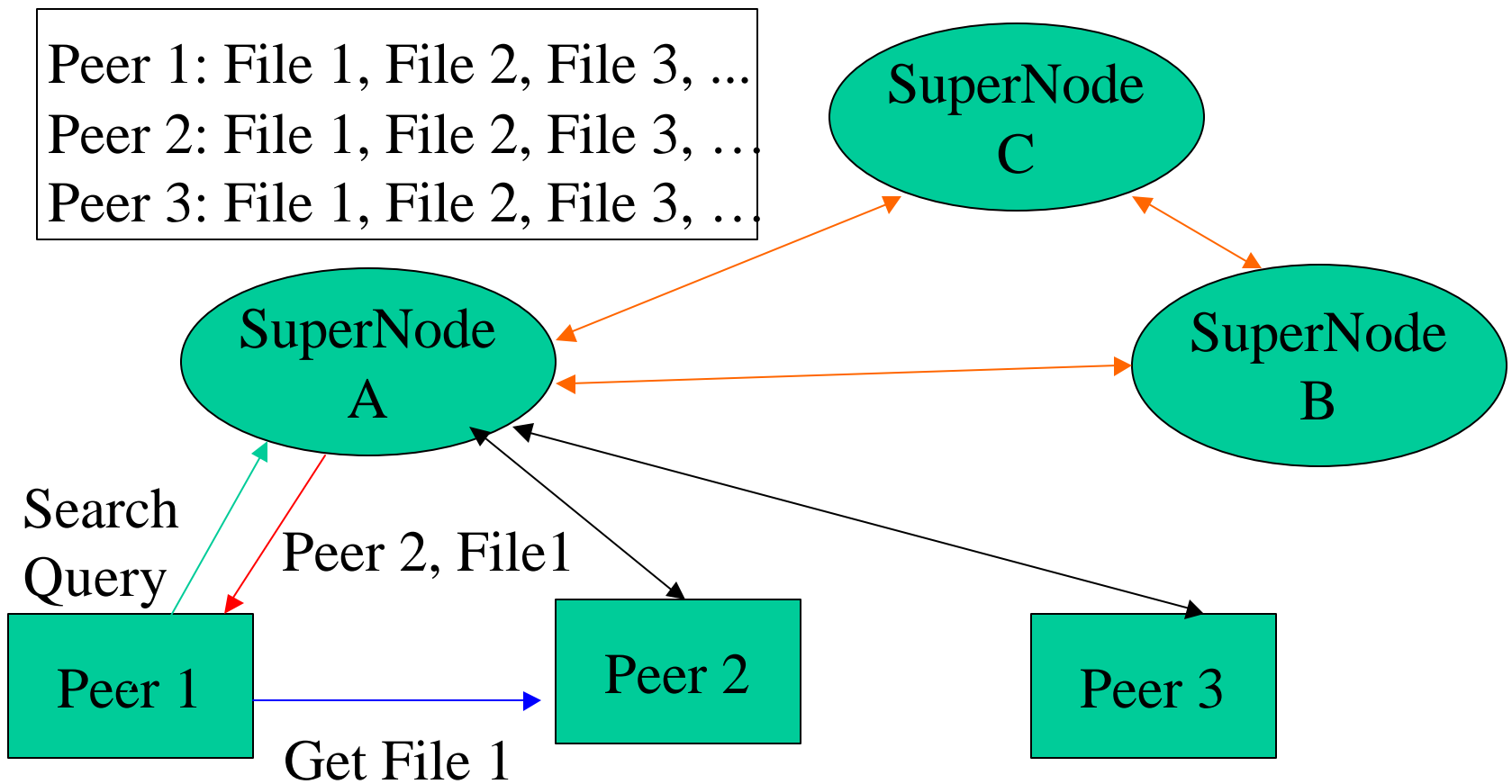
Architecture of Existing P2P File Sharing System (3)

Partially centralized indexing system:

A central server registers the users to the system and facilitates the peer discovery process. After a Morpheus peer is authenticated to the server, the server provides it with the IP address and port (always 1214) of one or more "SuperNodes" to which the peer then connects.

Local "SuperNodes" index the files shared by local peers that connected to it and proxy search requests on behalf of these peers.

- KazaA
- Morpheus



Queries are sent to super node not to other peers. Super nodes process the query received and send the search results back to the requestor directly. Search results in Morpheus contain the IP addresses of peers sharing the files that match the search criteria, and file downloads are purely peer-to-peer.

Morpheus's SuperNode

- Morpheus peers are automatically elected to become SuperNodes if they have sufficient bandwidth and processing power (a configuration parameter allows users to opt out of running their peer in this mode).
- Once a Morpheus peer receives its list of SuperNodes from the central server, little communication with the server is required.
- The selection of SuperNodes in Morpheus has been done using proprietary algorithms and protocols called the FastTrack P2P Stack, licensed from FastTrack. There is no documentation how this has been done.

Proposed Super Node Selection Mechanism in Gnutella Network

- The concept of super node has also been proposed in the new version of Gnutella Protocol. A dynamic super node selection mechanism was proposed to self-organize the Gnutella network into an interconnection of super-nodes and client-nodes (shielded node)[13].
- A slow node comes to the network as a client (shielded node) and tries to connect with a super-node. In the case where no super-node available to accept its connection, it acts as a super-node.

Super Node Selection Mechanism in Gnutella Network (Continued)

- A node with enough CPU power and network bandwidth comes to the network as a super-node and establish super-node connections with other super-nodes. The new super-node is on probation for a preset probation time. It also sets the number of clients (Min-Clients) needed for it to be a super-node in its configuration.
- If the new node receives at least Min-Clients connections from other nodes during the probation time, it remains as a super-node. Otherwise, it becomes a client node and establish client connection to a super-node in its area. If no super-node available, it enters another probation time as a super node.

Improvement over Gnutella

- While Morpheus is largely a decentralized system, the speed of its query engine rivals that of centralized systems like Napster because of its SuperNode.
- SmartStream addresses the issue of incomplete file downloads which can plague systems like Napster and Gnutella.
- FastStream, on the other hand, is intended to solve the other primary issues facing decentralized peer-to-peer file-sharing systems -- slow downloads.

Advantages of Partial Centralized Indexing

- Reducing discovery time in comparison with purely decentralized indexing system such as Gnutella and Freenet
- Reducing the workload on central servers in comparison with fully centralized indexing system such as Napster.

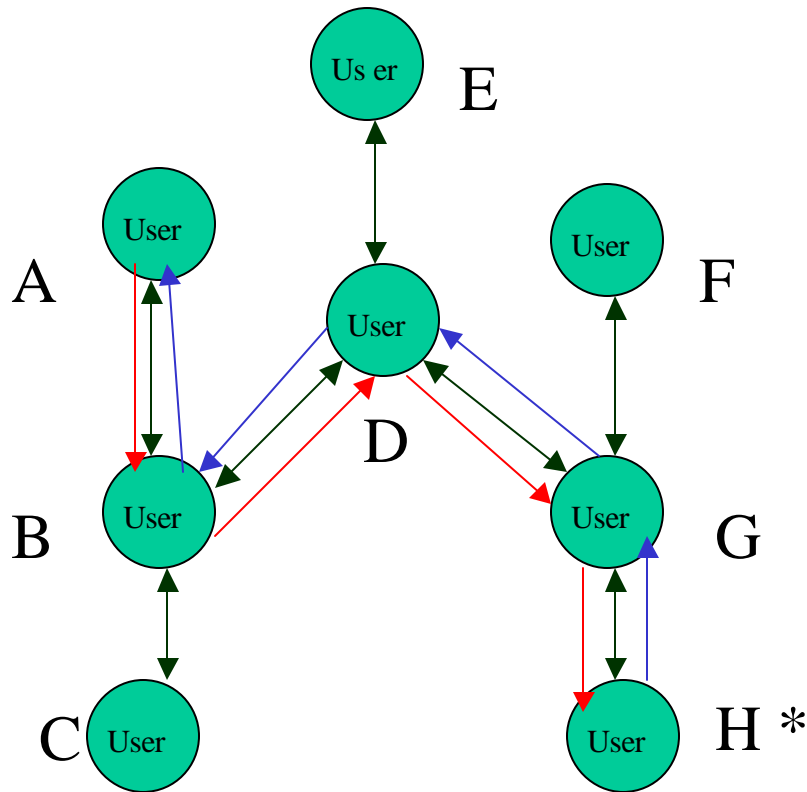
Comparisons between Super-nodes in Morpheus and Servers in Napster

- The central sever in Napster just keeps the index of the files shared in the system. The central server itself does not share any file with peers in the system or download files from other peers. In Morpheus and KazaA, a super node itself is a peer. It shares file with other peers in the system.
- Napster will collapse if the central server goes down. If one or several super nodes goes down, the peers connected to these super nodes can open connection with other super nodes in the system, the network will still function. If all super nodes goes down, the existing peers can become super node themselves.

File Discovery Mechanisms in Freenet: Chain Mode

- A request for a file can be forwarded through many different nodes.
- If a node does not have the document that the requestor is looking for, it forwards the request to one of its neighbor that is more likely to have the document. The messages form a chain as each node forwards the request to the next node.
- Message times out after passing through a certain number of nodes, so that huge chains don't form.
- The chain ends when the message times out or when a node replies with the data.

File Discovery Mechanisms in Freenet: Chain Mode



Computer A send its query to its Neighbor B, which forwards it to one of its neighbor D, which in turn forwards it to one of its neighbor G, finally, the query reach H who has the requested data. The reply is passed back through each node that forwarded the request, back to the original node that started the chain.

Routing Algorithm in Freenet

- Files in Freenet are identified by binary file keys. There are three types of file key: keyword-signed key, signed-subspace key, and content-hash key.
- To search for a file, the user sends a request message specifying the key and a hop-to-live value.
- When a node receives a request, it first looks up its own store and return the data if found, together with a note saying it was the source of the data. If not found, it looks up in its routing table for the nearest key to the key requested and forwards the request to the corresponding node where the data is most likely to be found.
- Nodes store the ID and other information of the requests they have seen to handle data reply message and Request Failed messages

Routing Algorithm in Freenet (continued)

- If the requested data are ultimately found and returned, the node on the pathway will pass the data back to the upstream requestor, cache the file in its own database for future requests, and create a new entry in its routing table associating the actual data source with the requested key for future routing.
- A subsequent request to the same key will be served immediately with the cached data. A request to a “similar” key (determined by lexicographic distance) will be forwarded to the node that provided the data previously.

Routing Algorithm in Freenet (continued)

- To keep the actual data source anonymous, any node along the way can decide to change the reply message to claim itself or another arbitrarily chosen node as the data source. Since the data are cached along the way, the node who claimed to be the data source will actually be able to serve future request to the same data.
- If a node can not forward a request to its preferred downstream node because the target is down or a loop would be created, it will try the second-nearest key, then the third-nearest key, and so on. If a node runs out of candidates to try, it sends a backtracking Request Failed message to its upstream requestor.

Routing Algorithm in Freenet (continued)

- If a node receives a backtracking Request Failed message from its downstream node, it forwards the request to its “next best” node in its routing table. The request inherits the ID and TTL from the backtracking Requested Failed message. If all nodes have been explored in this way, then a Request Failed message is sent back to the node that sent the Data Request.
- If a request’s TTL reaches 0 before the requested data is found, a timeout Request Failed message is generated and sent back to the node that sent the original Data Request to this node.

Data Transfer in Freenet

- When the requested data are found at a node, a Data Reply message that returns the requested data is generated and passed back to the node that initiated the request.
- Nodes only receive Data Reply message corresponding to the Data Request that they have forwarded previously. If a node receive a Data Reply message that it has not forwarded previously, the message is ignored. Data Reply takes the same path as the Data Request.
- Data Reply message is forwarded to the node that the original Data Request came from. Thus, Data Reply will eventually return to the node that initiated the Request.
- There is no direct connection between the requestor and the actual data source.

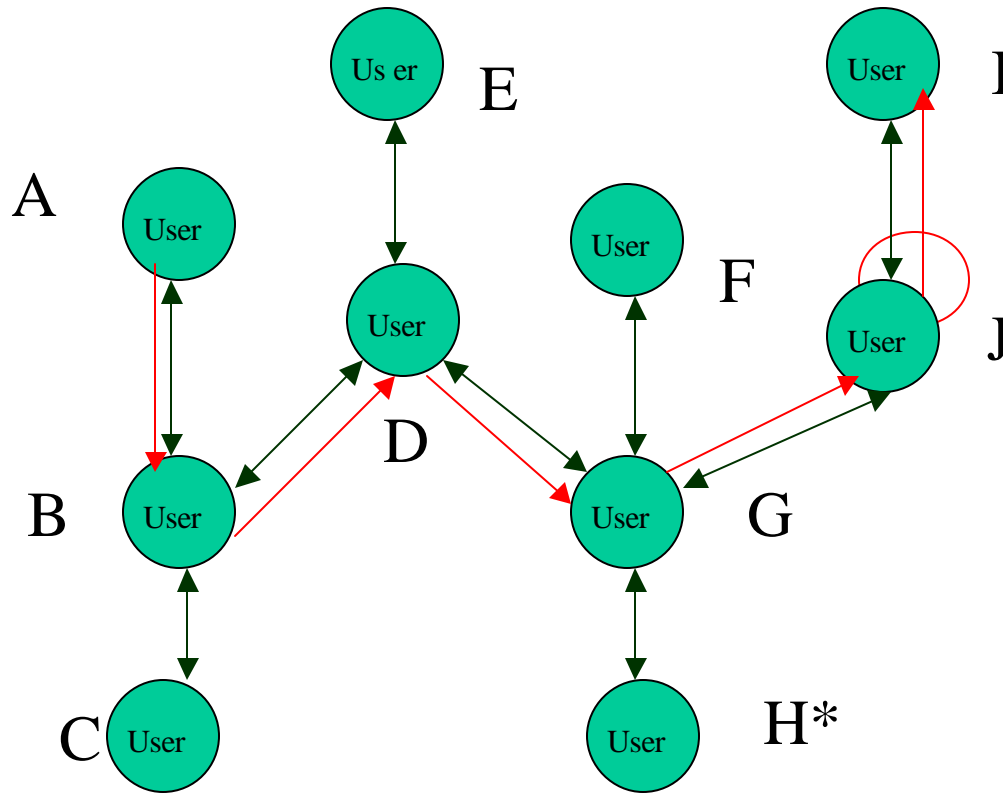
Different Message Type in Freenet

- The following information is included in every message in Freenet:
 - ID, TTL, Source, Destination
- Data Request Message has one more field Key.
- Data Reply Message has one more field Data.
- Data Failed Message has two more fields: Location and Reason.
- Data Insert Message has two more fields: Key and Data.

Advantages and Disadvantages of Chain Mode in Searching

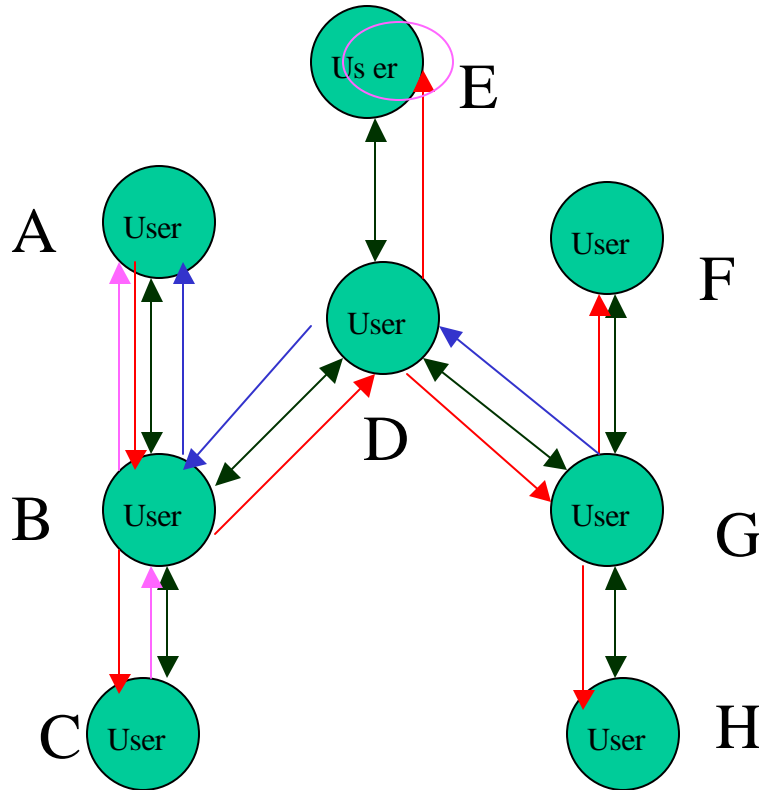
- Advantages:
 - Fast discovery with less network traffic in average case
 - Search stops once the requested file is found
 - More scalable
- Disadvantages:
 - Slow discovery in worst case

Worst Case Scenario in Chain Mode Discovery



Computer H has the requested file, but the query never reaches it because of poor routing decision at computer G. instead, the query ends Up with no file found.

File Discovery Mechanisms in Gnutella: Broadcast Mode



Computer A sends query request to B, which forwards the query to its neighbors C and D, which in turn forward the query to all neighbors. If a computer has a file that matches the request, it transmits the information (file name, size, etc) back through all the computers in the pathway towards A. In this example, both G and C have the requested file and both reply to A.

Routing in Gnutella

- After joining the gnutella network using hosts such as `router.limewire.com` or `gnutellahosts.com`, a node sends out a ping message. A unique ID identifies each ping message. A node that receives a ping message sends a pong message backwards to the originator with the same ID of the ping message and also forwards the ping to its own neighbors.
- Each node keep a routing table for the messages. When a node receives a pong message, it looks up its routing table for the connection that sent the ping message and route the pong backwards to the node that sent the ping message.
- The pong takes the ping's route backwards.

Routing in Gnutella (continued)

- Queries are routed in the same way with pings.
- A node posts a query to its neighbors. When a node sees a query message, it forwards it to its neighbors and also search its own files and sends back a query response to the node that originates the query if a match is found.
- A query response takes the query's route backwards.
- A node forward incoming ping and query message to all of its directly connected neighbors except the one that sent the incoming ping or query.
- If a node receives a same type of message with the ID it saw previously, it will not forward the message to any of its directly connected neighbors to avoid loops.

Routing in Gnutella (continued)

- Each message in Gnutella contains a Descriptor Header with a Descriptor ID uniquely identifying a message. A TTL field in the Descriptor Header specifies how many times the message should be forwarded. A Hops field in the Descriptor Header indicates how many times the message has been forwarded. At any given node i , the TTL and Hops fields must satisfy the following condition:

$$\text{TTL}(0) = \text{TTL}(i) + \text{Hops}(i)$$

where $\text{TTL}(0)$ is the TTL at the node that initiate the message.

- A node decrements a descriptor header's TTL field and increments its Hops field before forwarding it to any node.

Routing in Gnutella (continued)

- Pong may only be sent along the same path that carried the incoming ping. Only those nodes that routed the ping will see the pong in response.
- If a node receives a pong with descriptor ID = n, but has not seen a ping descriptor with the same ID should remove the pong from the network.
- QueryHit message may only be sent along the same path that carried the incoming Query message. If a node receives a QueryHit message with the ID it has not seen previously, it will remove the QueryHit message from the network.

Advantages and Disadvantages of Broadcast Mode in Search

- Advantages:
 - More robust
 - More comprehensive
- Disadvantages:
 - More network traffic and less scalability
 - Possible loops
 - Search continues even after the requested file is found in some nodes.

File Transfer in Gnutella

- Once a node receives a QueryHit message, it may initiate the direct download of one of the files described in Result Set of the QueryHit message.
- Files are downloaded out-of-network in Gnutella. A direct connection between the source and the target node is established in order to perform the data transfer.
- If the data source is behind a firewall, a direct connection to the source may not be possible. File download from a firewalled node is handled by Push Request. Details about the usage of Push Request is explained in next slide.

File Transfer in Gnutella (continued)

- If a direct connect to the data source can not be established, a node can request a file by sending back a Push request to the node that sent the QueryHit message. Upon receiving a Push request, a node attempts to establish a direct TCP/IP connection to the node requesting the file. If this direct connection can not be established, it is likely that the node issuing the Push Request is behind a firewall itself. In this case, file transfer can not be accomplished. If TCP/IP connection is established, the node will send:

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

Where <File Index> and <Servent Identifier> are the corresponding values from the Push Request.

File Transfer in Gnutella (continued)

- Upon receiving the GIV request header, the node will extract the <File Index> and <File Name> fields from the header and construct an HTTP GET request as the following:

GET /get/ <File Index >/ <File Name> /HTTP/1.0\r\n

Connection: Keep-Alive\r\n

Range:byte =0-\r\n

User-Agent:\Gnutella\r\n

\r\n

The remainder of the file download is identical to the file download between non-firewalled nodes.

Messages in Gnutella

- Messages in Gnutella are identified by a descriptor header in the following format

Descriptor ID	Payload descriptor	TTL	Hops	Payload Length		
0	15	16	17	18	19	22

- Ping (0x00): payload descriptor is 0x00, no payload data
- Pong (0x01): payload descriptor is 0x01, payload data contain port, IP address, number of files shared, number of KBs shared.
- Query (0x80): payload descriptor is 0x80, payload data contain minimum speed and search criteria. A node sends QueryHit only if it is able to communicate at the minimum speed specified in the query.

Messages in Gnutella (continued)

- QueryHit (0x81): payload descriptor is 0x81, payload data contain number of hits, port, IP address, speed, Result Set, Servent Identifier.
- Push (0x40): payload descriptor is 0x40, payload data contain Servent Identifier, File Index, IP address, Port.

The Small World Effect

- Stanley Milgram, 1967, Harvard University
- 60 letters sent to randomly chosen people in Nebraska
- Pass the letters to a target person in Boston using only intermediaries known to one another on a first-name basis. Each person passed the letter to a friend whom he/she thought might bring the letter closest to the target; the friend would then pass it on to another friend and so on until the letter reached the target person.
- 42 letters made it through via a median number of just 5.5 intermediaries.

Shortest Path Length

N vertices (people)

- Each vertex has K links to other vertices (relationship)
- In regular graph, where each vertex connects to the nearest K vertices, the path length is $N/2K$.
- In random graph, where each vertex connects to random K vertices, the path length is $\log N / \log K$.

Search Performance of Freenet

- Good average case path length due to random graph property ($\log N / \log K$)
- Poor worst-case path length due to poor local routing decisions.
- Scales logarithmically
- Performance suffers more in targeted attack than in random failure.

Search Performance of Gnutella

- Breadth-first search always finds the optimal path
- Performance is the same under random and target failure
- Scales logarithmically in search path length
- The search bandwidth used by query increases proportionally with the number of the nodes in the network.

Potential Problems with Existing File Sharing Systems

- Spurious content due to its decentralized nature
- Heterogeneous connection qualities
 - Gnutella: 35% have upstream bottleneck bandwidth of at least 100Kbps, only 8% have at least 10Mbps bandwidth, 22% have bandwidth 100kbps or less. [6]
- Free Rider:
 - 66% Gnutella users share no files and nearly 50% of all responses are returned by the top 1% of sharing hosts. [12]
- Copyright infringement, intellectual piracy
- Potential spread of undesirable content

Related Issues in P2P Systems

- Routing
- Security and Reputation
- Trust and Cooperation
- Searching / Indexing
 - Full indexing
 - Partial indexing

Lessons for P2P System Designers

- Take the heterogeneity of the peers into account when delegating the responsibility across nodes overlay;
- Directly measure the characteristics of the peers in the system;
- Encourage server-like peers and discourage client-like peers (free riders) with some resource management mechanisms.

Reference

1. Hurvitz, M. Groove networks: think globally, store locally. *Network Magazine*, vol. 16 (no.5), Miller, Freeman, May 2001. P 82 – 84.
2. Clarke, L.; Sandberg, O.; Wiley, B. Hong, T.W. Freenet: a distributed anonymous information storage and retrieval system. *Design Privacy Enhancing Technologies (Lecture Notes in Computer Science*, vol. 2009), p. 46 – 66. International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, July 2000, Proceedings
3. Fox, G. Peer to Peer Networks.
4. Parameswaran, M.; Susarla, A. & Whinston, A. P2P Networking: an Information-Sharing Alternative.
5. Smart Decentralized Peer-to-Peer Sharing.
<http://www.ohaha.com/design.html>.

Reference (continued)

6. Sarious, S., Gummadi, P.K., Gribble, S.D. A measurement study of peer-to-peer file sharing systems.
7. Kim, H. P2P overview.
8. Yima, The survey of the technologies of peer-to-peer.
9. <http://www.openp2p.com/pub/a/p2p/2001/07/02/morpheus.html>
10. <http://www.openp2p.com/pub/t/74>
11. Modern peer-to-peer file-sharing over the internet.
<http://www.limewire.com/index.jsp/p2p>.
12. http://www.firstmonday.dk/issues/issue5_10/adar/index.html
13. http://groups.yahoo.com/group/the_gdf/files/Supernodes.html