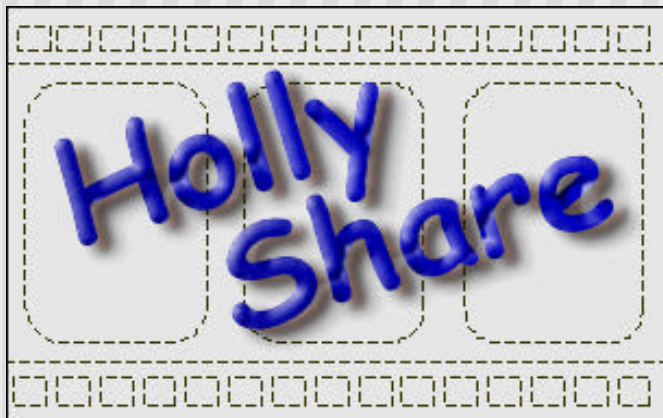


HollyShare: Implementation Details



Project team:

Bijit Hore

Ilya Issenin

Shannon Tauro

Sean McCarthy

Presentation by

Bijit Hore

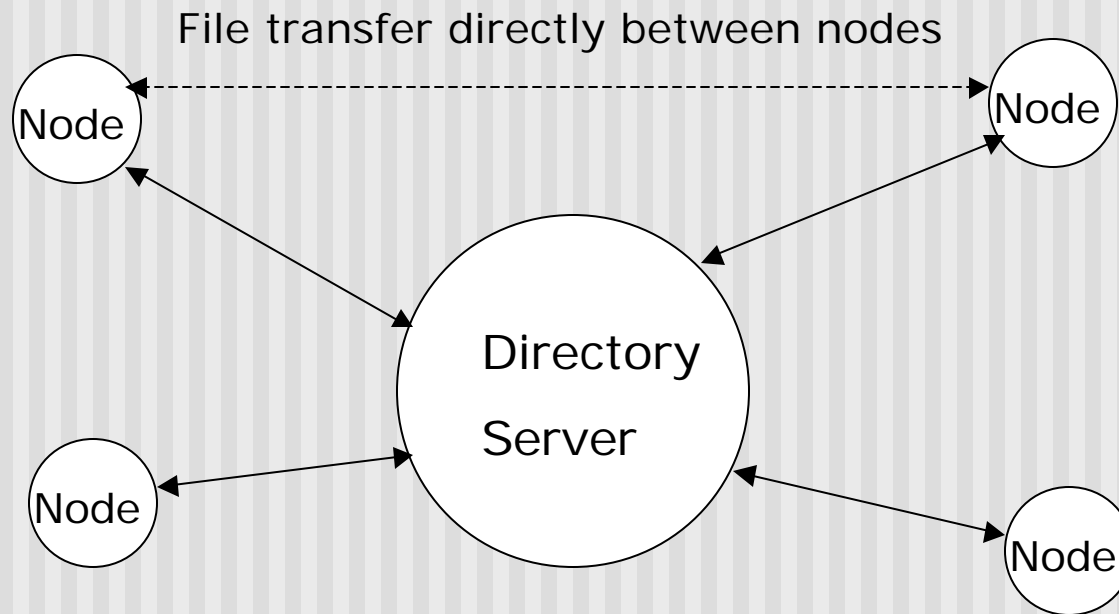
File sharing in Client/Server vs Peer-to-Peer

- Traditional model of file sharing in client-server has files located on one centralized server, and each client gets the needed files from the server
- In the peer-to-peer model, the files are distributed throughout the nodes in the system

File sharing (information discovery)

- In peer-to-peer, there are two main models of finding files:
 - A server system which directs traffic between individual nodes. The server maintains a directory of shared files, and the nodes interact directly with each other when doing transfers (Napster-like system)

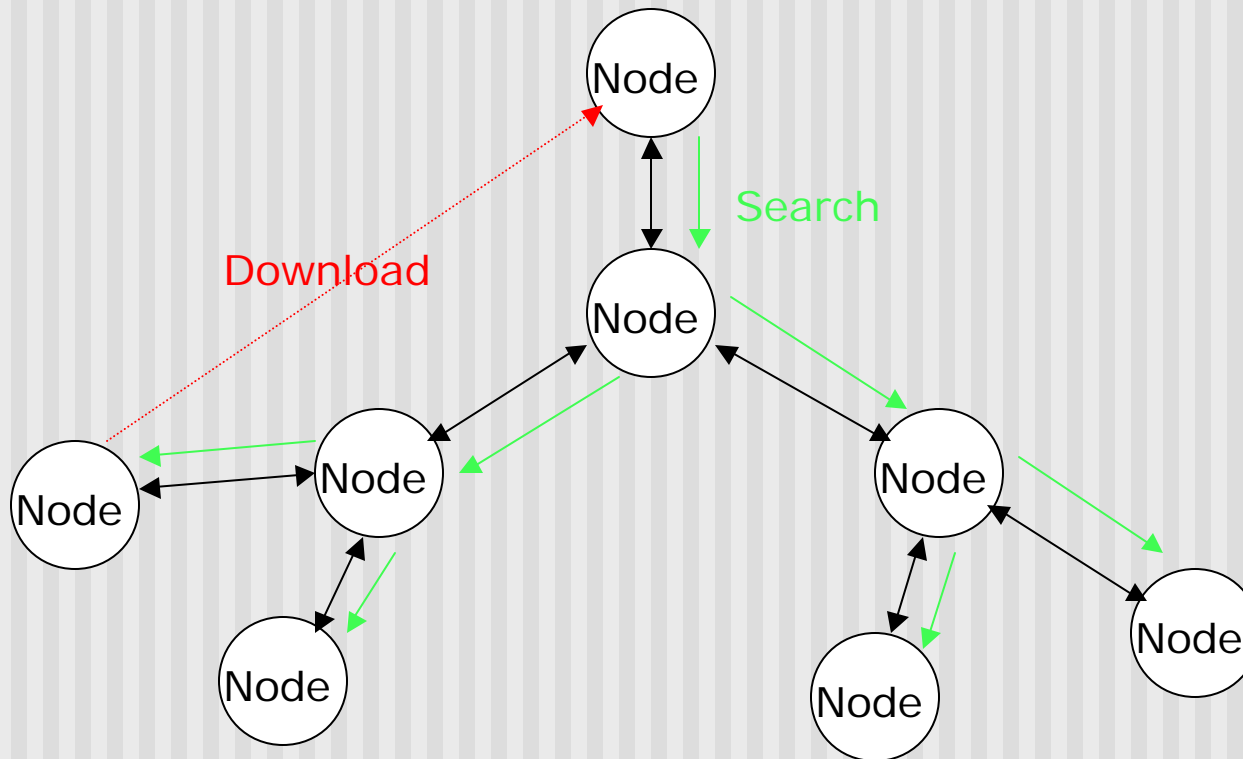
Napster-style architecture



File sharing (information discovery)

- Second p2p file-sharing model
 - Another model for file sharing is the decentralized model. In this model, searches are sent to all nodes in the network. A direct connection between nodes is then opened for a file transfer (e.g. Limewire)

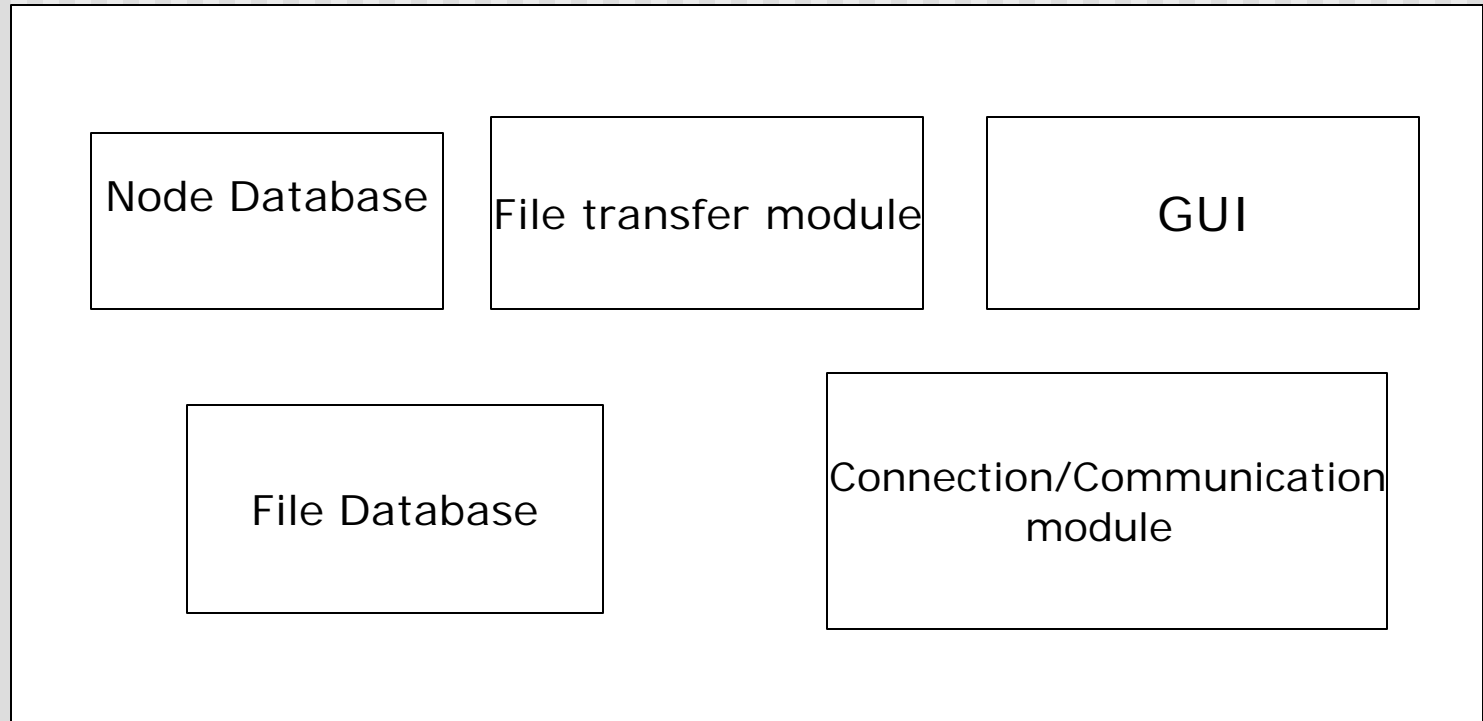
Limewire-style architecture



HollyShare: Network topology

- All nodes are assumed to be identical (running identical programs)
- All nodes have unique IDs (e.g. IP addresses)
- All nodes maintain the list of all IP addresses of the other authorized peer nodes (required for reconnecting – see later)
- Node connections form a tree (no loops)

Components of a node



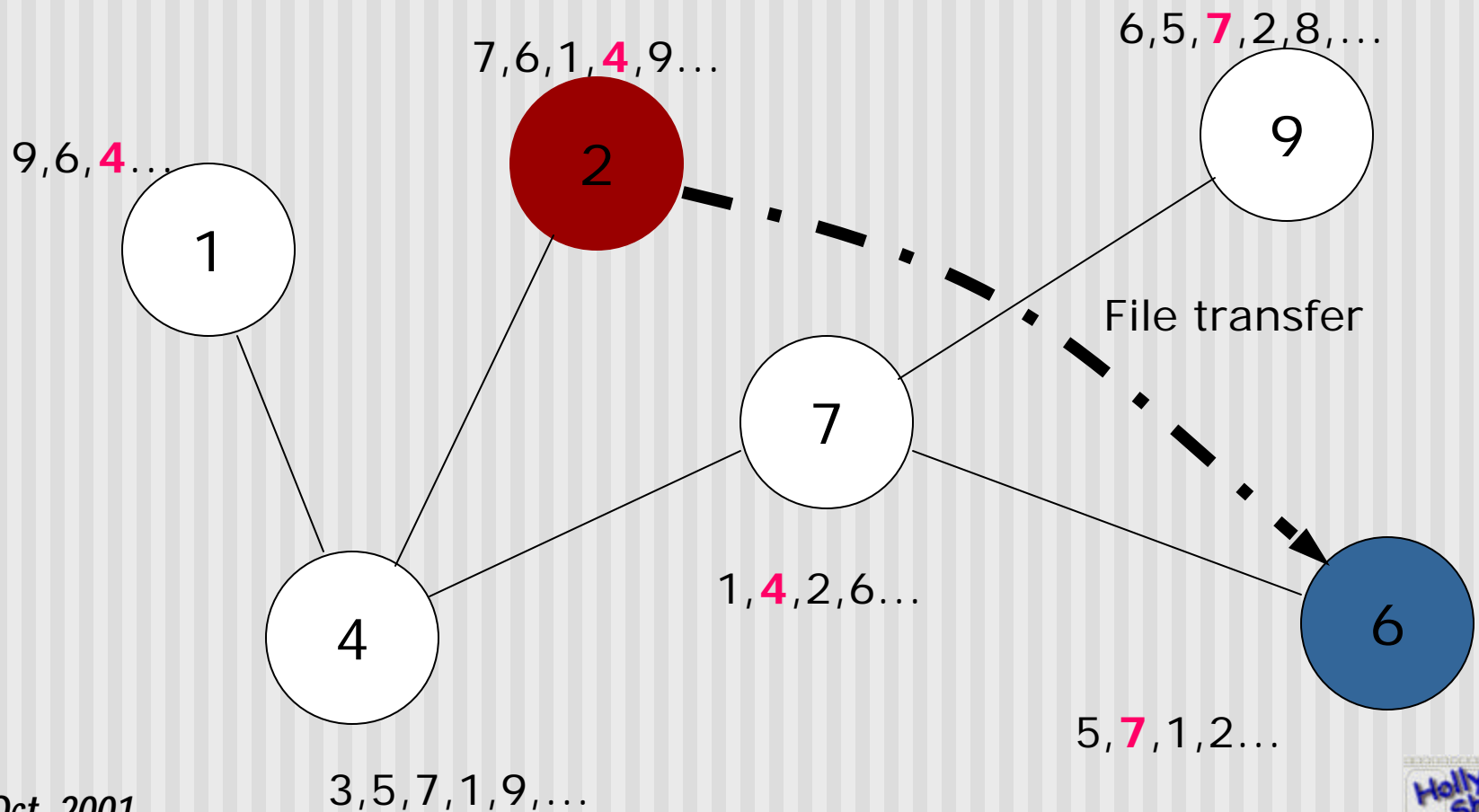
Node architecture

- Node database contains information about all nodes in the system (status, address, etc.)
- File database contains information about all files available in the system (name, location, etc.)
- File transfer module controls connections for file transfers (transfers only)
- Connection/communication module handles all node communication except file transfers (new connections, information propagation)
- GUI is the user interface for the program

Connecting to the network

- First time after installing the program:
 - The program is provided (along with the key) with list of the IP addresses of nodes that may be connected to the network.
- Subsequent connections:
 - During each run, a **random** sequence of IP address is selected and the node shall attempt to make a connection to one of the nodes in that order.
 - This will ensure a **tree** of active nodes and a **small degree** for each node (load balancing).

Connecting to the network



Staying Connected (Robustness)

- If any node disconnects, the network gets disconnected into several **connected components** (property of a tree)
- The neighbors of the disconnected node initiate the process of reconnection:
 - Each connected component determines the highest ID present in this group
 - Each neighbor tries to connect to one of the nodes with the ID, greater than one of its group

Information discovery mechanism

- Whenever new node connects to the network, it sends a list of its shared files to the neighbors, they retransmit it to their neighbors, till each node gets the information.
- All nodes keep the list of all shared files while they are active.
- Notifications of additions or deletions of shared files are propagated from the nodes in a similar manner.
- When one of the nodes disconnects, the list of the files is automatically updated in each node.

Thank you!
