Documentation of a Simple P2P Software

Ang Li (al171) & Wenshun Liu (wl88)

The P2PUDPServer Class:

This server class needs to be launch as a Java Application in order for all p2p clients to connect and obtain a list of available peers. The only public methods are the empty constructor and launchServer(), whose name explain what it does.

This server receives two types of request from any peer that is interested in p2p connections: either connect or disconnect itself from the peer pool. Messages are sent using UDP.

The P2PUDPClient Class:

This class is used by any client that wants to join the P2P peer pool and get information about other peers. The constructor takes 2 arguments: the name one wants to use for identification by other peers, and the IP address of the P2PUDPServer in String. The class has only two other public methods. Method connectAndGetPeerList() connects to the P2PUDPServer and fetches a list of peers (stored in a <String, String> map) with their names (keys), IP addresses, and port numbers (values) to be used for P2P TCP connection. Method disconnect() simply disconnects from the server and removes the calling peer from the pool.

Third-party Usage:

Anyone who's interested in using this P2P software does not have to write a single line of code. They simply need to launch the GUI class "P2PClient" in the "main" package as a Java Application, and click on buttons to connect to the server and subsequently connect to any number of peers available (assuming someone is running the P2PUDPServer Program). All users of this software will be able to talk to each other through a chat window, simply by selecting the desired peer to chat and click "Connect to this Peer". And we support chatting with multiple persons at the same time.

Other minor details:

No ports are occupied and wasted as all sockets are closed immediately when they are no longer needed.

As a specification for the project, disconnecting from the P2PUDPServer is done in the GUI by clicking the "Disconnect from P2P Server" button. Disconnecting from a peer can simply be done by closing the chat window corresponding to that peer.