Assignment #4: P+2 client part

This is the second of two assignments dealing with a P2P application. Protocol P+2 specification has been provided. This assignment will deal with creating the client components of a P+2 peer software. There are two basic client components on a P+2 peer:

1. The user interface will allow the user to query the P2P network to search for certain files. A list of matches will be provided to the user who will pick which file to download.
2. The list of known peers is a key element for keeping this P2P network updated. It is the way of learning about new peers or to know about peers that have left the network (failed connections). Keeping the list of peers is the second task you have to achieve in this assignment.

There may be different ways of achieving these two "client" tasks, but it is recommended each one is performed as a different thread. Of course this assignment is building on your previous work on assignment#3 whose code you are expanding now to create a fully functional P+2 protocol peer.

Your Code

Your code has to conduct the following tasks:

1. The program needs three command line parameters:
    - a port number (P) this peer will use for the client management service
    - a string specifying the name of the shared folder for the peer to use
    - a string in the format IPaddress:port specifying the initial peer address. This third parameter will be used to start populating your list of known peers.
2. User is asked what is the search string. Once provided the program will sequentially connect to each one of the peers contained in the list of peers (not to port mentioned there, which is P, but to P+1) to see whether that peer contains a matching file or not. Each matching file will be printed preceded by a sequence number. At the end of the search, the user will be asked which file to download. The user answer will be the number of the desired file or 0 if no file has to be downloaded.
3. Every ten seconds, the peer list management code will chose the next peer from the list and it will try to connect to it to port P. If connection is unsuccessful, that peer address will be removed from the list of know peers (it likely means this peer is gone and there is no purpose on keeping it on our list of known peers). If connection is successful our client will send its own peer address and it will received an unknown number of lines of text, containing each one a peer address. All of the received peer addresses that are not on the list will be added to it (except when it is your own peer address).

Some tips

- If the string you are looking for is not part of any filename on any peer then you won't get any answer. This is possible and ok. The only choice then for the user will be to respond "0" to the question of which file you want to download.
- Remember that peer addresses are stored on the list using always the main port number P. If for example peer address is 150.1.2.3:1234 and we want to download something from that peer, a connection will be made to address 150.1.2.3 but to port (1236, that is P+2). Or if we want to send a query to that peer then port number will be 1235 (that is P+1). Of course connection will be made to port 1234 (P) for requesting that peer's peer list.
- Your own peer address should not be part of your peer list. Please note that every peer you connect to will add your peer address to its own list. Therefore when you ask for the list your identity will be there. Always double check you don't add yourself to your peer list (or you'll

end up connecting to yourself to ask for files on your shared folder or even worse, trying to download files from yourself).

- Please remember you can have several peers running on your own computer. Just select different port numbers and different shared folders (with distinctive contents) for them and you can test everything even without an Internet connection.

Due date

Your Java source code has to be submitted by email by **April 30th, 2009**. Main class name MUST be `P2P`. Email subject line MUST be Assignment#4 to ensure proper processing of your submission. If you have any doubt about the assignment I suggest you to stop by my office during office hours (posted on the web).

misan@disca.upv.es