Fon + Rin

Data Communications - Project 3 Report

In our BitTorrent project, we have divided the project up into 3 parts consisting of:

- Discovery
- Transmission
- Tracking

How to run

You can run this code by taking .jar file then run on the computers. For the master, enter the filename but for the client, press ‘C’.

ClientSide

This code was taken from Stackoverflow (http://stackoverflow.com/questions/9520911/java-sending-and-receiving-file-byte-over-sockets) and altered to match our requirements.

- Function: ClientSide
    - Downloads the file from the ServerSide
- What it takes:
    - IP of server
    - Port of server
    - Pathfile
- How it does:
    - We open a socket using the given IP and Port and create an input stream as well as an output stream.
    - We check if there is anything to download and if there is then we write it to the pathfile which was assigned in our input stream.
    - When download is done, we close the socket.

ServerSide

This code was taken from the same stackoverflow post as above.

- Function: ServerSide
    - Starts a server and streams the file over to the destination computer
- What it takes:
    - Null
- How it does:
    - First we open a server socket to the selected port. This is the same port that will be used in ClientSide.
    - Then, while the server is open, we create a copy of the file and then put it into the file input stream.
    - Using the buffered input stream, we send out file onto the server

UDPRegister

This code was taken from Michiel De Mey's blog, however altered it a little. (http://michieldemey.be/blog/network-discovery-using-udp-broadcast/)

- Function: UDPRegister
    - Registers the client computer as a computer that needs to be send the file
- What it takes:
    - Null
- How it does:
    - First, we open a socket on any chosen port
    - We then broadcast to the default address which is 255.255.255.255
    - Loop through network interfaces of the computer and get their broadcast addresses
    - Send the UDP packet to the broadcast addresses in the loop and wait for a reply. If we get a reply, the package will be valid. If it is valid, we get the IP address of the reply which will be the server.
    - Close the socket

UDPRegisterM

- Same as above, except this keeps track of the IP address that is the Master node, so that we know all the files will be downloaded from the master.

UDPServ

We got this code from Michiel De Mey's blog (http://michieldemey.be/blog/network-discovery-using-udp-broadcast/)

- Function: UDPServ
    - Listens to UDP requests and checks to see if there are any incoming UDP packets.
- What it takes:
    - Null
- How it does:
    - First, it opens a socket that will listen for incoming UDP requests. Our port keeps changing due to "Connection refused" error.
    - Then, we loop through the UDP requests and also the responses in order to check if the UDP packet that we have received is valid.
    - If it is valid, we will send a response back to the IP and port from which the packet was received.

MakeSeeder

- Function: download()
    - This function will generate JFrame and also make seeder. Seeder is a person who has the complete file. So, when the leecher has done downloading, they will become the seeder.
- What it takes:
    - Torrent file path
    - Original file path
- ProgressBar:

    - To be honest, this code can be found on the internet, so I copy like 90% of the code on the stackoverflow.

Retrieved Address

- Function: getMyIP()
    - This function will find the IP which start with 10.XXXX otherwise it cannot be downloaded. So, it will keep finding the interface.

- We have struggled with this process a lot because in the window and mac are not the same.

Torrent Maker

I got this code from Magno C from StackOverFlow (http://stackoverflow.com/questions/28417276/how-to-use-ttorrent-to-create-a-torrent-file)

- Function: makeTorrent
    - This function will change a normal file to .torrent file. Torrent File contains information about files and a list of the network location of trackers.
- What it takes:
    - File_Location: to tell where the normal file is located
    - TorrentFile_Dir: Where to save the torrent file
    - Master: Own IP
    - Mas: IP Address of the host

- How: Once we have the torrent file, initiate a Tracker object. Then, for each torrent it will announce on this tracker and pass it to the tracker.announce() method.