Data Communication Networks Project 3: LFD — Large File Distributor Report

Group 7 (Thanapat Siripong 5580025, Samart Furuta 5780978)

1. Discovery

- All computers would act as both host and server.
- The host side would open a thread that keep broadcasting computer's local address to others in LAN.
- The server side would open another thread that keep listening to all broadcast including its computer's host and create ConcurrentSkipListSet that keep updating the set of local address of all computers.
- The computer will run both host thread and server thread.
- So at this point, there is no master yet and all running computers are clients.
- When a user type in the original file name and start running a master computer, it will start searching for original file.
- If the master computer finds an original file on its desktop, its host will start broadcasting its local address with letter "M" in front to others.
- All other computers will recognize a computer with address starting from "M" as a master and themselves as clients.
- The master computer will open a thread to set up local server for transferring torrent file to other clients.
- The master computer will create a tracker.
- The master computer then creates torrent file and save it on its desktop.
- The master computer will also announce tracker that new torrent file is created and it has to track this new torrent file.
- The master computer will run local server thread to allow clients to access to get torrent file.
- The clients will get the local address of master and access to the local server of master to get torrent file.

2. Transmission

- The master computer will start running a tracker.
- After getting torrent file, the clients will start downloading.
- The master will also run download program, but the program will check that it has all chunk, so it will not really download.
- All computers including master will share (seed) downloaded chunks.
- All computers will download until they got all the chunks, then combined them to create a downloaded file on desktop of each computer.

3. Progress Tracking
    - Before start downloading, all computers will create a visual progress bar that shows how much they have downloaded so far.

4. Files
    - getIP - It loop through all the address of a computer and return a local address.
    - Handle - a main file of a program. It asks a user to provide a information such as what os is computer, is it master computer or client and automate all other processes for both master and client cases. It also contains a function get_master_ip that will check the set of local addresses whether it contains address with “M” which shows that it is master’s local address.
    - progressBar - It create a progress bar to show how much is the file downloaded.
    - Seed - It start downloading chunks and share (seed) downloaded chunks to other clients.
    - SimpleFileClient - It is a class with method rq() that create a socket that will connect to the local server of master to download a torrent file.
    - SimpleFileServer - It creates a thread that will set up a local server of a master which when run, the server will keep listening and transfer a torrent file when requested.
    - To_torrent - It contains a function makeTorrent that takes in the torrent filename, torrent file path, target filename and path, and local address of master to create a torrent file and save it on provided location. makeTorrent also create a tracker and announce it a torrent file to be tracked. At the end of function, it will start running a tracker.
    - UDPregister - it creates a thread that when run, it will establish a UDP connection and broadcast to all computers in the same LAN a computer’s local address and the string "DISCOVER_FUIFSERVER_REQUEST" as a identifier.
    - UDPregisterM - it does something the same as UDPregister, but the local address it broadcast has “M” in front to show that this address is master’s local address.
    - UDPserv - it creates a thread that when run, it will create a UDP socket that will keep listening to all broadcasts in LAN with identifier string "DISCOVER_FUIFSERVER_REQUEST" and keep updating ConcurrentSkipListSet of local addresses that is running this program. It also contains a function getrekt that will return ConcurrentSkipListSet of local addresses.

5. Flow of a program
    - Program runs main function of Handle.
    - Handle initiates all the variables needed including ConcurrentSkipListSet for storing local addresses.
    - Handle open 3 threads, UDPregister, UDPregisterM, andUDPserv.

- Handle start running UDPregister and UDPserv.
- So a computers that run a program will start communicating with each other and share their local addresses.
- Handle asks the user which os is a computer running on (mac or window).
- Handle then create a path that is suitable for each os.
- Handle will go into a while loop to search for a target file.
    a. If it finds a target file, it will declare this computer as a master and break the loop.
    b. Else if it finds an address with “M” in front in ConcurrentSkipListSet, it will declare a computer as a client and break the loop.

A. If a computer is a master, handle will stop UDPregister thread and start running UDPregisterM thread to broadcast a master local address with “M” in front.
   Then it open new thread of SimpleFileServer to prepare for transferring a torrent file to clients.
   Handle then calls makeTorrent function from To_torrent to create a torrent file and save it on master computer. It also create a tracker and announce it so that torrent file created is tracked. Then tracker is started.
   After that, Handle will call LoadIt function from Seed which will run download of ttorrent. But since master has all chunks of target file, it will not download.
   Seed then let chunks of target files in a computer to be shared (seeded) for other clients. It also calls progressBar to create progress bar showing the progress of download.

B. If a computer is a client, Handle will call get_master_ip function to get local address of a master and sleep for 5 seconds to make sure that master is ready for transferring torrent file and download.
   Handle then creates a SimpleFileClient object and run rq() method to create a socket which will connect to the local server of a master to get a torrent file.
   After that, Handle will call LoadIt function from Seed which will run download of ttorrent. So the client will connect to a tracker to find where the chunk is and download it from there. It also calls progressBar to create progress bar showing the progress of download.
   Seed then shares (seeds) downloaded chunks to other clients. It will keep downloading until all chunks are downloaded, then it combines all chunks to create a downloaded target file.