

Data Communication: Super Distributor Report

By: Worapol Boontanonda (Ham)

Thanaphat Teeradatchusuk (Ice)

Soraya Sakonwittayanon (Sea)

Manual

There are two parts in this program, server and client, both need to be running.

Server:

**** Note:** all clients should be running this program as a client before stating a server.

- 1) Open **“Sea_Ice_DistroBUTER.jar”** file located in size a folder named **“Sea_Ice_DistroBUTER”**
- 2) Choose **“Server”**
- 3) Click **“Search”** button to search all computer that runs as a client.
 - a) All IP Addresses will shows up in the table.
- 4) Click **“Choose”** button and select desired file that will be sent to all the client that shows up in the table.
- 5) Click **“Send”** to starting distribute file.

Client:

- 1) Open **“Sea_Ice_DistroBUTER.jar”** file located in size a folder named **“Sea_Ice_DistroBUTER”**
- 2) Choose **“Client”**
- 3) Type instance name in the text field (instance name can be any name).
- 4) Click **“Send”** button
- 5) Click **“Start”** button to start the service.

Discovery

Overview:

We decided to user “Bonjour” service to discover all clients ip addresses. We used three “dns-sd” commands in total to get all clients ip addresses.

- 1) **“dns-sd --B _http._tcp”**
- 2) **“dns-sd -L “INSTANCE” .”**
- 3) **“dns-sd -G v4 COMPUTER_NAME”**

First, **“dns-sd --B _http._tcp”** commands, this will search all bonjour services that was runs in client's side. Then, we runs **“dns-sd -L INSTANCE”** commands use INSTANCE name that user input in their client’s side to look up their COMPUTER_NAME. Finally, use **“dns-sd -G v4 COMPUTER_NAME”** to look up for client’s ip address. In each commands, we used

“ProcessBuilder” to run those commands and used ArrayList to store all the ip addresses then pass it to torrent part.

Torrent

Overview

Run listen server (HTTP server) on client

Client has 3 HTTP handlers; “/connect”, “/getTorrent”, “/startTorrent”

1. /connect — check if client is alive
2. /getTorrent — send command to the server that the client is ready to receive the torrent file
3. /startTorrent — server sends command to client to start downloading the torrent.

Run main tracker server on Master (HTTP server)

There are 2 HTTP handlers; “/getFileName” and “/downloadTorrent”

1. /getFileName — get filename of the distributed file
2. /downloadTorrent — client download torrent file from server

The Master server act as both a tracker and a seeder. Master track the file from other peer according to the bittorrent protocol. The Master also distribute the file to other peers as the initial seeder of the file

The Master and client communicate through the HTTP protocol using HTTP Url Connection from the java library and the master and client run their own HTTP server. The client run as a listen server to enable the master to talk to the client through HTTP and make an initial acknowledgement connection. (by going CLIENT:port/connect) After doing the ACK, the server send command CLIENT:port/getTorrent to tell the client to do MASTER:port/getFileName and MASTER:port/downloadTorrent to get the distributed file name and get the actual torrent from tracker.

After the client finish receiving the torrent the server send CLIENT:port/startTorrent to initialize the client to start download the torrent file with the receive torrent file from server, the client with act as a peer to help seed the file to other client.

Library used: ttorrent, HTTPServer from sun system, and java standard library such java.io and java.util etc.(ttorrent library pull from maven)