ReadMe for the client and server applications.

Name: Giric Goyal

ID: ggoyal2

UIN: 657440995

(This readme assumes that you are in the project root that is “ApplicationLayer/”)

How to make:

This project uses make file to compile and generate the executables for both client and server.

To clean the project:

At the terminal/command line, type

make clean

To compile the client and the server applications:

At the terminal/command line, type

make

Or you can use:

make all

All object files are stored under ApplicationLayer/build/objects/

All the executables are stored at the project root I.e. ApplicationLayer/

All the source file are under ApplicationLayer/

How to run:

There are two programs/executables:

**Client:**

To run client, at the terminal/command line:

./client [destination-ip] [destination-port] [relay-ip] [relay-port]

Use the “./client “ command to run the client application. You can provide command line arguments while running the client application.

./client 54.207.17.177

If you give just one argument (destination-ip), then the application will use the default destination-port 54321 and no relay as default.

./client 54.207.17.177 54321

If you give two arguments (destination-ip and destination-port), then the application will use no relay as default.

./client 54.207.17.177 54321 54.207.17.177

If you give three arguments (destination-ip, destination-port and relay-ip), then the application will use 54320 as default relay port.

./client 54.207.17.177 54321 54.207.17.177 54320

If you give four arguments (destination-ip, destination-port, relay-ip and relay-port), then the application will use the destination ip and port and relay ip and port that you provide.

If you do not provide any command line arguments, the application will show you instructions and quit.

Be sure that you type only numbers when asked for port numbers.

After you have run the client,

The client application will ask the filename that you want to transfer. You will need to give the name of the file along with the path. The file will be stored with the same file name and relative path on the server.

Enter file name with path : [filename-with-path]

After you give the name, if the application is not able to read the file, it will ask you again for the file name and try again.

Upon successful opening of the file, the application will ask you if you wish to save this file on the server. If you wish to save the file, press ‘y’ or ‘Y’ and hit enter/return.

If you don’t wish to save the file, press ‘n’ or ‘N’ and hit enter/return.

Save File? (y/n) : [y or n]

Then the application will ask you if you wish to send another file after this file.

If you wish to send another file, press ‘y’ or ‘Y’ and hit enter/return.

If you don’t wish to send another file, press ‘n’ or ‘N’ and hit enter/return.

Send more after this file? (y/n) : [y or n]

If you wish to send another file and pressed ‘y’ and enter/return, then after successful transfer of the file, the application will ask you about the new destination-ip, destination-port, relay-ip and relay-port to send another file.

Enter destination IP and Port Number : [destination-ip] [destination-port]

Enter the destination-Ip and destination-port number separated by a white space.

Enter relay IP and Port Number : [relay-ip] [relay-port]

Enter the relay-Ip and relay-port number separated by a white space.

**Server:**

To run server, at the terminal/command line:

./server [port-number]

Use the “./server” command to run the server application. You can provide port number as command line argument.

./server 54321

If you don’t provide any command line arguments, the server application will use the default port number 54321 to connect. You should connect to the same port number when using the client application. I.e. specify the same port number as destination-port number that you use here.

What the application can do?

**Client:**

1. Client can send files of any size to the server.
2. Client will ask you for the destination-IP, destination-port, relay-IP and relay-port.
3. Client will ask you if you wish to save the file to the server.
4. Client will ask you if you wish to send another file after sending the current file and will again ask you the destination IP and port number and relay IP and relay port. If you do not wish to use a relay, enter “none: for relay-IP and any integer for the relay-port.
5. Client can connect to just one server at a time.
6. Client will try to connect and communicate with a relay if you specified a relay. If you don’t wish to use a relay, omit the command line argument or enter “none”.
7. Client will show you the summary of your transaction after you have sent all the files.

Client will tell you the files that you transferred, the total number of bytes transferred (sent by the client and received by the server) and the total time taken.

**Server:**

1. Server can accept just one connection at a time.
2. Server will listen and connect to the port number that you specify while running the server.
3. Server will keep the connection open as long as you wish to send the files from the client.
4. After the server closes the connection, it will quit. You will need to run the server again if you wish to accept more connections.
5. Server can only accept connections directly from the client and not the relays.
6. Server will save the file on the disk, if the save file is set in the header.