ReadMe for the client and server applications. (Homework 3, RDT Go Back N)

Name: Giric Goyal
ID: ggoyal2
UIN: 657440995

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 TransportLayer1/build/objects/
All the executables are stored at the project root I.e. TransportLayer1/
All the source file are under TransportLayer1/

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] [client-port] [garble chance]`

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, no relay as default and 54329 as client port.

`./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 and 54329 as client port.

`./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 and 54329 as client port.

`./client 54.207.17.177 54321 54.207.17.177 54320`

If you give five arguments (destination-ip, destination-port, relay-ip, relay-port and client 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, relay-port and client-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.

`Enter client Port Number : [client-port]`

Enter the client-port number.

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. Divides the file in chunks and creates packets for each chunk.
2. Client will ask you if you wish to save the file to the server.
3. 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.
4. Client will send the packets to the server or the relay.
5. Client will show you the summary of each transaction.
6. Send the packet to the server based on RDT 3.0 Go back N

Server:
1. Server will accept packets form the client.
2. Server always waits to accept packets from the clients.
3. Calculate checksum and send acknowledgement based on RDT 3.0 Go Back N.
4. Server will save the file on the disk, if the save file is set in the header.