Computer Networks

Instructors: Shashi Prabh

Lab 5: A client-server application for file transfer

In this lab, you will gain familiarity with socket programming by modifying the a simple client-server code provided in the textbook. *This lab is to be done individually.*

File transfer over a C socket

Save the given client-server code `client.c` and `server.c`, preferably, in separate "client" and "server" folders. Compile and test the client-server code. For compiling, you can invoke:
`gcc srcfilename -o execfilename`

1. Implement file transfer. Add code to `client.c` and `server.c` so that once the server receives a string `GET`, it sends a specified hardcoded file to the client ("sample.txt" for example).
2. Next, implement command-line switch `-f file_name` such that user can provide filename to the server from terminal.
3. Test the code on a large file. Compare the sent and received files.
4. Do the previous step using a UDP socket. Are the two files the same? Why or why not?

   Normally, checking for the end of file character `EOF` is sufficient. However, it is possible that the datagram containing `EOF` gets lost and as a result, your client will not terminate. You can use timeout (say 10 times the RTT) on the client side. Further options to consider are to have server send the file size before initiating the transfer and to have the server send a flag e.g. `BYE` to indicate the completion of transfer.

Extra Credit Extend the implementation so that the client sends a filename after `GET`. The server then sends this file if the file is found else sends a `File not found!` message.

Submission The files must contain your name, AU ID and email address. Upload the code to the submission server.