ECE 5600 Project 4 (Phase 4)

OBJECTIVE

1. Be familiar with the TCP protocol.
2. Implement a TCP transmission.

BACKGROUND

The Transport Control Protocol (TCP) belongs to the transport layer of the Protocol stack. TCP is a reliable, connection-oriented service.

The TCP services that are applicable to this phase of the project are listening and reading. The listen service sets up a port to receive connections. For the purpose of this project, you may make your port listen whenever it is not connected, so you don't really have to provide any interface to the application layer. The read service allows an application to read the bytes that have been delivered over the connection.

For this project, you will need to set up TCP port 5600 to listen for a connection (i.e. respond to a TCP packet with SYN = 1). After the connection is made, your program will have to receive a stream of bytes and hand them off to the application layer. You will need to write a tiny bit of application layer code to read the data from the port and write it to a file. Do not make this project more complicated than it needs to be.

PRE-LAB READING

Chapter 6. P552~565

PROJECT PROCEDURE

1. Write a server application according to standard Linux socket (including opening the socket, binding, waiting for incoming connection and write the coming message to a text file). The port number is 5600. This program should contain a little bit file operation.

2. Find tcp_test.cpp in the sample codes archive and save it to another computer, read it carefully and you will find that it is a client application. tcp_test.cpp will send some information to the server you implemented above. You may need to change the port number in it to 5600 as well. Compile (use g++ tcp_test.cpp –o tcp_test) and then execute it with server's IP address. For example: ./tcp_test 192.168.1.20

3. You should first execute the server program, and then run the client program.

4. If all goes well, tcp_test will report that the connection was successful and that the data were transferred.

NOTE: In this project, you **do not** have to write a standalone protocol as in lab 2 and 3, the more important part is to observe and verify TCP connection which includes three-way handshake, data transfer and connection closure.

REPORT REQUIREMENTS

1. Copy and paste the transmitted content in your report.

2. Screenshot the command window when you successfully establish the TCP connection. It should display “connect succeeded”.

3. From Wireshark, show the three-way handshake process by highlighting the SEQ and ACK number. Then highlight the frame you transmitted.

4. Attach your codes as an appendix in your lab report.

5. **Project due: Dec 2, 2016**