ECE 4740 Project (Phase 4)

OBJECTIVE

1. 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 read. 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 6600 to listen for a connection (i.e. respond to a TCP packet with SYN = 1). After the connection is made, your protocol stack 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. Once all the data have been transferred, your protocol stack must respond to a disconnect packet (a TCP packet with FIN=1).

Do not make this project more complicated than it needs to be. There's no need to implement piggyback acknowledgements; send a separate acknowledge packet each time you receive a packet. There's no need to use any control bits other than SYN, FIN and ACK. Don't bother with selective repeat. Use a go-back-N strategy, so if you get a packet with the wrong sequence number, return an acknowledgement with the sequence number you expect, but don't buffer the packet.

PRE-LAB READING

Chapter 6. P552~565

PROJECT PROCEDURE

1. Write code to implement the TCP services discussed above. Note that options in the TCP header are common. Your code doesn't need to look at them, but it needs to account for them when computing the starting address of the payload.

2. Go to the course website and download the file tcp_test.cpp. Compile this program and load the executable onto a computer near you.

3. Compile and execute your program. Run tcp_test (on the other computer) and pass it your stack's IP address (command line argument).

4. If all goes well, tcp_test will report that the connection was successful and that the data were transferred. Include the data in the file that your application layer code wrote, your program, and the usual introduction and conclusions in your report, and e-mail the report by midnight on the due date on the website. (Hint, the result should be text that is almost English)

5. If you've done it correctly, you should be able to run tcp_test again without restarting your protocol stack.

How to run udp_echo:
1. Complie the program: g++ tcp_test.cpp –o tcp_test

2. Execute the tcp_test: ./tcp_test 192.168.1.10(dest ip address)

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 the wireshark, highline the frame your transmit and make sure the checksum value is correct.
4. **Project due: Dec 5, 2014**