

HOMEWORK 12/ PROGRAM 5 (10 points)

DUE 11:55 PM Friday December 8 with last submit date of 11:55 PM December 10

Your C program should already be doing the following. These are the requirements from Program 4

After the simulated TCP connection is established, the service will open a jpg file of your choice, read it and send it in 1500 bytes chunks. The client will acknowledge each packet. The server will initiate the FIN after the last ACK is received.

In addition to the requirements for the TCP handshake from Program 3, this version will have the following enhancements:

- Number every packet sent according to the sequence number of the first byte it contains.
- Store this packet in a transmission buffer. Design a linked list to do this.
- When the ACK arrives remove the corresponding packet from the transmission buffer.

Use the following fields for each data packet:

1. Source TCP port number –
2. Destination TCP port number –
3. Sequence number –
4. Acknowledgment number -
5. TCP data offset - Make it all zeros for now
6. Reserved data – Make it all zeros for now
7. Control flags – flags should be set correctly as needed
8. Window size – Use 1500
9. TCP checksum – Make it all ffffs for now
10. Urgent pointer – Make it all zeros

New requirements:

Send 3 packets at a time (transmission buffer should contain a maximum of 3 packets). Using multi-threading, implement a timer on the server side for x milliseconds where x is a large enough number (I would probably use 3 seconds unless you run into any issues with this)

Randomly drop a packet 25% of the time by not sending an ACK from the client.

When the timer expires, have the server resend all packets starting with the last ACK until the transmission buffer is full

Compile and run this program on LINUX. If UHUNIX, use a high port so there is no other application running on it.

Submit:

- Source code for server and client
- Proof of compilation using a make file
- Test output – (Use printf's to show the data packets being passed back and forth including all the fields in the simulated TCP header) DO NOT print the data bytes since this is binary
- Show the transfer was successful by viewing the image on the client and printing this.