Congestion Control

Steps to follow:

1. Run the command "***$make***" to create the binaries in the respective folder.
2. Change the directory to server and type ./udpserver <portname> to run the server side code
3. Open new terminal, change to client and type ./udpclient <serverip> <serverport> <filename> to run client side code and send the file.

Working of the code

Client side

- Initial packet containing the filename, file size and number of packets are sent.
- Two pointers, *start* and *end* are maintained for beginning and the end of the current window.
- A buffer of size equal to window size is maintained.
- All the packets in the current window are checked for their presence in buffer. If present, they will be resent, otherwise the checksum is updated, the packet is stored in the buffer and is resent.
- After each window is sent, it calls the *recvfrom* call for *windowsize* times and the maximum sequence number for which acknowledgement is received is kept track of.
- If the maximum sequence number is not equal to the sequence number corresponding to the end packet, window is made half (as it denotes packet loss) else the window size is doubled.
- The start pointer is pointed to the packet next to the max acknowledged packet and end pointer is pointed to start pointer plus the *windowsize.*
- The packets with sequence number less than max acknowledged seq number are removed from the buffer. The process is repeated till the entire file is sent.

Server side

- Initial packet containing the filename, file size and number of packets are received.
- A file with the received details is created.
- A variable, *expectedseqnumber* is maintained, which denotes the sequence number it is exepecting to receive
- If the sequence number received is not equal to *expectedseqnumber,* a duplicate acknowledgement is sent.
- If the sequence number received is equal to *expectedseqnumber ,* checksum is updated and the acknowledgement is sent.
- With *drop-probability,* the server drops the packets.