Lab 6

In this lab we will building off of the work you completed for lab 5. We will be modifying your Python client and server code (make sure to copy your files first) to implement the **Go - Back - N protocol** (see the lecture slides for detailed description). Also, you can check the below link for viewing an animation of GBN protocol :

http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/

The instructions are simple: implement the protocol using a window size of four. You must have a mechanism that shows how the protocol responds in the case of a dropped packet (a packet from sender to receiver). **Hint:** Think about how you might alter the sequence numbers in the packets leaving the sender. Make sure you hardcode a loop for sending messages one by one (to give an illusion of a ' stream ' of data) rather than getting input from user console.

Demo: Show the exchange of packets between your client and server in 2 cases:

i. No packets lost
ii. Packets lost

Note, we're looking for the correct recovery behavior for the protocol. To show output, you can use Wireshark; else if you prefer to print the output on terminal, make sure what you print makes clear how the protocol is working correctly (in terms of sequence numbers, etc.).

Be able to explain the difference in the protocol's response to a lost packet. If using Wireshark to explain your output, you'll have to think about what network interface you will have to capture the packets from and which filter you should use in Wireshark to most easily display your packet exchanges. (**Hint:** What underlying protocol are we using to send packets?) You might also use screenshots so that both packet exchanges can be looked at together for easier comparison.