Computer Networks Programming Assignment 3

Group Members:

Madhav Sainanee - 2017063
Rishabh Sharma - 2017087
Dhruv Kundu - 2017146
Jay Rawal - 2017240

Go-back-n protocol:

In the go-back-n protocol, then sender continues to send frames till a window size (which is predetermined) even without ACK packet from the receiver.

The receiver knows the sequence number of whichever frame it is expecting to receive.
If the sequence number of the frame is not equal to the sequence number that the receiver expects, the receiver discards it and resends the ACK for the last “correct” frame it receives.

After the sender has sent every frame (in the window), it will go back to the sequence number of last ACK it received from the receiver and will empty and then refill the window from that frame and continues it.

About server protocol:

We have used a connection-oriented TCP protocol to implement the server and client. These are used to implement the go back n protocol.

Code Analysis:

Both server and client can send packets to the other end of the connection. Let us suppose, for explanation sake, that client wants to send the packets to the server.

We have used python random number generator [0,1] to generate a value for each packet, and if that probability comes out to be greater than p = 0.9 then we drop that packet otherwise the packet is sent on the network.

If the packet was dropped, according to the above procedure, then the client wouldn’t receive an ack for that same.

Thus, the go-back-n protocol would come into effect and client would go back to the last packet sent correctly(for which it received an ack, in order) and would send the whole n-window to the receiver again starting from the first packet for which it didn't receive the ack correctly.

On the other hand, the receiver, when receives a packet, it puts a timestamp on it and sends the ack back to the sender and then again waits for the next packet.

If a packet received by the receiver is wrong (aka out of order) then:
It resets it count and wait for the correct (missed) packet again, followed by all other packets (in-order).

Results:

Packet drop probability is 1/2

Part a:	Time the first packet was	sent:	1573836865.190237
		received:	1573836900.999021
Part b:	Average Repeats by Client:	10.66	
	Average Repeats by Server:	6.866666666666666	

Packet drop probability is 1/10

Part a:	Time the first packet was	sent:	1573837938.3132982
		received:	1573837938.4133325
Part b:	Average Repeats by Client:	2.68	
	Average Repeats by Server:	3.5714285714285716	
