CS456/A2 Marking Scheme

There are 8 test cases that we use to mark your assignment. For each test case the similarity between input and output file worth 30% of that test case marks. The remaining 70% could be received iff your implementation exactly follows the Go-Back-N protocol for that particular test case. [There are no partial marks]

Make sure that the log of the emulator print meaningful seqnum since we also use the emulator log to examine the correctness of your implementation.

1. [35] Successfully transmit a file without delay and loss. Test SeqNum and window

1.1. [20] a file less than 10 packets (6 packets) without delay and loss.
exp result: logs: #0 -- #5 & correct transmitted file

1.2. [15] a file larger than 32 packets (43 packets) without delay and loss.
exp result: log: #0 -- #31 -- #10 & correct transmitted file

2. [10] Timer

Data: send #0 -- #6 (discard #3 data packet twice)
exp result: retransmit packets 3,4,5,6 twice & correct transmitted file

3. [45] GBN behavior

3.1. [5] first #0 data get lost
Data: send # 0,1,2,3,4,5,6 (#2 arrive after #3, and #5 get lost)
exp result: SeqNum.log 0,1,2,3,0,1,2,3
Arrival.log 1,2,3,0,1,2,3
ack.log 0,1,2,3

3.2. [10] data with delay and loss, test receiver side behavior.
Data: send # 0,1,2,3,4,5,6 (#2 arrive after #3, and #5 get lost)
exp result: SeqNum.log 0,1,2,3,4,5,6,3,4,5,6,
Arrival.log 0,1,3,2,4,6,3,4,5,6
Ack.log 0,1,1,2,2,2,3,4,5,6

3.3. [10] data with delay and loss, test window sliding.
Data: send # 0,1,2,3,4,5,6,...,16 (data #5 get lost)
exp result: SeqNum.log 0,...,9,...,14,5,...,14,15,16
Arrival.log 0,...,4,6,...14,5,...,14,15,16
Ack.log 0,...,4,...,4(9),5,...,14,15,16

3.4. [10] ACK with delay and loss, test cumulative ACK (all data received)
ACK: ack # 0,1,2,3,4,5,6 (ack #1 arrive after #2,#4,#6 get lost)
exp result: SeqNum.log 0,1,2,3,4,5,6,6
Arrival.log 0,1,2,3,4,5,6,6
Ack.log 0,2,1,3,5,6

3.5. [10] Both Data and ACK get delay and loss (file size > 32 packets(43))
Data: send #0,…,43, discard 11th(#10), 36th(#25) arrive after 37th(#26)
ACK: ack 6th(#5) get lost and 47th(#27) arrive after 48th(#28)
exp result: SeqNum.log 0,…,19,10,…,19,20,…,31,0,…,3,26,…,31,0,…,10
Arrival.log 0,…9,11,…,19,10,…,19,20,…,24,26,25,27,…,3,26,…,10
Ack.log 0,…4,6,…,9,…9(9),10,…,24,24,25,…,25(9),26,28,27,29,…,10

4. [10] Other Implementation Issues

4.1. [5] README & Makefile

4.2. [5] Comment and code style