NETWORK
2nd Term AY 2014-2015

A UDP-Based Reliable File Transfer
Machine Problem #2

You will implement an application that performs reliable data transfer over UDP. Each group should not exceed four members. You may use any programming language. There will be two applications, a sender and a receiver. The sender takes a file (any filetype) breaks it into chunks and send the chunks as UDP datagram. Packet loss will be simulated in the receiver side using a loss probability set at the command line or graphical user interface. The receiver then saves the file into its own directory. The following input information should be specified by the user:

For the sender:

1. The name of the file to be sent
2. Chunksize (in bytes)
3. Sender Window (in bytes)
4. Initial Sequence Number
5. Loss Probability for the acknowledgements
    a. A scale between 0-100. 100% means all packets are lost.
6. Timeout (in milliseconds)
    a. if a packet is not received within this time period a packet is deemed lost.
7. Delay (between successive datagram transmission)
8. IP Address of the receiver
9. Port number
    a. The same port number should be used to receive acknowledgements
10. Verbosity
    a. Level 1: Print the sequence number of lost packets
    b. Level 2: Print the sequence number of datagrams sent, acknowledgement numbers received, and lost packets.
    c. Level 3: Print the sequence number of datagrams sent, acknowledgement numbers received, and lost packets with time stamp.

For the receiver:

1. Filename (to save the file)
2. Receiver window (in bytes)
3. UDP Port number
4. Delay (in processing the received datagram)
    a. This will be needed to test flow control.
5. Loss Probability
    a. A scale between 0-100. 100% means all packets are lost.
6. Timeout (in milliseconds)
    a. if a packet is not received within this time period a packet is deemed lost.
7. Verbosity
    a. Level 1: Print sequence numbers of dropped packets.
    b. Level 2: Print sequence number of received datagrams and indicate if it is lost, discarded or saved.
    c. Level 3: Print sequence number of received datagrams and indicate if it is lost, discarded or saved with time stamp.

This is divided into four milestones. You will be designing your own headers. Sequence and acknowledgement numbers are both 16 bits. You can't continue to the next milestone without finishing the previous milestone. The requirements for each milestone are not necessarily in order.

Milestone 1: A Stop and Wait Reliable Data Transfer (40%)
In a stop and wait reliable data transfer, the sender sends one packet at a time. It then waits for an acknowledgement from the receiver before sending the next packet. The packets received are then assumed to be in order.
This milestone is worth 40% and are distributed as follows. T

1. Successful File Transfer at 0% loss. Make sure you terminate the file transmission. (10%)
2. Implementation of sequence and acknowledgement numbers. (10%)
3. Implementation of packet loss and timeout at both sender and receiver. (20%)

The sender and receiver should print the statistics of the number of byte datagrams received, lost and retransmitted and the total elapsed time.

Milestone 2: Implementation of Go-Back-N (85%)
As compared to the stop and wait protocol, Go-Back-N is an example of a pipeline protocol which allows multiple packets to be sent at the same time. A sender window is then added to the system, hence adding more complexity to our reliable data transfer.

1. The sender sends multiple packets without waiting for an acknowledgement. The receiver only accepts in-order packet. If a packet received is out of order it will discard that packet. (15%)
2. The timer is implemented for the last unacknowledged packet (10%)
3. The sender processes cumulative acknowledgements. This can be seen when acknowledgements are lost. (10%)

Milestone 3: Implementation of TCP Reliable Data Transfer (125%)
TCP is a hybrid of Go-Back-N and Selective Repeat. Hence, a receiver window is added to the system. In this case, the received packets are not necessarily in order anymore as long as the receiver window can still accommodate them.

1. Implementation of the receiver window. When a packet is received, the receiver buffers it to the receiver window but sends an acknowledgment for the last in-order packet. (15%)
2. Implementation of Fast Retransmit (10%)
3. Implementation of Flow Control (5%)
4. Computation of the Round Trip Time and adjustment of the timeout value. (10%)

Milestone 4: Congestion Control (175%)
This milestone is worth 50%. The congestion control mechanism is not discussed in class hence you may have to study it by yourself. You can ask the instructors if you have any questions. The details of this milestone will be released once a group have fully implemented Milestone #3.

Deliverables
You are required to give a 10 minute demo and presentation of your program that shows the requirements of each milestone. Presentations shall be scheduled on December 5. A final documentation which should be submitted before midnight of December 5 in PDF format. A pre-checking will be scheduled on November 12 and 13. Your final documentation must explain how each milestone is reached and answer the questions accordingly. All machine problem codes and the final documentation should be sent to your instructor's email address with a subject of "NETWORK [section] Final Docu". Make sure your codes compiles without any error.