Programming project2
Due date : midnight of August 12th , 2011

UDP Client/Server , Reliable data transfer

In this project you will modify the UDP Clinet-Server program in the text book (chapter 2), such that the client can send more than one line to the server. You will also implement a reliable data-transfer protocol for this application, RDT, such that using GO_BACK_N method, the packet loss is recovered by retransmitting the packet. The project has two steps.

Step1:
Modify UPD client-server program in the text book as follows:
1- Modify the client program in a way to accept the server_host_name as input argument when executing it in command line.
2- Client opens a text file named "LowerCase.txt", reads each line and sends it to the server.
3- Server converts each received line to to upper case and writes it in a text file on the server named "UpperCase.txt".

Step2:
Implement a reliable data transfer protocol, RDT, for the above application such that your protocol can handle packet loss using GO_BACK_N method.

RDP Packet :

Every RDT packet transmitted between client and server has the following structure:

- Maximum size of the RDT packet is 100 byte: (1byte header + 99 byte data)
- The fields of header are defined as follows:
  - **Ver (3bits):** indicates the version of RDT and is set 1.
  - **Type (2bits):** defines the type of the packet and can be one the following values:
    - 0 : Data (DAT) packet : sent by the client and carries data.
    - 1: Acknowledgement (ACK) packet: sent by the server to acknowledge the reception of DAT,SYN or FIN packet.
    - 2: SYN packet : sent by the client to initiate the data transfer.
    - 3: FIN packet: sent by the client to finish data transfer.
  - **Seq/Ack Number (3bits)** : the sequence number of packets are between 0 to 7. (Don't follow the text book rules to compute the sequence number of packets). You need to assign 0,1,2,3,...,7,0,1,2,..., numbers to the sequential packets.

I have written a sample class, general.java (www.cs.sfu.ca/CC/371/mmarzban/assignments.html), that gives you an idea on how to create RDT packet in an array of bytes and also how to extract data and information of the header from a RDT packet .

Client or Sender

The following pseudo code describes the client actions:

begin

(1) Send SYN packet, start timer
(2) if time out and no ACK packet
resend SYN packet, start timer, go to the beginning of the step
otherwise start transferring data
(3) While there is an unread line in "LoweCase.txt" file
transfer data based on the FSM for the sender in
GBN in the text book (page 233)
(4) Send FIN Packet to the server.
(5) if time out and no ACK packet
resend FIN packet, start timer, go to the beginning of the step
otherwise close "LowerCase.txt" file.

end

-The size of the window in GBN is 3 and the retransmission time is 500 milliseconds.

Server or Receiver

The corresponding pseudo code for the server is as follows:

begin

(1) wait for SYN packet
(2) Send ACK packet for SYN
(3) While receive DAT packets
transfer data based on the FSM for the receiver in
GBN in the text book (page 233).
(4) if receive FIN packet, send ACK and close the "UpperCase.tst" file.

end

Submission:

1- Create a directory X where X is your surname. Put all files that you create in the directory X. (at least two files required: UDPServer.java, UPDClient.java)

2- Compress your directory X to create a .zip file.

3- Submit your compressed file online at https://courses.cs.sfu.ca/ .