Sender Server:

1. Sender creates a data structure for sending the data as frames containing the following attributes: seq\_no (integer), ack (integer), data(char array).
2. Sender server can select to send data or receive ack.
3. send\_frame.seq\_no is set as value of counter which alternates between 0 and 1.
4. Set send\_frame.ack as 1.
5. Sender takes user input as data of send\_frame.
6. Set x as 0 for checking the correctness of ack.
7. send\_frame is send to the receiver.
8. Immediately after sending the data a timer is started.
9. If ack with expected seq\_no is received within the timeout limit, set x as 1.
10. Else print timeout occurred indicating loss of data/ack.
11. Repeat steps 7-10 until x is 1.
12. If recv\_size > 0 and recv\_frame.ack = counter, expected ack is received, counter is set a s (counter+1)%2 and sender prompts for user input.
13. Else print ack not received.
14. Repeat steps 3-13 indefinitely.

Receiver Server:

1. Frame data structure is created in the receiver.
2. Counter is initially set as 0.
3. A recv\_frame is received from sender
4. If recv\_size > 0 and recv\_frame = counter
   1. print seq\_no and data of the frame
   2. set send\_frame.seq\_no and send\_frame.ack as counter
   3. send send\_frame to sender
   4. print the seq\_no of ack
   5. counter is set as (counter+1)%2
5. Else
   1. set ssend\_frame.seq\_no and send\_frame.ack as recv\_frame.seq\_no
   2. send send\_frame to sender
   3. print ack sent indicating seq\_no of the data receive is incorrect.
6. Repeat 3-5 indefinitely.