Sender window
- Contains all the packets that are ready to send or already sent, but not yet acknowledged
- In-order acknowledgments move window to the right – if this means that new packets fall into the window you can now send these new packets to the receiver
- Only packets in the window can be sent

Receiver window
- Contains all the packets that are received but not yet sent to the application
- In-order packets can be sent to the application – this moves window to the right

Go-back-N sender
- Sender has timer per window
- Out-of-order acknowledgments are ignored
- On timeout, sender retransmits the whole window

Go-back-N receiver
- In-order packets generate acknowledgments
- Out-of-order packets are discarded and acknowledgment is sent for the last packet that was received in order

Selective-repeat sender
- Sender has timer per packet
- Out-of-order acknowledgments lead to corresponding packets being marked as received
- On timeout, one packet is retransmitted – the one whose timer expired

Selective-repeat receiver
- A packet that falls in window is stored (even if it is out-of-order) and acknowledgment for this packet is sent to the sender
- A packet that falls before the window is discarded but acknowledgment for it is sent to the sender

TCP with fast retransmit sender
- Sender has timer per window – timer expiration means there has been a loss
- Duplicate acknowledgments are counted – third duplicate acknowledgment means that there has been a loss (this is fast retransmit)
- On loss, sender retransmits the leftmost packet in the window (when we add congestion control we will change this behavior)

TCP with fast retransmit receiver
- All packets generate acknowledgments bearing the number of the next expected packet
- In-order packets are stored, out-of-order packets are discarded
These problems can bring you up to 11 makeup points towards your midterm grade. Solve these problems and bring the answers to the TA or the instructor until the last lecture. You are not allowed to ask your classmates for help but you can ask the instructor or the TA. If the TA or the instructor provides significant help towards the problems she may issue a replacement problem for you to solve so you can earn the points that the problem was carrying.

1. **(3 points)** Host A sends 20 packets to host B. After each packet B generates an acknowledgment message and sends it to A. Packets 2, 3, 4, 15, 16 are lost and acknowledgments for packets 8, 9 and 17 are lost. Show a sequence of packets and acknowledgments between A and B in case of selective-repeat with sender and receiver window size equal to 6. For each packet write P_x above the arrow-line if this is packet with sequence number x; for each acknowledgment for packet P_x write ACK_x above the arrow-line. For the following events write a message specifying the event in the space left and right of vertical lines: packet is stored at B, acknowledgment is accepted at A, A’s or B’s window is shifted to the right, B delivers packets to the application, packet or acknowledgment is discarded, timeout occurred. Assume that all retransmitted packets arrive at B. Sender application has nothing more to send after packets 1-20. A’s timeout value is 1s and RTT is 100ms between A and B. Sequence number space is very large and each packet is 1 byte long. In all cases assume that sequence numbers start from 1.

2. **(3 points)** Host A sends 20 packets to host B. After each packet B generates an acknowledgment message and sends it to A. Packets 1, 3, 4, 10 are lost and acknowledgments for packets 8, 10 and 17 are lost. Show a sequence of packets and acknowledgments between A and B in case of go-back-N with sender and receiver window size equal to 6. For each packet write P_x above the arrow-line if this is packet with sequence number x; for each acknowledgment for packet P_x write ACK_x above the arrow-line. For the following events write a message specifying the event in the space left and right of vertical lines: packet is stored at B, acknowledgment is accepted at A, A’s or B’s window is shifted to the right, B delivers packets to the application, packet or acknowledgment is discarded, timeout occurred. Assume that all retransmitted packets arrive at B. Sender application has nothing more to send after packets 1-20. A’s timeout value is 1s and RTT is 100ms between A and B. Sequence number space is very large and each packet is 1 byte long. In all cases assume that sequence numbers start from 1.

3. **(3 points)** Host A sends 20 packets to host B. After each packet B generates an acknowledgment message and sends it to A. Packets 2, 10, 12 and 16 are lost and acknowledgments for packets 8 and 17 are lost. Show a sequence of packets and acknowledgments between A and B in case of TCP with fast retransmit with sender and receiver window size equal to 6. For each packet write P_x above the arrow-line if this is packet with sequence number x; for each acknowledgment for packet P_x write ACK_x above the arrow-line. For the following events write a message specifying the event in the space left and right of vertical lines: packet is stored at B, acknowledgment is accepted at A, A’s or B’s window is shifted to the right, B delivers packets to the application, packet or acknowledgment is discarded, timeout occurred. Assume that all retransmitted packets arrive at B. Sender application has nothing more to send after packets 1-20. A’s timeout value is 1s and RTT is 100ms between A and B.
between A and B. Sequence number space is very large and each packet is 1 byte long. In all cases assume that sequence numbers start from 1.

4. **(2 points)** Host `mylaptop.udel.edu` wants to talk to host `www.iana.org`. Assume that DNS cache of `mylaptop` only contains information about a local name server at `udel.edu` called `dns.udel.edu`. DNS client at `mylaptop` sends a recursive query to `dns.udel.edu`, while `dns.udel.edu` sends iterative queries. Local name server’s cache only contains information about one root DNS server `root1.dns.us`. Draw all the participants and DNS messages exchanged before `mylaptop` obtains a DNS mapping for `www.iana.org`. Now draw messages that are exchanged when we ask for `www.ieee.org`. 