Time-line examples of protocols

Stop-and Wait (RDT 3.0) protocol
   Also called the alternating bit protocol
Go-back-N protocol
Selective Repeat protocol

The conventions used in these examples regarding when timers start and stop, processing times, and so forth will be that used on exams.
Rdt3.0 timeline example problem:
Suppose that the transmission time for a packet by the sender is 1 sec., and the transmission time for an ACK by the receiver is 0.5 sec., and the propagation time for a bit sent from the sender to the receiver (or receiver to sender) is 2 seconds.

Suppose that the processing time for a complete packet or ACK is negligible (0 sec). The amount of time for a receiver to deliver a successfully received packet to the application layer is similarly negligible.

Suppose at time = 0 the sender begins to send a sequence of 4 packets of identical size to the receiver using the ABP rdt3.0 with a timer set to 6 sec. The timer in the sender starts immediately when the first bit of the packet has been transmitted, and the timer stops when the last bit of the ACK is received. The sender uses sequence # 0 for the first packet sent.

While attempting to successfully deliver this 4 packet sequence with rdt3.0, three problems occur: (i) during the initial attempt to send the 2nd packet in this sequence of 4 packets, the packet is corrupted in transit; (ii) in the initial attempt to transmit the 3rd packet, the timer prematurely has a timeout in 3 sec; (iii) the initial acknowledgement of the 4th packet of the sequence is lost.

1. At what time is the 2nd packet of the sequence successfully delivered to the receiver?

2. What is the sequence number used for 4th packet successfully delivered to the transport layer of the receiver?

3. At what time does the receiver successfully deliver the 3rd packet in the sequence to the application process?

4. What is the minimum timeout value that could successfully be used in this rdt3.0 protocol?

5. What is Utilization value for the successful delivery of the set of three packets by the send to the receiver?
1st bit of 1st packet with seq#0 is transmitted by send to rec

Last bit of ACK for 1st packet reaches send at 5.5 sec, and send begins to transmit 2nd packet with seq#1

Last bit of duplicate ACK for 1st packet (with seq#0) reaches send at 11 sec, and send ignores it

Timeout for 2nd packet and retransmit at 11.5 sec. Note that it was actually known by sender from the duplicate ACK received at 11 sec that a resend of packet (seq#1) is necessary, but rdt3.0 waits until timeout before resending

Last bit of ACK for 2nd packet reaches send at 17 sec, and send begins to transmit 3rd packet (with seq#0)

PREMATURE TIMEOUT for 3rd packet (seq#0) and retransmit at 20 sec

Last bit of ACK for first transmission of 3rd packet reaches send at 22.5 sec, and send begins to transmit 4th packet (with seq#1) to rec

Last bit of ACK with seq#0 for second transmission of 3rd packet reaches send at 25.5 sec, and send ignores it

Timeout for 4th packet and retransmit at 28.5 sec

Last bit of ACK for 4th packet reaches send at 34 sec,

1st bit of 1st packet reaches rec at 2 sec

Last bit of 1st packet reaches rec. at 3 sec and rec begins to transmit ACK (with seq#0) and delivers packet to application layer

Last bit of corrupted 2nd packet reaches rec at 8.5 sec and rec discards it, and then retransmits duplicate ACK for 1st packet (with seq#0)

Last bit of 2nd packet reaches rec at 14.5 sec, and rec begins to transmit ACK with seq#1

Last bit of duplicate of 3rd packet reaches rec at 23 sec, and rec begins to retransmit ACK with seq#0

Last bit of 3rd packet reaches rec at 20 sec, and rec begins to transmit ACK with seq#0

Last bit of 4th packet reaches rec at 25.5 sec, and rec begins to transmit ACK with seq#1

Last bit of 4th packet reaches rec at 31.5 sec, and rec begins to transmit ACK with seq#1

Last bit of ACK with seq#0 for second transmission of 3rd packet reaches send at 25.5 sec, and send ignores it

Last bit of ACK with seq#0 for 4th packet reaches send at 34 sec,
Suppose that the transmission time of a packet by the sender is 1 sec., and the transmission time of an ACK by the receiver is 1 sec., and the propagation time for a bit sent from the sender to the receiver (or receiver to sender) is 3 sec.

Suppose that the processing time for a complete packet or ACK is negligible (0 sec.). The amount of time for the transport layer of the receiver to deliver a non-corrupted completely received packet to the application layer is similarly negligible.

Suppose at time = 0 the sender begins to send a sequence of 7 packets (denoted P1, P2, …, P6, P7) of identical size to the receiver using the GBN where N=3 with a single timer set to 10 sec.

The single timer runs only for the packet with the base sequence number. When the sender gets the ACK for that packet, the timer is re-started for the packet that has the new base sequence number.

1. Using the smallest set of sequence numbers possible for the successful transmission of 6 the packets using GB3, show the complete time diagram for their delivery to the receiver assuming there is no packet loss or corruption.

Assume next that while attempting to successfully deliver this 7 packet sequence with GB3, **two problems occur:**
1. The Ack for the 2nd packet of the sequence is lost.
2. The initial transmit of the 5th packet of the sequence is corrupt.

2. At what time(s) is the P6 packet of the sequence successfully delivered to application layer of the receiver?

3. What is the sequence number used for the 6th packet successfully delivered to the transport layer of the receiver?

4. At what time does the receiver successfully deliver the 3rd packet in the sequence to the application process?

5. What is the minimum timeout value that could successfully be used in this rdt3.0 protocol?

6. What is Utilization value for the successful delivery of the 3 packets by the sender to the receiver?

7. Indicate the times when the packets are delivered to the application layer.
How does GBN handle replicate Acks? In this example, the duplicate ACK seq#3 shown causes a GB3 event. Instead, sender could have ignored multiple Acks and waited for timeout at t=26 secs (Slow!) before GB3 event.

Note that a triplicate Ack seq#3 occurs, but it will be assumed that the sender is smart enough to know P5 has already been resent following duplicate Ack, and so there is not another P5 resend and the timer is not restarted again.

TCP has the sender preempt the timer only when the sender sees a triplicate Ack.

Go Back N Example: Window size N=3. Packet/Ack transmission time = 1 sec. Propagation time 3 sec. Negligible queuing/processing time. Timeouts = 10 sec., and timer is started when 1st bit of packet is sent, and is stopped when last bit of packet is received or when it times out.
A sequence of 7 packets denoted P1,P2, ...,P7 must be delivered to rec. Transmission problems: 1st ACK of P2 lost; 1st transmit of P5 corrupted.
Seq#'s used: [0,1,2,3].
Suppose at time = 0 the sender begins to send a sequence of 7 packets (denoted P1, P2, P3, P4) of identical size to the receiver using the GBN where N=3 with a single timer set to 10 sec.

The single timer runs only for the packet with the base sequence number. When the sender gets the ACK for that packet, the timer is re-started for the packet that has the new base sequence number.

Using the smallest set of sequence numbers possible for the successful transmission of the 4 packets using GB3

This example shows that at least 4 sequence #’s are needed with a window N=3. It can be shown that for GBN with window size N, N+1 seq #’s are necessary and sufficient.
Selective Repeat Example: Window size N=3. A sequence of 7 packets denoted P1, P2, …, P7 must be delivered to rec. Transmission problems: 1st ACK of P2 lost; 1st transmit of P5 corrupted. Seq#’s used: [0, 1, 2, 3, 4, 5]. Timeouts = 10 sec.

Sendwindow = P1(seq#0), P2(1), P3(2)
This shows packs sent but not acked

Recwindow = P4(seq#3), P5(4), P6(5)
This shows packs that rec expects

Sendwindow = P2(seq#1), P3(2), P4(3)

Recwin = P5(4), P6(5), P7(0)

Sendwindow = P5(seq#4), P6(5), P7(0)

All 7 packets delivered and ACKed in 37 sec.
Utilization = 7/37