CMPSCI 453
HW10, Problem Set 4

Tony Gao

March 26, 2018

1 Problem 1

a False, each sequence number in the sender window in SR must be acknowledged for the sender window to move forward. It is possible for the receiver to acknowledge a sequence number outside of the receiver window, but impossible for the sender to receive an ack outside of the sender window. The sender window does not move forward until the earliest unacked packet is acked.

b False, when the sender window moves forward in GBN, it is guaranteed that all packets up to and including send_base have been acknowledged. If a certain packet is lost, all packets that are transferred after it within the sender window will be retransmitted until all packets have been cumulatively acked. It is possible that the receiver receives a packet that it has already acked. However, the sender is guaranteed by design to not send a packet that is outside of the sender window.

c True. The behavior in RDT3.0 is the same as SR with a sender/receiver window of 1. If a corrupted packet is received by the receiver, no ack is sent, causing a timeout and retransmission. If a duplicate packet is received by the receiver, the sequence number of the duplicate packet is acked. If an ack is lost, the sender will timeout and resend the packet. The sender does not move to the next sequence number until the previous sequence number is correctly acked. This mirrors SR.

d False. As a counterexample: If a duplicate packet is received by the receiver in GBN, the sequence number of the next expected sequence number is sent due to the cumulative ack. In RDT 3.0, the sequence number of the duplicate packet is sent in the ack. Additionally, there is no receiver window in GBN, so the premise of this statement is false.

2 Problem 2

For the time intervals, interval notation is used to denote open and closed intervals.

1.
- Slow Start: [1,4), [8, 11), [27, 31)
- Congestion Avoidance: [4,8), [11,16), [17, 27) [31, 40]
- Fast recovery: [16, 17)
  Fast recovery has not timed out because cwnd continues to grow after T=17, implying the timeout is about 1 RTT. At T = 17, a new ack for a sequence coming after the duplicate ack number must have arrived at the sender, transitioning TCP to congestion avoidance per the FSM.

2.
- Packet is lost just after T = 7. Timeout occurs just before T=8 since cwnd = 1 at T=8.
- Packet is lost just after T=15, cwnd is set to $12/2 + 3 = 9$ at T=16 due to triple duplicate ack loss event.
- Packet is lost just after T=26. Timeout occurs at T=27 since cwnd is set to 1 at T=27.

3.
- SSThresh set to $11/2 = 5$ at T=8 since cwnd = 11 at T=7, when the timeout loss event occurred
- SSThresh set to $12/2 = 6$ at T= 16 since cwnd = 12 at T=15, when the triple duplicate ack loss event occurred.
- SSThresh set to $19/2 = 9$ at T=27 since cwnd = 19 at T=26, when the timeout loss event occurred.

3 Problem 3

For the time intervals, interval notation is used to denote open and closed intervals.

1.
   - Slow start: [1,13), [20,22), [35,38)
   - Congestion Avoidance: [13,20), [22, 27), [28,32), [33,35), [38,40]
   - Fast Recovery: [27, 28), [32,33)
2.
   - Lost just after T=3, timeout just before T=4.
   - Lost just after T=7, timeout just before T=8
   - Lost just after T=11, timeout just before T=12
   - Lost just after T=19, timeout just before T=20
   - Lost just after T=26, triple duplicate ack just before T=27
   - Lost just after T=31, triple duplicate ack just before T=32
   - Lost just after T=34, timeout just before T=35
3.
   - ssthresh set to $4/2 = 2$ at T=4.
   - ssthresh set to $4/2 = 2$ at T=8.
   - ssthresh set to $4/2 = 2$ at T=12.
   - ssthresh set to $8/2 = 4$ at T=20.
   - ssthresh set to $8/2 = 4$ at T=27.
   - ssthresh set to $11/2 = 5$ at T=32.
   - ssthresh set to $10/2 = 5$ at T=35.

4 Problem 4

S-to-R	Time Segment Sent	S-to-R sequence #	Time Segment Received	R-to-S ACK #
Segment 1	1	91	8	$91 + 564$
Segment 2	2	$91 + 564$	9	$91 + 2 * 564$
Segment 3	3	$91 + 2 * 564$	segment was lost	segment was lost, no ack
Segment 4	4	$91 + 3 * 564$	segment was lost	segment was lost, no ack
Segment 5	5	$91 + 4 * 564$	12	$91 + 2 * 564$

S-to-R	Time Segment Sent	S-to-R sequence #	Explanation
Segment 1	15	$91 + 5 * 564$	Currently 4 unacked segments in the cwnd since ack for segment sent at T=1 was received at T=15, so there is space for 1 more sequence to be sent
Segment 2	16	$91 + 6 * 564$	Currently 4 unacked segments in cwnd since ack for segment sent at T=2 was received at T=16, so there is space for 1 more sequence to be sent
Segment 4	19	$91 + 2 * 564$	The cwnd was full from T=16 until T=19. There have not yet been 3 duplicate acks, so this transmission must be due to timeout on the oldest not acked segment, which is the one sent at T=3 with sequence number $91 + 2 * 564$

5 Problem 5

a. With 4 byte sequence numbers, the maximum amount of sequence numbers and thus maximum number of bytes is $2^{32} - 1B = 4294967295B$

b.

$$\text{Number of segments that the file must be divided into} : ceil((2^{32} - 1)/536) = 8012999 \quad (1)$$

$$\text{Total header overhead} : 8012999 * 66B = 528857934B \quad (2)$$

$$\text{Total transmission size} : 4294967295B + 528857934B = 4823825229B \quad (3)$$

$$\text{Transmission time to nearest ms} : 4823825229B/(155Mbps/8 * 1000000)Bps \approx 248.972s \quad (4)$$

6 Problem 6

S-to-R	Time Segment Sent	S-to-R sequence #	Time Segment Received	R-to-S ACK #
Segment 1	1	148	8	$148 + 509$
Segment 2	2	$148 + 509$	9	$148 + 2 * 509$
Segment 3	3	$148 + 2 * 509$	Not received	Not received, no ack sent
Segment 4	4	$148 + 3 * 509$	Not received	Not received, no ack sent
Segment 5	5	$148 + 4 * 509$	Not received	Not received, no ack sent

1. Segment 1 R-to-S explanation: TCP Cumulative ACK defines an ack for a received segment as equal to the sequence number of the next expected byte, which is $148 + 509$
2. Segment 2 R-to-S explanation: TCP Cumulative ACK defines an ack for a received segment as equal to the sequence number of the next expected byte, which is $148 + 2 * 509$
3. Segment 3 R-to-S explanation: There is no ack sent, because the segment was never received.
4. Segment 4 R-to-S explanation: There is no ack sent, because the segment was never received.
5. Segment 5 R-to-S explanation: There is no ack sent, because the segment was never received.