Quinn Roemer

Professor Xuyu Wang

CSC 138

November 29, 2020

1. **Computing an Internet Checksum**

Consider the two 16-bit words shown in binary below. Computer the internet checksum value for these two 16 bit words.

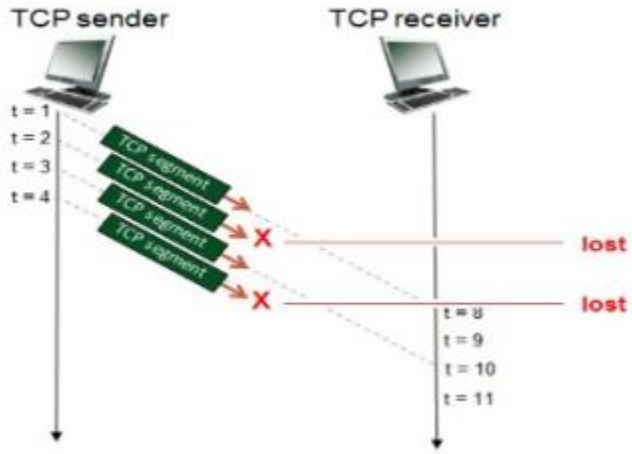
0110 0110 1001 1111 + 1010 1010 1101 0001 = 0001 0001 0111 0000 (add the 17th carry bit to the 1s places)

0001 0001 0111 0001 (Take the 1s complement)

1110 1110 1000 1110 (Checksum)

2. **TCP Sequence Number**

A TCP sender sends for segments at t=1, 2, 3, 4 respectively. Suppose the initial sequence number is 149 and the first four segments contain 556 bytes. The delay between the sender and receiver is 7 time units, hence, the first segment arrives at the receiver at t=8. Two of the four segment(s) are lost. Answer the following questions.



- Give the sequence number associated with each of the four segments send by the sender

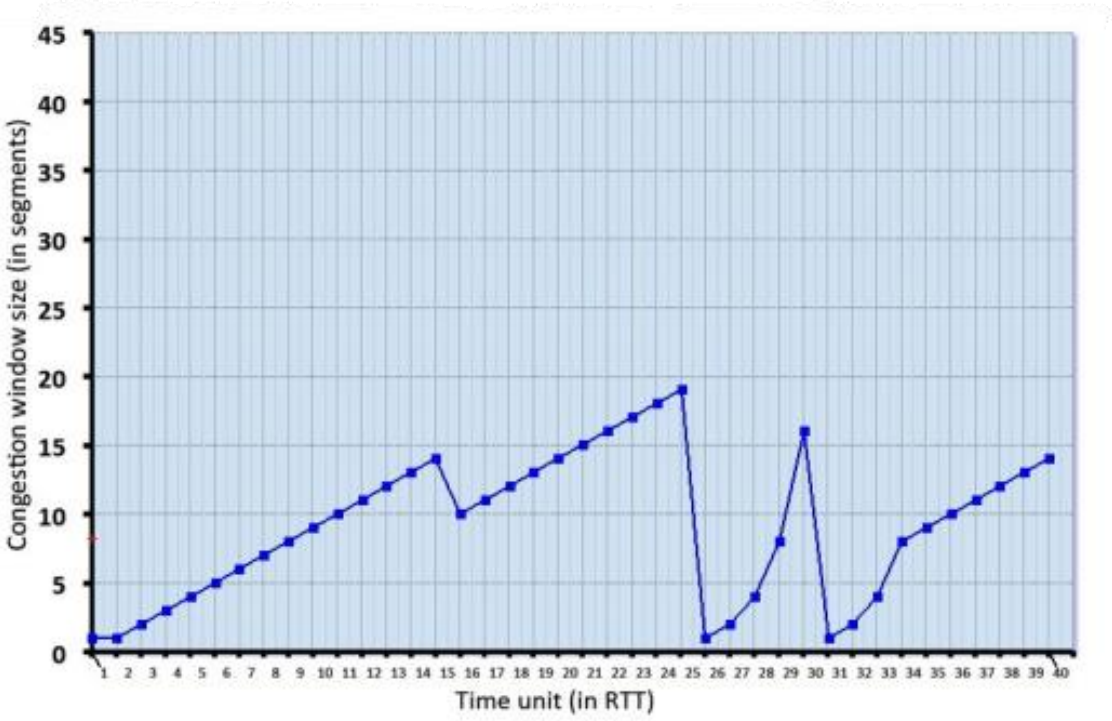
Segment 1	Segment 2	Segment 3	Segment 4
149	705	1261	1817

- List the sequence of acknowledgements transmitted by the TCP receiver in response to the receipt of the segments received. Give the value in the acknowledgment field of each receiver-to-sender acknowledgement and give a brief explanation as to why that acknowledgement number is being used.

Segment 1	Segment 2	Segment 3	Segment 4
705	Segment not received	705	Segment not received
Segment received successfully, as a result the sender sends back an ACK with initial + bytes received which equals 705	No ACK is sent as the segment fails to arrive.	Out of order segment received. Due to cumulative acknowledgements ACK's only acknowledge up to the first missing byte. As a result, 705 is sent again.	No ACK is sent as the segment fails to arrive.

3. **TCP in Action: Slow Start, Congestion Avoidance, and Fast Retransmit**

TCP sends a "flight" of packets of size *cwnd* at the beginning of each time unit. The result is either all packets are ACK'ed, there is a timeout, or there is a triple duplicate ACK for the first packet. In this problem reconstruct the sequence of events (ACKs, losses) that result in the evolution of TCP's *cwnd* shown below:



The initial value of *cwnd* is 1 and the initial value of *ssthresh* (show as a red +) is 8. Answer the following question below.

- Give the times at which TCP is in slow start, congestion avoidance, and fast recovery at the start of a time slot, when the flight of packets is sent

Slow Start	1, 2, 26, 27, 28, 29, 31, 32, 33
Congestion Avoidance	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 30, 34, 35, 36, 37, 38, 39, 40
Fast Recovery	16

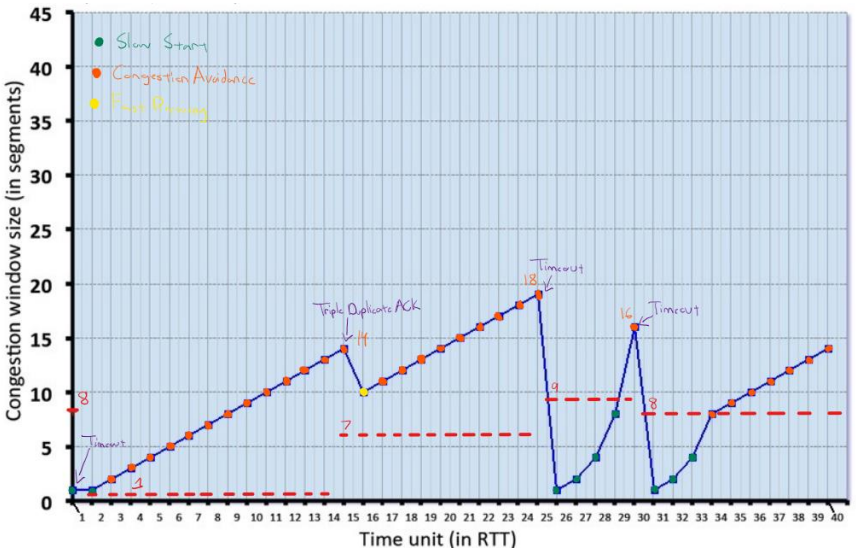
- Give the times at which the first packet in the sent flight of packets is lost and indicate whether that packet loss is detected via timeout, or by triple duplicate ACK's.

Timeout Loss	1
Triple Duplicate ACK's	15
Timeout Loss	25
Timeout Loss	30

- Give the times at which the value of *ssthresh* changes and give the new value of *ssthread.*

(Initial Value = 8)

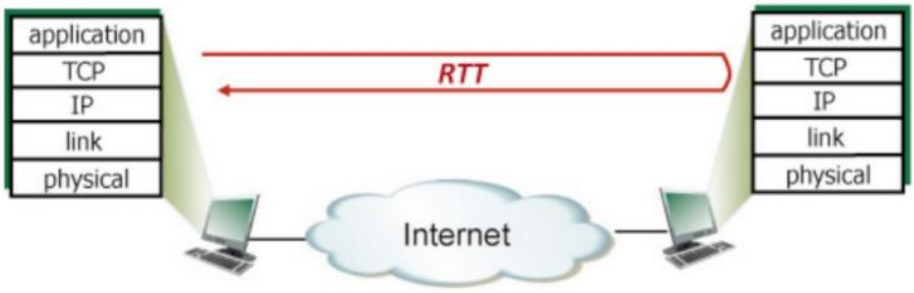
Time = 1 *cwnd* = 1	*ssthread* = 1
Time = 15 *cwnd* = 14	*ssthread* = 7
Time = 25 *cwnd* = 18	*ssthread = 9*
Time = 30 *cwnd* = 16	*ssthread* = 8



(Result)

4. **Computing TCP's RTT and Timeout Values**

Suppose current estimated values for round trip time (estimatedRTT) and deviation in RTT (devRTT) are 270msec and 29msec, respectively. Suppose that the next 3 measured values of the RTT are 230, 290, and 210, respectively. Compute TCP's new value of estimatedRTT, devRTT, and the TCP timeout value after each of these three measured RTT values is obtained. Use the values of $\alpha = 0.125$ $\beta = 0.25$.



	estimatedRTT	*devRTT*	*TCP Timeout*
Initial	**270**	**29**	**NA**
RTT1 230	265	31.75	392
RTT2 290	268.13	30.06	388.37
RTT3 210	260.86	37.08	409.18

RTT1

estimatedRTT = (1 – 0.125) * 270 + 0.125 * 230 = **265**

devRTT = (1 – 0.25) * 29 + 0.25 | 270 – 230 | = **31.75**

TCP Timeout = 265 + (4 * 31.75) = **392**

RTT2

estimatedRTT = (1 – 0.125) * 265 + 0.125 * 290 = **268.13**

devRTT = (1 – 0.25) * 31.75 + 0.25 |265 – 290 | = **30.06**

TCP Timeout = 268.13 + (4 * 30.06) = **388.37**

RTT3

estimatedRTT = (1 – 0.125) * 268.13 + 0.125 * 210 = **260.86**

devRTT = (1 – 0.25) * 30.06 + 0.25 | 268.13 – 210| = **37.08**

TCP Timeout = 260.86 + (4 * 37.08) = **409.18**