

# CSC138\_sec06\_ASSIGNMENT5\_Matthew\_Mendoza - ASSIGNMENT 5: Homework #5

Author: Matthew Mendoza

Course: CSC 138

Section: 06

- [CSC138\\_sec06\\_ASSIGNMENT5\\_Matthew\\_Mendoza - ASSIGNMENT 5: Homework #5](#)
  - [Question 1: Computing an Internet checksum \(1 point\)](#)
    - [SOLUTION 1: Computing an Internet checksum](#)
  - [Question 2 Prompt](#)
    - [Question 2.1](#)
      - [Solution 2.1](#)
    - [Question 2.2](#)
      - [Solution 2.2](#)
  - [Question 3 Prompt: TCP in action: slow start, congestion avoidance, and fast retransmit. \(2 points\)](#)
    - [Question 3.1](#)
      - [Solution 3.1](#)
    - [Question 3.2](#)
      - [Solution 3.2](#)
    - [Question 3.3](#)
      - [Solution 3.3](#)
  - [Question 4: Computing TCP's RTT and timeout values](#)
    - [Solution 4](#)
  - [All Solutions](#)
    - [Question 1 : Solution 1](#)
    - [Question 2 : Solution 2.1](#)
    - [Question 2 : Solution 2.2](#)
    - [Question 3 : Solution 3.1](#)
    - [Question 3 : Solution 3.2](#)
    - [Question 3 : Solution 3.3](#)
    - [Question 4 : Solution 4](#)

## Question 1: Computing an Internet checksum (1 point)

Consider the two 16-bit words (shown in binary) below. Recall that to compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum [1] of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result.

**Compute the Internet checksum value for these two 16-bit words:**

01100110 10011111 this binary number is 26271 decimal (base 10)

10101010 11010001 this binary number is 43729 decimal (base 10)

SOLUTION 1: Computing an Internet checksum

bit placement	17th	16th	15th	14th	13th	12th	11th	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st	result
bin value		32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
----- --	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----
carry bits		1	1		1	1	1		1			1	1	1	1	1		
value one		0	1	1	0	0	1	1	0	1	0	0	1	1	1	1	1	26271
value two		1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1	43729
wrap-around carry bit	(1)	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	
sum		0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	

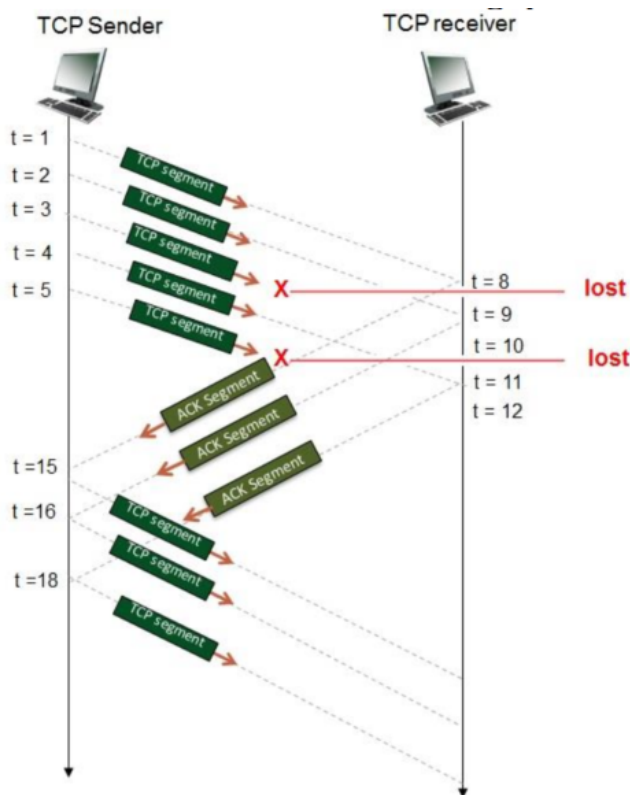
bit placement	17th	16th	15th	14th	13th	12th	11th	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st	result
----- --	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----
		0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	
add wrap-around to sum																	(1)	
		=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	
result		0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	
----- --	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----
flip result bits (1's comp)		1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	
#####	###	###	###	###	###	###	###	###	###	###	###	###	###	###	###	###	###	###
Therefore the checksum is		1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	61070

#### SOLUTION - CHECKSUM:

- DEC: 61,070
- Bin: 1110 1110 1000 1110

## Question 2 Prompt

Consider the figure below in which a TCP sender and receiver communicate over a connection in which the sender-to-receiver segments may be lost. The TCP sender sends initial window of five segments at  $t=1,2,3,4,5$ , respectively. Suppose the initial value of the sender-to-receiver sequence number is **110** and all sender-to-receiver segments each contain **594** bytes. The delay between the sender and the receiver is 7 time units, and so the first segment arrives at the receiver at  $t=8$ , and a receiver-to-sender ACK for this segment arrives at the TCP sender at  $t=15$ . As shown in the figure, 2 of the five segments are lost between the sender and the receiver, but none of the receiver-to-sender ACKs are lost.



Answer the following questions: (1 point)

### Question 2.1

Give the sequence number field value in the sender-to-receiver segments in the first set of segments that are sent beginning at  $t=1$ , and the acknowledgement number field value in the receiver-to-sender ACK segments that are sent from the receiver back to the sender beginning at  $t=8$ .

### Solution 2.1

Give the sequence number field value in the sender-to-receiver segments in the first set of segments that are sent beginning at  $t=1$

- Sequence number of the segment sent at  $t=1$ ?
  - The sequence number of the first segment is the starting sequence number, which is 110.
- Sequence number of the segment sent at  $t=2$ ?
  - The sequence number of the second segment is  $= 110 + 594 = 704$ .
- Sequence number of the segment sent at  $t=3$ ?
  - The sequence number of the third segment is  $= 704 + 594 = 1298$ .
- Sequence number of the segment sent at  $t=4$ ?
  - The sequence number of the fourth segment is  $= 1298 + 594 = 1892$ .
- Sequence number of the segment sent at  $t=5$ ?
  - The sequence number of the fifth segment is  $= 1892 + 594 = 2486$ .

and the acknowledgement number field value in the receiver-to-sender ACK segments that are sent from the receiver back to the sender beginning at  $t=8$

- The value of the ACK sent at  $t=8$ ?
  - The ACK value is the sequence number of the next expected segment, which is 704.
- The value of the ACK sent at  $t=9$ ?
  - The ACK value is the sequence number of the next expected segment, which is 1298.
- The value of the ACK sent at  $t=10$ ?
  - Since the segment was lost, so the ACK is never sent.
- The value of the ACK sent at  $t=11$ ?
  - The ACK value is the sequence number of the next expected segment, which is 1298.
  - It's also 1298 because TCP Segment 3 has not reached the TCP Receiver yet (because the signal was lost) and when ACK Segment 3 is received by sender than sender will know that TCP Segment 3 hasn't reached yet. However, the TCP Receiver will receive the TCP Segment 4 because TCP allows out of order segment acceptance.
- The value of the ACK sent at  $t=12$ ?
  - Since the segment was lost, so the ACK is never sent.

### Question 2.2

Next consider the sender-to-receiver segments generated by the sender in response to the ACKS that are received beginning at  $t=15$ . Give the sequence number of the segments transmitted and a brief explanation of why a given segment is transmitted, or not transmitted, on receipt of an ACK.

### Solution 2.2

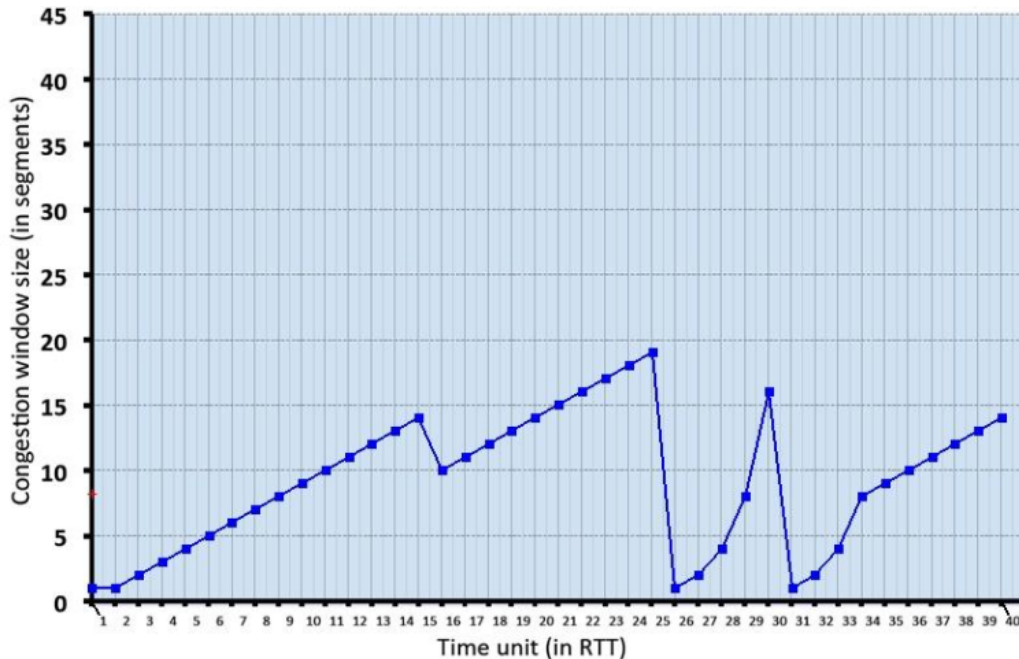
The sequence number of this segment is calculated the same way as the first five segments

- Sequence number of the segment sent at  $t = 15$ ?
  - At  $t=15$ ,  $(2486 + 594 = 3080)$ , the sequence number = 3080 will be sent
- Sequence number of the segment sent at  $t = 16$ ?

- At  $t=16$ ,  $(3080 + 594 = 3674)$ , sequence number = 3674 will be sent.
- Sequence number of the segment sent at  $t = 17$ ?
  - Since there was nothing to send (as we're waiting for ACKs), the answer is NAK.
- Sequence number of the segment sent at  $t = 18$ ?
  - Since there was nothing to send (as we're waiting for ACKs), the answer is NAK.

### Question 3 Prompt: TCP in action: slow start, congestion avoidance, and fast retransmit. (2 points)

Consider the figure below, which plots the evolution of TCP's congestion window at the beginning of each time unit (where the unit of time is equal to the RTT); see Figure 3.53 in the text. In the abstract model for this problem, TCP sends a "flight" of packets of size  $cwnd$  at the beginning of each time unit. The result of sending that flight of packets is that either (i) all packets are ACKed at the end of the time unit, (ii) there is a timeout for the first packet, or (iii) there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP's  $cwnd$  shown below



Consider the evolution of TCP's congestion window in the example above and answer the following questions. The initial value of  $cwnd$  is 1 and the initial value of  $ssthresh$  (shown as a red +) is 8

#### Question 3.1

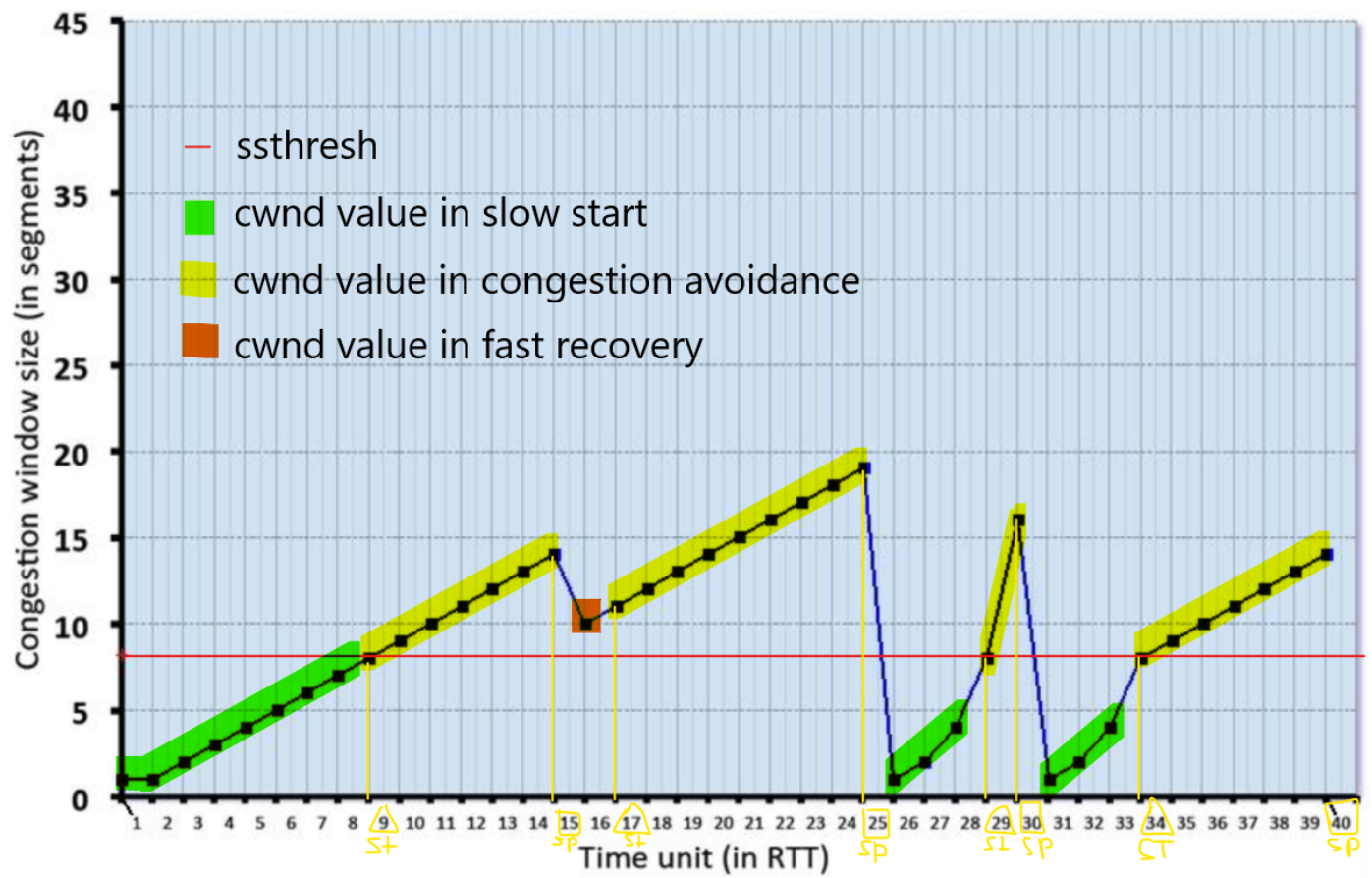
Give the times at which TCP is in

1. slow start
2. congestion avoidance
3. fast recovery

at the start of a time slot, when the flight of packets is sent

#### Solution 3.1

1. **The times where TCP is in slow start are:** [1, 8], [26, 28], [31, 33]
2. **The times where TCP is in congestion avoidance are:** [9, 15], [17, 25], [29, 30], [34, 40]
3. **The times where TCP is in fast recovery:** 16



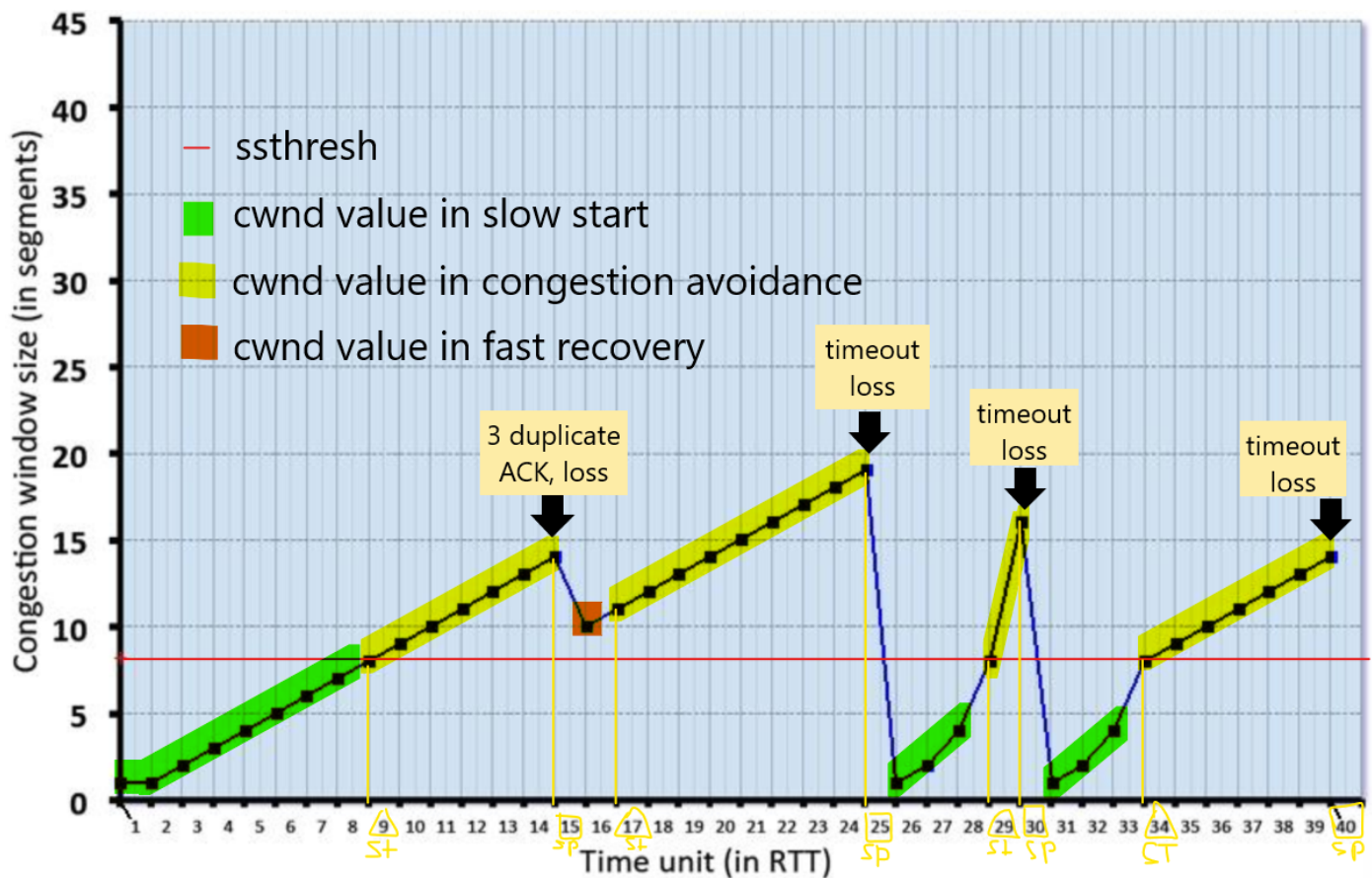
### Question 3.2

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 ACKs.

### Solution 3.2

The times where TCP has a loss by timeout are: 25, 30, 40

The times where TCP has a loss by triple duplicate ACK are: 15



Question 3.3

Give the times at which the value of ssthresh changes, and give the new value of ssthresh.

**Solution 3.3**

**The times where the ssthresh changes are:** 3

**The times where the ssthresh changes are:** 16, 26, 31

Question 4: Computing TCP's RTT and timeout values

Suppose that TCP's current estimated values for the round trip time (estimatedRTT) and deviation in the RTT (DevRTT) are 260 msec and 36 msec, respectively (see Section 3.5.3 for a discussion of these variables).

Suppose that the next three measured values of the RTT are 250 msec, 210 msec, and 400 msec respectively.

**Computing TCP's RTT and timeout values**



Compute TCP's new value of DevRTT, estimatedRTT, and the TCP timeout value after each of these three measured RTT values is obtained. Use the values of  $\alpha = 0.125$ , and  $\beta = 0.25$ . Round your answers to two decimal places after leading zeros.

**Solution 4**

**DevRTT is calculated with the following equation:**

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * |\text{estimatedRTT} - \text{sampleRTT}|$$

**estimatedRTT is calculated with the following equation:**

$$\text{estimatedRTT} = (1-\alpha) * \text{estimatedRTT} + \alpha * \text{sampleRTT}$$

**TCP timeout is calculated with the following equation:**

$$\text{TCP timeout} = \text{estimatedRTT} + (4 * \text{DevRTT})$$

GIVEN:

estimatedRTT\_prev = 260 msec

DevRTT\_prev = 36 msec

SampleRTT1 = 250 msec

SampleRTT2 = 210 msec

SampleRTT3 = 400 msec

alpha (\*\*α\*\*) = 0.125 and beta (\*\*β\*\*) = 0.25

G = 80 msec

1. estimatedRTT after the first RTT?

$$\text{estimatedRTT} = (1-\alpha) * \text{estimatedRTT\_prev} + \alpha * \text{sampleRTT}$$

$$\text{estimatedRTT} = (1-0.125) * 260 + 0.125 * 250$$

$$\text{estimatedRTT} = (0.875) * 260 + 31.25$$

$$\text{estimatedRTT} = 227.5 + 31.25$$

$$\text{estimatedRTT} = 258.75 \text{ msec}$$

**The estimatedRTT for RTT1 is 258.75 msec**

2. RTT Deviation for the the first RTT?

$$\text{DevRTT} = (1-\beta) * \text{DevRTT\_prev} + \beta * |\text{SampleRTT1} - \text{EstimatedRTT}|$$

$$\text{DevRTT} = (1-0.25) * 36 \text{ msec} + 0.25 * |250 \text{ msec} - 258.75 \text{ msec}|$$

$$\text{DevRTT} = 0.75 * 36 \text{ msec} + 0.25 * 8.75 \text{ msec}$$

$$\text{DevRTT} = 27 \text{ msec} + 2.1875 \text{ msec}$$

$$\text{DevRTT} = 29.1875 \text{ msec}$$

**The DevRTT for RTT1 is 29.1875 msec**

3. TCP timeout for the first RTT?

$$\text{TCP timeout} = \text{estimatedRTT} + (4 * \text{DevRTT})$$

$$\text{TCP timeout} = 258.75 \text{ msec} + (4 * 29.1875 \text{ msec})$$

$$\text{TCP timeout} = 258.75 \text{ msec} + 4 * 29.1875 \text{ msec}$$

$$\text{TCP timeout} = 258.75 \text{ msec} + 116.75 \text{ msec}$$

$$\text{TCP timeout} = 375.5 \text{ msec}$$

**The TCP timeout for RTT1 is 375.5 msec**

4. estimatedRTT after the second RTT?

$$\text{estimatedRTT} = (1-\alpha) * \text{estimatedRTT\_prev} + \alpha * \text{sampleRTT2}$$

$$\text{estimatedRTT} = (1-0.125) * 258.75 \text{ msec} + 0.125 * 210 \text{ msec}$$

$$\text{estimatedRTT} = 226.40625 \text{ msec} + 26.25 \text{ msec}$$

$$\text{estimatedRTT} = 252.65625 \text{ msec}$$

**The estimatedRTT for RTT2 is 252.65625 msec**

5. RTT Deviation for the the second RTT?

```
DevRTT = (1-beta) * DevRTT_prev + beta * |estimatedRTT - sampleRTT|  
DevRTT = (1-0.25)* 258.75 msec + 0.25 * |250 msec - 252.65625 msec|  
DevRTT = 194.0625 msec + 0.6640625 msec  
DevRTT = 194.7265625
```

**The DevRTT for RTT2 is 194.7265625 msec**

6. TCP timeout for the second RTT?

```
TCP timeout = EstimatedRTT + 4 * DevRTT  
TCP timeout = 252.65625 msec + 4 * 194.7265625 msec  
TCP timeout = 252.65625 msec + 778.90625 msec  
TCP timeout = 1031.5625 msec
```

**The TCP timeout for RTT2 is 1031.5625 msec**

7. estimatedRTT after the third RTT?

```
estimatedRTT = (1-alpha) * estimatedRTT_prev + alpha * sampleRTT2  
estimatedRTT = (1-0.125) * 252.65625 msec + 0.125 * 400 msec  
estimatedRTT = 221.07421 msec + 50 msec  
estimatedRTT = 271.07421 msec
```

**The estimatedRTT for RTT3 is 271.07421 msec**

8. RTT Deviation for the the third RTT?

```
DevRTT = (1-beta) * DevRTT_prev + beta * |estimatedRTT - sampleRTT|  
DevRTT = (1-0.25) * 194.7265625 msec + 0.25 * |271.07421 msec - 400 msec|  
DevRTT = 146.04492 msec + 0.25 * |271.07421 msec - 400 msec|  
DevRTT = 146.04492 msec + 32.2314475 msec  
DevRTT = 178.27636 msec
```

**The DevRTT for RTT3 is 178.27636 msec**

9. TCP timeout for the third RTT?

```
TCP timeout = estimatedRTT + (4 * DevRTT)  
TCP timeout = 271.07421 msec + (4 * 178.27636 msec)  
TCP timeout = 271.07421 msec + 713.10544 msec  
TCP timeout = 984.17965 msec
```

**The TCP timeout for RTT3 is 984.17965 msec**

## All Solutions

Question 1 : Solution 1

**SOLUTION - CHECKSUM:**

- DEC: 61,070
- Bin: 1110 1110 1000 1110

Question 2 : Solution 2.1

- Sequence number of the segment sent at t=1?
  - The sequence number of the first segment is the starting sequence number, which is 110.



- Sequence number of the segment sent at  $t=2$ ?
  - The sequence number of the second segment is  $= 110 + 594 = 704$ .
- Sequence number of the segment sent at  $t=3$ ?
  - The sequence number of the third segment is  $= 704 + 594 = 1298$ .
- Sequence number of the segment sent at  $t=4$ ?
  - The sequence number of the fourth segment is  $= 1298 + 594 = 1892$ .
- Sequence number of the segment sent at  $t=5$ ?
  - The sequence number of the fifth segment is  $= 1892 + 594 = 2486$ .
- The value of the ACK sent at  $t=8$ ?
  - The ACK value is the sequence number of the next expected segment, which is 704.
- The value of the ACK sent at  $t=9$ ?
  - The ACK value is the sequence number of the next expected segment, which is 1298.
- The value of the ACK sent at  $t=10$ ?
  - Since the segment was lost, so the ACK is never sent.
- The value of the ACK sent at  $t=11$ ?
  - The ACK value is the sequence number of the next expected segment, which is 1298.
  - It's also 1298 because TCP Segment 3 has not reached the TCP Receiver yet (because the signal was lost) and when ACK Segment 3 is received by sender than sender will know that TCP Segment 3 hasn't reached yet. However, the TCP Receiver will receive the TCP Segment 4 because TCP allows out of order segment acceptance.
- The value of the ACK sent at  $t=12$ ?
  - Since the segment was lost, so the ACK is never sent.

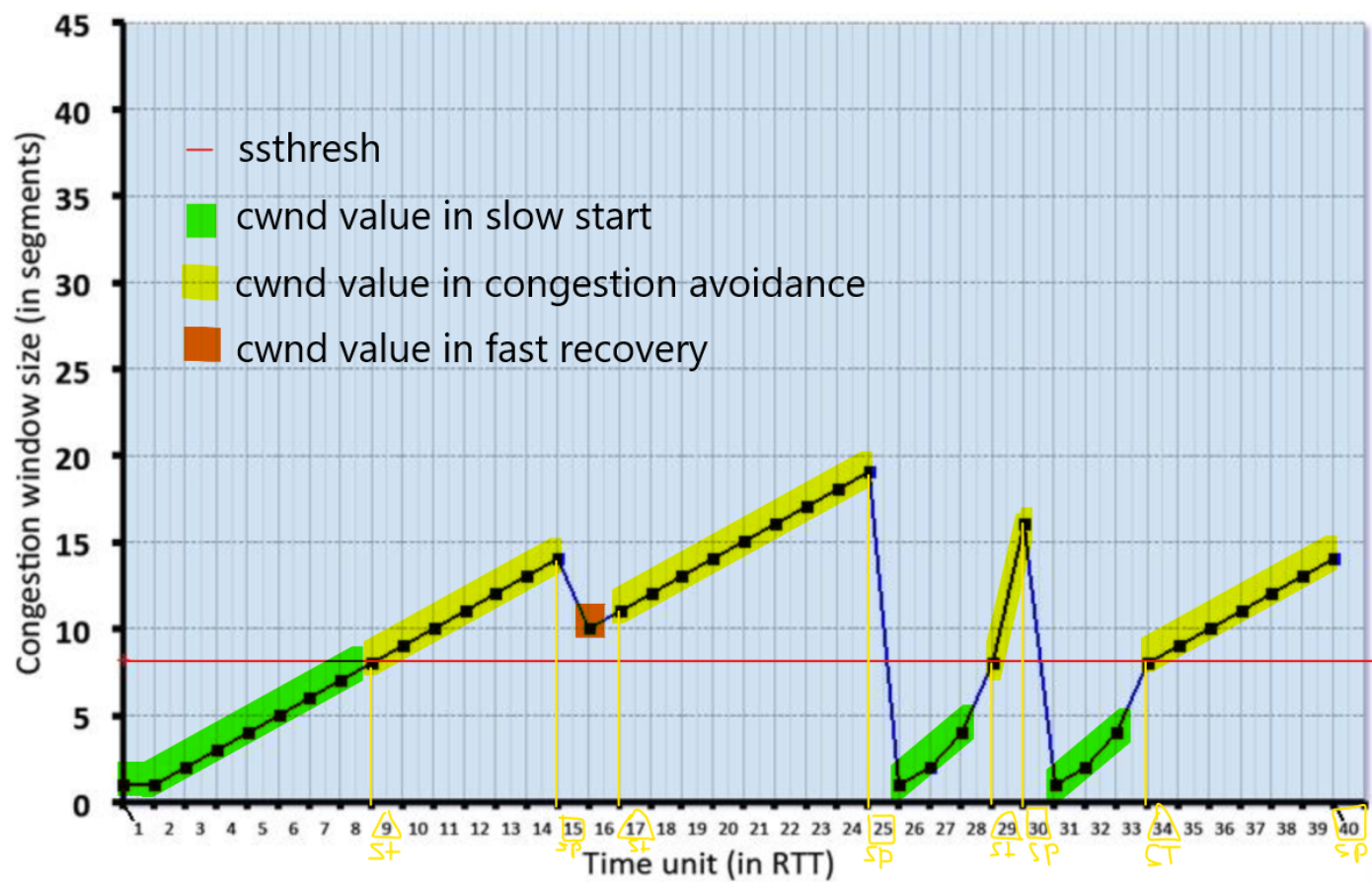
## Question 2 : Solution 2.2

The sequence number of this segment is calculated the same way as the first five segments

- Sequence number of the segment sent at  $t = 15$ ?
  - At  $t=15$ ,  $(2486 + 594 = 3080)$ , the sequence number = 3080 will be sent
- Sequence number of the segment sent at  $t = 16$ ?
  - At  $t=16$ ,  $(3080 + 594 = 3674)$ , sequence number = 3674 will be sent.
- Sequence number of the segment sent at  $t = 17$ ?
  - Since there was nothing to send (as we're waiting for ACKs), the answer is NAK.
- Sequence number of the segment sent at  $t = 18$ ?
  - Since there was nothing to send (as we're waiting for ACKs), the answer is NAK.

## Question 3 : Solution 3.1

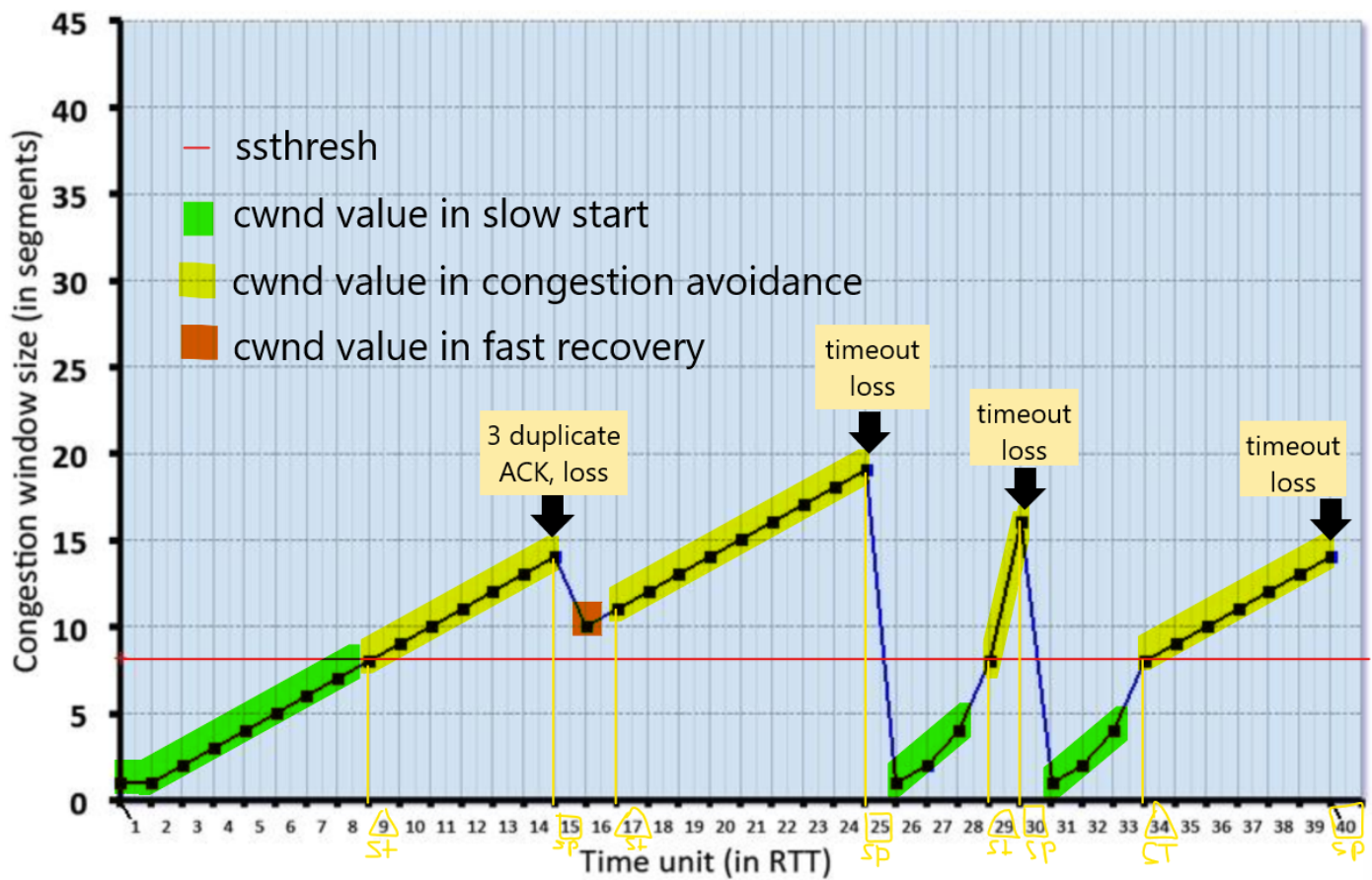
1. **The times where TCP is in slow start are:** [1, 8], [26, 28], [31, 33]
2. **The times where TCP is in congestion avoidance are:** [9, 15], [17, 25], [29, 30], [34, 40]
3. **The times where TCP is in fast recovery:** 16



Question 3 : Solution 3.2

The times where TCP has a loss by timeout are: 25, 30, 40

The times where TCP has a loss by triple duplicate ACK are: 15



Question 3 : Solution 3.3

The times where the ssthresh changes are: 3

The times where the ssthresh changes are: 16, 26, 31

Question 4 : Solution 4

1. estimatedRTT after the first RTT?

$$\text{estimatedRTT} = (1-\alpha) * \text{estimatedRTT\_prev} + \alpha * \text{sampleRTT}$$

$$\text{estimatedRTT} = (1-0.125) * 260 + 0.125 * 250$$

$$\text{estimatedRTT} = (0.875) * 260 + 31.25$$

$$\text{estimatedRTT} = 227.5 + 31.25$$

$$\text{estimatedRTT} = 258.75 \text{ msec}$$

The estimatedRTT for RTT1 is 258.75 msec

2. RTT Deviation for the the first RTT?

$$\text{DevRTT} = (1-\beta) * \text{DevRTT\_prev} + \beta * |\text{SampleRTT1} - \text{EstimatedRTT}|$$

$$\text{DevRTT} = (1-0.25) * 36 \text{ msec} + 0.25 * |250 \text{ msec} - 258.75 \text{ msec}|$$

$$\text{DevRTT} = 0.75 * 36 \text{ msec} + 0.25 * 8.75 \text{ msec}$$

$$\text{DevRTT} = 27 \text{ msec} + 2.1875 \text{ msec}$$

$$\text{DevRTT} = 29.1875 \text{ msec}$$

The DevRTT for RTT1 is 29.1875 msec

3. TCP timeout for the first RTT?

$$\text{TCP timeout} = \text{estimatedRTT} + (4 * \text{DevRTT})$$

$$\text{TCP timeout} = 258.75 \text{ msec} + (4 * 29.1875 \text{ msec})$$

TCP timeout = 258.75 msec + 4 \* 29.1875 msec

TCP timeout = 258.75 msec + 116.75 msec

TCP timeout = 375.5 msec

**The TCP timeout for RTT1 is 375.5 msec**

4. estimatedRTT after the second RTT?

estimatedRTT = (1-alpha) \* estimatedRTT\_prev + alpha \* sampleRTT2

estimatedRTT = (1-0.125) \* 258.75 msec + 0.125 \* 210 msec

estimatedRTT = 226.40625 msec + 26.25 msec

estimatedRTT = 252.65625 msec

**The estimatedRTT for RTT2 is 252.65625 msec**

5. RTT Deviation for the the second RTT?

DevRTT = (1-beta) \* DevRTT\_prev + beta \* |estimatedRTT - sampleRTT|

DevRTT = (1-0.25) \* 258.75 msec + 0.25 \* |250 msec - 252.65625 msec|

DevRTT = 194.0625 msec + 0.6640625 msec

DevRTT = 194.7265625

**The DevRTT for RTT2 is 194.7265625 msec**

6. TCP timeout for the second RTT?

TCP timeout = EstimatedRTT + 4 \* DevRTT

TCP timeout = 252.65625 msec + 4 \* 194.7265625 msec

TCP timeout = 252.65625 msec + 778.90625 msec

TCP timeout = 1031.5625 msec

**The TCP timeout for RTT2 is 1031.5625 msec**

7. estimatedRTT after the third RTT?

estimatedRTT = (1-alpha) \* estimatedRTT\_prev + alpha \* sampleRTT2

estimatedRTT = (1-0.125) \* 252.65625 msec + 0.125 \* 400 msec

estimatedRTT = 221.07421 msec + 50 msec

estimatedRTT = 271.07421 msec

**The estimatedRTT for RTT3 is 271.07421 msec**

8. RTT Deviation for the the third RTT?

DevRTT = (1-beta) \* DevRTT\_prev + beta \* |estimatedRTT - sampleRTT|

DevRTT = (1-0.25) \* 194.7265625 msec + 0.25 \* |271.07421 msec - 400 msec|

DevRTT = 146.04492 msec + 0.25 \* |271.07421 msec - 400 msec|

DevRTT = 146.04492 msec + 32.2314475 msec

DevRTT = 178.27636 msec

**The DevRTT for RTT3 is 178.27636 msec**

9. TCP timeout for the third RTT?

TCP timeout = estimatedRTT + (4 \* DevRTT)

TCP timeout = 271.07421 msec + (4 \* 178.27636 msec)

TCP timeout = 271.07421 msec + 713.10544 msec

TCP timeout = 984.17965 msec

The TCP timeout for RTT3 is 984.17965 msec