Alan Palayil
A20447935

ECE 545

Homework #3

Due Date: 3/5/2023

Problems to be submitted.
Review Questions: R#
Problem Questions: P#

R#3 The source and destination port numbers for the segments travelling from Host B to Host A:
Source port number y and destination port number x.

R#4 The TCP can choke the application's sending rate at times of bottleneck. The UDP does not keep joining state and does not track any of the limits. Even though data transfer by TCP is dependable, some applications do not need dependable TCP data transfer.

R#14
a) False
b) False
c) True
d) False
e) True
f) False
g) False

R#15 a) First segment = 90
Second segment = 110
Data in the first segment = 110 − 90
= 20 //

b) If the first segment is lost but the second segment arrives at B. In the acknowledgement that Host B sends to Host A, then the acknowledgement number will be first segment of sequence number. ie. 90 //

R#17 Suppose there are two TCP connections sharing a bottleneck link with a transmission rate of R bps. Both connections need to send a large file and begin their transmission simultaneously. To ensure a fair allocation of the bandwidth, TCP would ideally like to allocate a transmission rate of R/2 to each connection.

R#18 False. During congestion control in TCP, the slow start threshold is set to half of the current congestion window size, not its previous value.

P#1

a)	A → S	467	23
b)	B → S	513	23
c)	S → A	23	467
d)	S → B	23	513

e) Yes, it is possible.

f) No, it is not possible.

P#3 To calculate the sum of the given 3 bytes:

```
 01010011
 01100110
 --------
 10111001
```

Add the result with the 3rd byte:

```
  10111001
  01110100
 ---------
 100101101      // The extra bit is wrapped.
         1
 ---------
  00101110
```

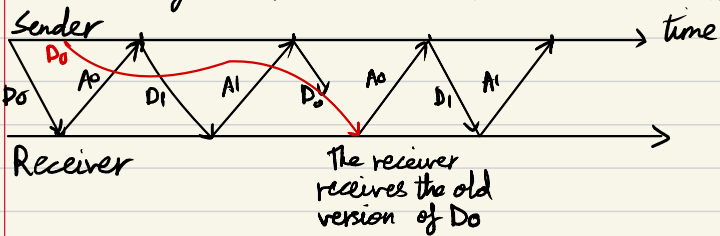
To get the check sum (we invert all bits) = 11010001.

The 1's compliment of (sum) 00101110 is 11010001

UDP uses 1's compliment as it is same as the checksum of the sum.

The checksum is used by the receiver (host) to identify the errors in the segment.

P#13 Here is a diagram illustrating the issue with the rdt 3.0 protocol when messages can be reordered in the network connection:



The message exchange process in this protocol is as follows:

* The sender initially sends data D_0 to the receiver and waits for an acknowledgement.
* If the sender doesn't receive an acknowledgement with a specific period of time, it assumes D_0 was not received and retransmits.
* After sending D_0 for the second time, the sender receives acknowledgement A_0 from the receiver and proceeds to send data D_1.
* The receiver receives D_1 and sends acknowledgement A_1 back to the sender.

However, if the receiver receives the retransmitted D_0 instead of the original D_0 due to reordering in the network, it sends acknowledgement A_0 again assuming that it didn't receive the previous acknowledgement. As a result, the old version of D_0 is acknowledged by the receiver. The sender resends D_1 and waits for A_1, the new version D_0 is replaced by the old version of D_0. This results in the protocol not functioning correctly and the message exchange becoming improper. The data is not transmitted as intended.

P# 15 The round-trip propagation delay between A and B (RTT) = 30 ms.
Transmission rate of the link between A and B (R) = 1 Gbps = 10^9 bps.
The size of the data packet (L) = 1500 bytes = 1500 x 8 bits

$$D_{trans} = \frac{L}{R}$$ (D_{trans} is the time required)

$$= \frac{1500 \times 8}{10^9} \approx 12 \text{ ms} = 0.012 \text{ } \mu s$$

$$\text{Chanel Utilization} = N \times \frac{D_{trans}}{D_{trans} + RTT}$$

$$\frac{98}{100} = N \times \frac{0.012}{0.012 + 30}$$

$$\frac{0.98 \times 30.012}{0.012} = N$$

$$N = 2450.98.$$

∴ To utilize 98% of the channel, the window size should be 2451 packets approximately.

P# 22 a) $N = 4$

Range of sequence number = 1024

Case 1: Consider receiver k and acknowledged all the k-1 packets. The sender's window will be in the range of [k, k+N-1] sequence numbers.
Case 2: If the sender's window will be in the range of [K-N, k-1] sequence numbers. The senders' window will be in the range of [K-N, k-1] sequence numbers. So, the possible sets of sequence numbers inside the sender's window at time t are in the range [k-N, K].

b) The ACK field will be [K-N, k-1]. The sender sent all the K-N packets ACK less than the k-N-1 ACK. So, all possible values of the ACK field in all messages currently range between K-N-1 and K-1.

P#23 For GBN and SR protocols, the largest allowable sender window for GBN is indeed $(K-1)/2$, and for SR it is $K/2$. This is because GBN uses a modulo K sequence number space and can have up to $(K-1)$ unacknowledged packets in flight, but if the window size is greater than $(K-1)/2$, it may result in overlapping sequence numbers and confusion at the receiver. SR, on the other hand, uses a sliding window and can have up to K unacknowledged packets in flight, but if the window size is greater than $K/2$, it may result in a sender timeout due to the receiver's window being too small to hold all the unacknowledged packets.

P#32 Case 1: SampleRRT value is 106 ms:

$$\text{EstimatedRTT}_{106} = (1-\alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$
$$= (1-0.125) \times 100 + 0.125 \times 106$$
$$= 100.75 //$$

$$\text{DevRTT}_{106} = (1 \times \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}|.$$
$$= (1-0.25) \times 5 + 0.25 \times |106 - 100.75|$$
$$= 5.06 //$$

$$\text{TimeoutInterval}_{106} = \text{EstimatedRTT} + 4\,\text{DevRTT}_{106}$$
$$= 100.75 + 4 \times 5.06$$
$$= 120.99 //$$

Case 2: SampleRRT value is 120 ms:

$$\text{EstimatedRTT}_{120} = (1-\alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$
$$= (1-0.125) \times 100.75 + 0.125 \times 120$$
$$= 103.156 //$$

$$\text{DevRTT}_{120} = (1 \times \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}_{120}|$$
$$= (1-0.25) \times 5 + 0.25 \times |120 - 103.156|$$
$$= 8 //$$

$$\text{TimeoutInterval}_{120} = \text{EstimatedRTT} + 4\,\text{DevRTT}_{120}$$
$$= 103.156 + 4 \times 8$$
$$= 135.156 //$$

Case 3: SampleRRT value is 140 ms:

$$\text{EstimatedRTT}_{140} = (1-\alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$
$$= (1-0.125) \times 103.156 + 0.125 \times 140$$
$$= 107.761 \;//$$

$$\text{DevRTT}_{140} = (1 \times \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}_{140}|$$
$$= (1-0.25) \times 5 + 0.25 \times |140 - 107.761|$$
$$= 14.05 \;//$$

$$\text{TimeoutInterval}_{140} = \text{EstimatedRTT} + 4\,\text{DevRTT}_{140}$$
$$= 107.761 + 4 \times 14.05$$
$$= 163.961 \;//$$

Case 4: SampleRRT value is 90 ms:

$$\text{EstimatedRTT}_{90} = (1-\alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$
$$= (1-0.125) \times 107.761 + 0.125 \times 90$$
$$= 105.54 \;//$$

$$\text{DevRTT}_{90} = (1 \times \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}_{90}|$$
$$= (1-0.25) \times 5 + 0.25 \times |90 - 105.54|$$
$$= 6.73 \;//$$

$$\text{TimeoutInterval}_{90} = \text{EstimatedRTT} + 4\,\text{DevRTT}_{90}$$
$$= 105.54 + 4 \times 6.73$$
$$= 132.46 \;//$$

Case 5: SampleRRT value is 115 ms:

$$\text{EstimatedRTT}_{115} = (1-\alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$
$$= (1-0.125) \times 105.54 + 0.125 \times 115$$
$$= 106.722 \;//$$

$$\text{DevRTT}_{115} = (1 \times \beta) \times \text{DevRTT} + \beta \times |\text{SampleRTT} - \text{EstimatedRTT}_{115}|$$
$$= (1-0.25) \times 5 + 0.25 \times |115 - 106.722|$$
$$= 7.1 \;//$$

$$\text{TimeoutInterval}_{115} = \text{EstimatedRTT} + 4\,\text{DevRTT}_{115}$$
$$= 106.722 + 4 \times 7.1$$
$$= 135.122 \;//$$

P#40 a) If TCP slow start is operating, then the intervals of time 1 to 6 and 23 to 26.

b) If TCP congestion avoidance is operating, then the intervals of time 6 to 23.

c) After the 16th transmission round, then the segment loss detected by a triple duplicate ACK.

d) After the 22nd transmission round, then the segment loss detected by timeout.

e) The initial value of ssthresh at the first transmission round 32.

f) The value of ssthresh at the 18th transmission round 21.

g) The value of ssthresh at the 24th transmission round 13.

h) The transmission round is the 70th segment sent is 7.

i) A triple duplicate ACK received after the 26th round indicates a packet loss, and the corresponding value is 4.

j) Assuming the use of TCP Tahoe, if triple duplicate ACKs are received at the 16th round, then the ssthresh value and congestion window size at the 19th round are 1, and the transmission round is 21.

k) Assuming the use of TCP Tahoe, if a timeout event occurs at the 22nd round, then the number of packets sent out from the 17th round until the 22nd round is 52.