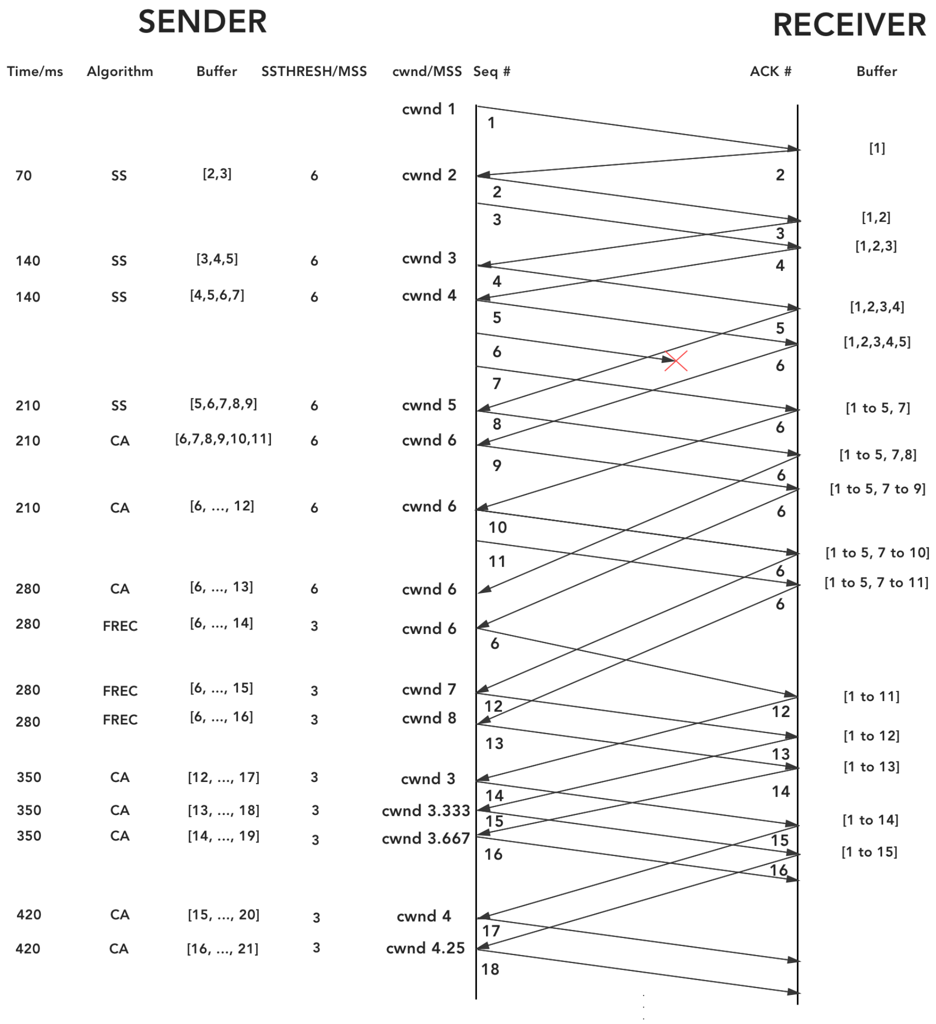
CS 118 - Homework 5

Kuruvilla Cherian, Jahan
UID: 104436427
Section 1A

November 9, 2016

Problem 1

The time of receiving the 15th ACK is essentially 6 RTT's which comes out to 420ms as shown below.



Time/ms	Algorithm	Buffer	SSTHRESH/MSS	cwnd/MSS
				cwnd 1
70	SS	[2,3]	6	cwnd 2
140	SS	[3,4,5]	6	cwnd 3
140	SS	[4,5,6,7]	6	cwnd 4
210	SS	[5,6,7,8,9]	6	cwnd 5
210	CA	[6,7,8,9,10,11]	6	cwnd 6
210	CA	[6, ..., 12]	6	cwnd 6
280	CA	[6, ..., 13]	6	cwnd 6
280	FREC	[6, ..., 14]	3	cwnd 6
280	FREC	[6, ..., 15]	3	cwnd 7
280	FREC	[6, ..., 16]	3	cwnd 8
350	CA	[12, ..., 17]	3	cwnd 3
350	CA	[13, ..., 18]	3	cwnd 3.333
350	CA	[14, ..., 19]	3	cwnd 3.667
420	CA	[15, ..., 20]	3	cwnd 4
420	CA	[16, ..., 21]	3	cwnd 4.25

Figure 1: TCP Diagram showing the different states and the associated values.

Problem 2

a

For a floor of 100 developers the subnet is ***131.179.45.128/25***, thus giving us reign over 7 bits (128) worth of IP addresses.

b

Since we don't want to overlap with the IP addresses given in (a), we can put the subnet addresses under ***131.179.45.0/26*** which would give us reign over 6 bits (64) worth of IP addresses. We choose a subnet mask of 26 1's because it prevents a clash with the availability of IP addresses.

c

Without causing any overlap with either of the ranges given in (a) and (b) we give the labs the choice of 64 IP addresses (6 bits) by using the subnet ***131.179.45.64/26***.

Problem 3

a

Network Mask: *255.255.255.0*
Network Address: *131.179.196.0*
Broadcast Address: *131.179.196.255*
Number of Endpoint addresses: Excluding the network and broadcast addresses (which would be the only special ones in this scenario) we have **254** endpoint addresses left over.

b

Network Mask: *255.255.255.252*
Network Address: *169.232.34.48*
Broadcast Address: *169.232.34.51*
Number of Endpoint addresses: Excluding the network and broadcast addresses (which would be the only special ones in this scenario) we have **2** endpoint addresses left over.

c

Network Mask: *255.255.248.0*
Network Address: *196.22.136.0*
Broadcast Address: *196.22.143.255*
Number of Endpoint addresses: Excluding the network and broadcast addresses (which would be the only special ones in this scenario) we have $2^{32-21} - 2 = 2^{11} - 2 = \mathbf{2046}$ endpoint addresses left over.

d

Network Mask: *255.255.224.0*
Network Address: *93.181.192.0*
Broadcast Address: *93.181.223.255*
Number of Endpoint addresses: Excluding the network and broadcast addresses (which would be the only special ones in this scenario) we have $2^{32-19} - 2 = 2^{13} - 2 = \mathbf{8190}$ endpoint addresses left over.

e

Network Mask: *255.192.0.0*
Network Address: *10.128.0.0*
Broadcast Address: *10.191.255.255*

Number of Endpoint addresses: Since this is defining a private network, all the endpoints are private addresses. Thus excluding the broadcast and network addresses we have a total of $2^{32-10} - 2 = 2^{22} - 2 =$ **4194302** endpoint addresses left over.

Problem 4

a

For Prefix Match: 11100000, Destination IP Addresses range from **224.0.0.0 to 224.255.255.255** (11100000 00000000 00000000 00000000 to 11100000 11111111 11111111 11111111).

For Prefix Match: 11100001 00000000, Destination IP Addresses range from **225.0.0.0 to 225.0.255.255** (11100001 00000000 00000000 00000000 to 11100001 00000000 11111111 11111111).

For Prefix Match: 11100001, Destination IP Addresses range from **225.0.0.0 to 225.255.255.255** (11100001 00000000 00000000 00000000 to 11100001 11111111 11111111 11111111).

For Prefix Match: otherwise, Destination IP Addresses range from **0.0.0.0 to 223.255.255.255 and 226.0.0.0 to 255.255.255.255** (00000000 00000000 00000000 00000000 to 11011111 11111111 11111111 11111111 and 11100010 00000000 00000000 00000000 to 11111111 11111111 11111111 11111111).

b

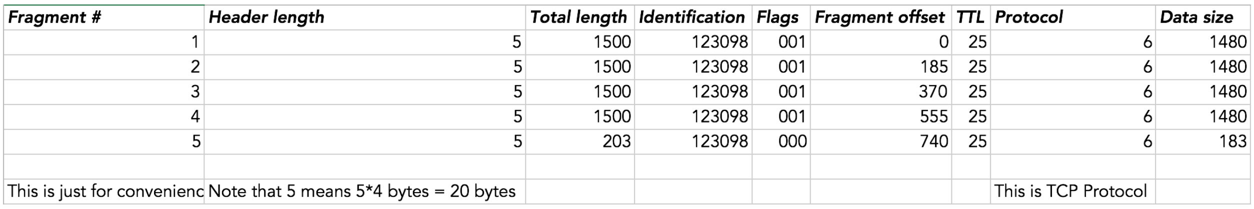
We always match with the longest prefix available, and so arranging the destination addresses as given in the question we have the following:
1.) 11001000 10010001 01010001 01010101 will match with *Link Interface 3* since there is no prefix match as per the given table.
2.) 11100001 00000000 11000011 00111100 will match with *Link Interface 1* since the longest prefix match is 11100001 00000000 as per Link Interface 1.
3.) 11100001 10000000 00010001 01110111 will match with *Link Interface 2* since the longest prefix match is 11100001 as per Link Interface 2.
4.) 11100000 10111011 10100000 00000001 will match with *Link Interface 0* since the longest prefix match is 11100000 as per Link Interface 0.

Problem 5

a

Since we have a total of 6103 bytes within the IP packet, and an MTU of 1500 bytes, then we have to fragment the packet into $\lceil \frac{6103}{1500} \rceil = 5$ fragments.

b



Fragment #	*Header length*	*Total length*	*Identification*	*Flags*	*Fragment offset*	*TTL*	*Protocol*	*Data size*
1	5	1500	123098	001	0	25	6	1480
2	5	1500	123098	001	185	25	6	1480
3	5	1500	123098	001	370	25	6	1480
4	5	1500	123098	001	555	25	6	1480
5	5	203	123098	000	740	25	6	183
This is just for convenienc	Note that 5 means 5*4 bytes = 20 bytes						This is TCP Protocol	

Figure 2: Completed table representing the IP header information for each of the five fragments.