Bryan Arnold

11/26/18

CSE 3300

Homework 3

1a) The max window size in segments can be found with the equation that contains the RTT, link, and TCP segment size. Here is the equation: (WindowSize * 1500 bytes) / RTT = 10 Mbps / 8. WindowSize * 1500 bytes = (1.25 * 0.1*10^6), WindowSize * 1500 bytes = 187500, WindowSize = 125. The maximum window size in segments is 125 segments.

b) Since this connection works under congestion avoidance, we need to find the average window size by using 2 * WindowSize / 3. 2 * 125 segments / 3 = 94 average window size. To get the throughput, we calculate the following: AverageWindowSize * 1500 bytes * 8 / 0.15. 94 segments * 1500 bytes * 8 / 0.15 = 7.52*10^6 bps.

c) For this, you need to take the average window size and divide it in half, but since the RTT effects the segments sent, you also need to multiply by the RTT delay in seconds. 94 segments / 2 * 0.15 seconds = 7.05 seconds to reach maximum window again after a packet loss.

2a) For a forwarding table, all we need is the destination of the datagram and the interface involved in getting the datagram to that destination. In this case, the destination is H3 and the interface we are using is 3, so the table looks like the following:

Destination	Interface

H3	3

b) There can be no forwarding table for this network, since a forwarding table is based on the interface and destination of the datagram. In this case, the interfaces for H1->H3 and H2->H3 are different, but the destination addresses are the same, H3. This is not possible in forwarding rules, but if it were, the table would look like the following:

Destination	Interface
H3	3
H3	4

c) For a virtual circuit, the forwarding table is more based upon the incoming interface, the VC number, and the corresponding outgoing interface along with another VC number. The VC numbers for outgoing and incoming must all be different values for both flows. Here is an example table:

Incoming Interface	Incoming VC Number	Outgoing Interface	Outgoing VC Number
1	10	3	32
2	51	4	28

3) First, we need to find out the number of addresses that are needed (the size) given the number of interfaces each subnet must support. Subnet 1 needs to support at least 60 interfaces, so we round up to the nearest power of 2, which is 64, which means subnet 1 needs 64 addresses.

Subnet 2 needs to support at least 90 interfaces, so we round up to the nearest power of 2, which means subnet 2 needs 128 addresses. Lastly, subnet 3 needs to support at least 12 interfaces, so we round up to the nearest power of 2, which means subnet 3 needs 16 addresses. Given these sizes of the addresses, here is a sample address for each subnet:

Subnet 1: 223.1.17.0/26 Subnet 2: 223.1.17.128/25 Subnet 3: 223.1.17.64/28

4) The number of fragments would be the size of the datagram in bytes divided by total number of MTU bytes. Since the IP address in a datagram is 20 bytes, the MTU = 700 – 20 = 680 bytes, and the total datagram size available would be 2400 – 20 = 2380. So, the number of fragments that are generated is 2380 / 680 = 3.5, so round up to 4.

The fields in the datagrams that are relevant for fragmentation include the ID number, the size of the datagram, the offsets of each datagram, as well as a flag. Each datagram shares the same fragmentation ID number, 422. The sizes of the first 3 datagrams will be 680 + 20 (IP header) = 700 bytes, while the last one is only 340 + 20 = 360 bytes. The last datagram is smaller since most of the data has already been handled by the proceeding datagrams. The first 3 datagrams also have their flag = 1, while the last one flag = 0. Finally, the offsets of each datagram are the following in their respective orders: 0, 85, 170, 255.

5) Here is the table using the shortest-path algorithm on the network:

Step	N’	D(t), p(t)	D(u), p(u)	D(v), p(v)	D(w), p(w)	D(y), p(y)	D(z), p(z)
0	x	∞	∞	3, x	6, x	6, y	8, z
1	xv	7, v	6, v	3, x	6, x	6, x	8, x

2	xvu	7, v	6, v	3, x	6, x	6, x	8, x
3	xvuw	7, v	6, v	3, x	6, x	6, x	8, x
4	xvuwy	7, v	6, v	3, x	6, x	6, x	8, x
5	xvuwyt	7, v	6, v	3, x	6, x	6, x	8, x
6	xvuwytz	7, v	6, v	3, x	6, x	6, x	8, x