CSC358 Winter 2018 Problem Set 2

Name: Akhil Gupta
SN: 1000357071
Utorid: guptaak2

Q1.
1. Provider Q's BGP speakers could receive upto 3 routes to Provider P. These routes would be along inter-provider links 1, 2 and 3.
2. Traffic from Host A to Host B would be along link 1, and traffic from Host B to Host A would be along link 2.

Q2.
1. The routing inefficiency for the organization's inbound traffic is that all traffic enters the organization from a single point source (due to the fact that they only have a single IP network address) even if shorter alternative routes do exist.
2. The organization might solve this problem by entering all the highest-level geographical blocks which allows the organization to route outbound traffic to the geographically closest exit to the destination.
3. For my approach to work for inbound traffic, the organization could divide itself into subnets which would be based on their geographical location. For the outside world, they would have to accept routing entries for each subnet.
4. If the organization changes its addressing to separate geographical address for each office, it will each need an internal router to have entries for the internal routes to all other internal IP networks.

Q3. The sequence number field in the TCP header may still wrap around from $2^{32} - 1$ to 0 because when a new connection is created, the sequence number is set to some random number; if however, that random number is set close to FF:FF:FF:FF then without wrapping, we would be in trouble.

Q4.
1. The sender has lost $1100ms$. The sender waits $300ms$ to detect the third duplicate ACK from the receiver and then a RTT of $800ms$ as the sender waits for the ACK of the retransmitted segment.
2. The sender has lost $1100ms - 400ms = 700ms$. The sender uses the continued arrival of each duplicate ACK to slide the window forward one segment, so we could've sent it 4 times during that interval.

Q5. (We assume that TCP AIMD congestion control protocol is being used)
To verify whether a client was not using slow start at all, we could determine the actual number of bytes remaining by checking the SEQ and ACK values. Using this, we can check whether the host used slow start on startup if only 1 more packet is remaining than the total numbers of ACKs received. We should be able to see the congestion window reduce by half if any packet is retransmitted due to the timeout.

Q6.
1. To show that R1 cannot become congested, we look at the left and right links of R1. We only examine the left link, as it is symmetrical to the right link. For R1 to become congested, all hosts from the left link attempt to communicate to the respective hosts on the right side of the link. The traffic generated by the left side of the link to a host in the right side of the link is 1MBps * 4 = 4MBps. Similarly to prove the other direction, hosts from the right side of the link generate 4MBps as well. As a result, the links are utilized to their full capacity (2 * 2MBps = 4MBps)

but since that capacity is equal to the demanding throughput, R1 cannot be congested. Same argument can be made for hosts from the right side of the link.

2. For notation, let's denote the hosts from left to right by H1-H8, and the routers R1 $\rightarrow$ (R2 $\rightarrow$ (R4, R5), R3 $\rightarrow$ (R6, R7)). The ”$\rightarrow$“ denotes children of the router. Now, let's find a traffic pattern that congests R7. Suppose hosts H5-H7 generate traffic towards H8. The total throughput equals 1MBps * 3 = 3MBps. These leads to congestion for the between router R7 and host H8 since it exceeds the maximum capacity of the link which is 1MBps. Hence, we can apply the above argument to any router R (except R1) that congests that router alone.

Q7. Possible source and destination port numbers for:

1. Segments sent from A to S: Source port = 10985, Destination port = 25
2. Segments sent from B to S: Source port = 10425, Destination port = 25
3. Segments sent from S to A: Source port = 25, Destination port = 10985
4. Segments sent from S to B: Source port = 25, Destination port = 10425
5. Yes, the source port number in the segments from A to S and B to S can be the same. Since we know that Telnet is reliable and connection-oriented, it uses a 4-tuple combination (source IP, source port, dest IP, dest port), so the server will use a combination of source port and source IP to determine the actual source.
6. If A and B are the same host, they are different client applications running on the same machine. So, when they initiate a Telnet session with Server S, they must have different source port numbers.

Q8.

	0	1	0	1	0	0	1	1	first word
+	0	1	1	0	0	1	1	0	second word
	1	0	1	1	1	0	0	1	first sum
+	0	1	1	1	0	1	0	0	third word
	0	0	1	0	1	1	1	0	sum wraps around

Checksum is 1's complement of the sum: 1 1 0 1 0 0 0 1.
By taking the 1's complement of the sum, we can check for errors by adding up all 4 words (3 original and 1 checksum). With the 1's complement scheme, the receiver can detect errors if the sum of the original sum and checksum contains a 0. No, all 1-bit errors will be detected. But, a 2-bit error could be undetected, for example, if the last bit of the first word changes to a 0 and the last bit of the second word changes to a 1.

Q9.

1. Using TCP with AIMD congestion control, it takes 1 RTT to increase the congestion window to 7 MSS. 2 RTT's to increase the cwnd to 8 MSS. 3 RTT's to increase to 9 MSS. Following the pattern, we can conclude that it will take 6 RTT's for the cwnd to increase from 6 MSS to 12 MSS.
2. Connection up through time = 6 RTT. Average throughput = $\frac{MSS}{RTT}$. In the first RTT, 6 MSS were sent. 2nd RTT, 7 MSS were sent. 3rd RTT, 8 MSS were sent. Following the pattern, in the 6th RTT, 11 MSS were sent. Thus, upto 6th RTT, average throughput = $\frac{6+7+8+9+10+11}{6} = \frac{51MSS}{6RTT} = 8.5MSS/RTT$.

Q10

1. eBGP
2. iBGP
3. eBGP
4. iBGP

Q11. Shortest-path from x to all nodes (used alphabetical tie-breaking for equal distances)

step	visited	dist(t)	dist(u)	dist(v)	dist(w)	dist(y)	dist(z)
0	{x}	∞	∞	3	6	6	8
1	{xv}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8
2	{xvu}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8
3	{xvuw}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8
4	{xvuwy}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8
5	{xvuwyt}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8
6	{xvuwytz}	7 (x $\rightarrow$ v)	6 (x $\rightarrow$ v)	3	6	6	8

Q12. Fragmentation:

ID	Length	Frag Flags	Frag Offset
422	680 bytes	flag = 1	0
422	680 bytes	flag = 1	85
422	680 bytes	flag = 1	170
422	260 bytes	flag = 0	255

4 fragments are generated. The maximum data that we can send in each fragment is MTU − 20 bytes for IP header = 680 bytes.