EECE 7374 – Fall 2023
Homework #3 Solution

Chapter 4

1) **TCP congestion control**
   a) It takes 1 RTT to increase cwnd to 2 MSS; 2 RTTs to increase to 3 MSS; 3 RTTs to increase to 4 MSS; 4 RTTs to increase to 5 MSS; 5 RTTs to increase to 6 MSS.
   b) In the first RTT 1 MSS was sent; in the second RTT 2 MSS were sent; in the third RTT 3 MSS were sent; in the fourth RTT 4 MSS were sent; in the fifth RTT 5 MSS were sent. Thus, up to time 5 RTT, 1+2+3+4+5 = 15 MSS were sent (and acknowledged). Thus, we can say that the average throughput up to time 5 RTT was (15 MSS)/(5 RTT) = 3 MSS/RTT.

2) **TCP throughput**
   a) Let W denote the max window size measured in segments. Then, W*MSS/RTT = 15 Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W*1500*8/0.15=15*10^6, then W is about 188 (ceiling of 187.5) segments.
   b) As congestion window size varies from W/2 to W, then the average window size is 0.75W=141 segments. Average throughput is 141*1500*8/0.15=11.28 Mbps.
   c) 188/2 *0.15= 14.1 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. Recall the window size increases by one in each RTT.
   d) W = 187500 segments, average window size = 140725 segments, average throughput = 11.258 Gbps. After recovering from a packet loss, it would take 234 minutes to reach its maximum window again – a very long time! In order to speed up the window increase process, we can increase the window size by a much larger value, instead of increasing the window size by only one in each RTT. Some protocols are proposed to solve this problem, such as ScalableTCP or HighSpeed TCP (any answer along these lines is correct, you don't have to specifically mention ScalableTCP or High Speed TCP or TCP Cubic).

Chapter 4

3) **Longest prefix matching**
   a)

Prefix	Outgoing link interface
0000000	0
01011101	1
0111011	2

b)

Prefix	Outgoing link interface
00000000	0
00000001 0	0
00000001 10000000 00000000 00000000	0
01010101	1
0101011	2

4) **Subnets** – Giving each department a single subnet, the nominal subnet sizes are 2^7, 2^6, 2^5, 2^5 respectively; we obtain these by rounding up to the nearest power of 2. A possible arrangement of subnet numbers is as follows.

A	222.3.4.0/25
B	222.3.4.128/26
C	222.3.4.192/27
D	222.3.4.224/27

5) **OpenFlow**

S2 Flow Table	
Match	**Action**
Ingress Port = 1; IP Src = 10.3.*.*; IP Dst = 10.1.*.*	Forward (2)
Ingress Port = 2; IP Src = 10.1.*.*; IP Dst = 10.3.*.*	Forward (1)
Ingress Port = 1; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 2; IP Dst = 10.2.0.3	Forward (3)
Ingress Port = 1; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 2; IP Dst = 10.2.0.4	Forward (4)
Ingress Port = 4	Forward (3)
Ingress Port = 3	Forward (4)

6) **OpenFlow**

S2 Flow Table	
Match	**Action**
IP Src = 10.1.0.1; IP Dst = 10.2.0.3	Forward (3)
IP Src = 10.1.0.1; IP Dst = 10.2.0.4	Forward (4)
IP Src = 10.3.0.6; IP Dst = 10.2.0.3	Forward (3)
IP Src = 10.3.0.6; IP Dst = 10.2.0.4	Forward (4)

S2 Flow Table	
Match	**Action**
IP Src =.*.*.*.*; IP Dst = 10.2.0.3; IP prot = TCP	Forward (3)
IP Src =.*.*.*.*; IP Dst = 10.2.0.4; IP prot = TCP	Forward (4)

S2 Flow Table	
Match	**Action**
IP Src =.*.*.*.*; IP Dst = 10.2.0.3	Forward (3)

S2 Flow Table	
Match	**Action**
IP Src = 10.1.0.1; IP Dst = 10.2.0.3; IP prot = UDP	Forward (3)