

Assignment #1
Matthew Langlois - 7731813
Oct 12

Part One

1. Since we know the address is 193.1.1.0 which is class C then the subnet mask 255.255.255.0 is applied. This means we can modify the last octet in the mask to modify the network which gives us 256 possible addresses. By taking 2 bits of the last octet we can apply a 193.1.1.0/26 mask (255.255.255.192) to get 4 separate networks (.0, .64, .128, .192).

	Router	Device 1	Device 2
Network 1	193.1.1.0	193.1.1.1	193.1.1.2
Network 2	193.1.1.64	193.1.1.65	193.1.1.66
Network 3	193.1.1.128	193.1.1.129	193.1.1.130
Network 4	193.1.1.192	193.1.1.193	193.1.1.194

2. In each of these questions the mask is applied to the first n-bits of the ip address to determine the subnet. Once the subnet is determined we can look up the address in the routing table to determine which device to route the packet to next.
 - (a) Taking the first 22 bits of 135.46.63.10 you get 135.46.60.0. Thus this matches Interface 1 which is where the packet will be forwarded to.
 - (b) Taking the first 22 bits of 135.46.57.14 you get 135.45.56.0. Thus this matches Interface 0 which is where the packet will be forwarded to.
 - (c) Taking the first 23 bits of 135.46.52.2 you get 135.45.52.0. This does not match any of the subnets in the routing table thus the packet will be forwarded to the default gateway which in this case is Router 2.
 - (d) Taking the first 23 bits of 192.53.40.7 you get 192.53.40.0. Thus this matches Router 1 which is where the packet will be forwarded to.
 - (e) Taking the first 23 bits of 192.53.56.7 you get 192.53.56.0. This does not match any of the subnets in the routing table thus the packet will be forwarded to the default gateway which in this case is Router 2

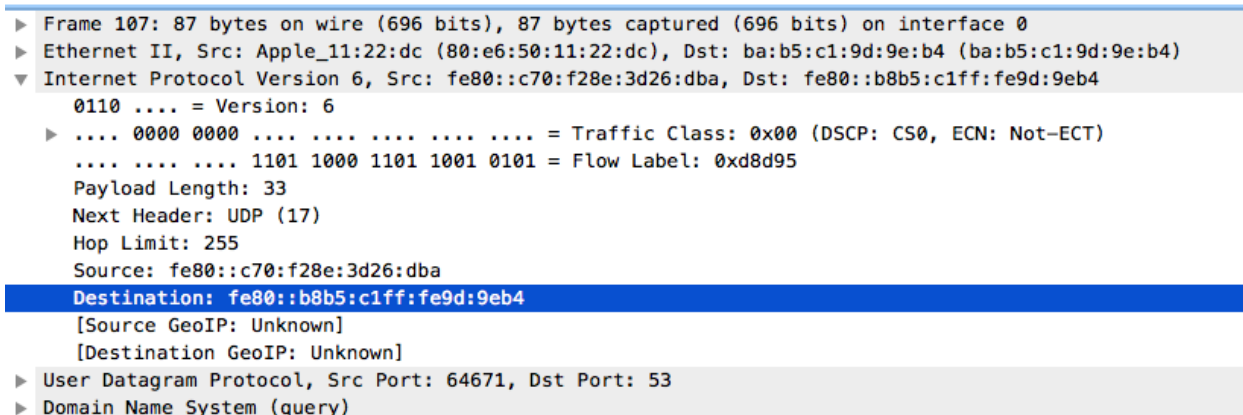
3. Dijkstra's Algorithm - Each iteration the values in the table are updated if there is a shorter path found to the node which already exists in the cloud.

Iter	B	C	D	E	F	G	H
1	2	∞	∞	∞	∞	6	∞
2	2	9	∞	4	∞	6	10
3	2	9	∞	4	6	5	10
4	2	9	10	4	6	5	8

4. a) The window size is constantly increased by 1, until the maximum size is hit. In this case we need to transmit 2000 packets of 1KB with the max window size of 500KB. To transmit the packets the window will increase by 1, until all 2000 packets are sent ($1 + 2 + 3 + \dots + n = 2000$) or $\frac{n(n+1)}{2} = 2000$
Solving for only the positive side of n (since it is a quadratic function) we get 63. Thus the congestion window size of 500KB is never reached so no decrease occurs. Therefore the total time is $63 \cdot 3.3 = 207$ which is 207 ms.
- b) The window size, starting at 1KB, is doubled, until the maximum size is hit. To fit all 2000 packets we will need to make 11 trips, which is less than the 500KB maximum window size: $1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 + 1024 = 2047$
Thus the total time is $11 \cdot 3.3 = 36.3$ which is 36.3ms.

Part Two

- 1) The IP address was found in the IPV6 frame which come right after the ethernet 2 header and before the UDP datagram. The destination address was `fe80::b8b5:c1ff:fe9d:9eb4`. The destination address can be seen in the following image:



```

0000  ba b5 c1 9d 9e b4 80 e6 50 11 22 dc 86 dd 60 0d ..... P."...`
0010  8d 95 00 21 11 ff fe 80 00 00 00 00 00 00 0c 70 ...!.....p
0020  f2 8e 3d 26 0d ba fe 80 00 00 00 00 00 00 b8 b5 ..=&.....
0030  c1 ff fe 9d 9e b4 fc 9f 00 35 00 21 51 98 76 bf ..... .5.!Q.v.
0040  01 00 00 01 00 00 00 00 00 00 03 62 62 63 03 63 ..... ...bbc.c
0050  6f 6d 00 00 01 00 01 ..... om.....

```

- 2) The two types of messages being exchanged are ICMP request (Echo ping request) and ICMP response (Echo ping response). The ping request is sending a request to the server with a TTL of 64. The server then responds with a TTL of 49. Therefore the server is reachable after 15 hops. The messages being exchanged can be seen in the image below:

No.	Time	Source	Destination	Protocol	Length	Info
107	5.990811	fe80::c70:f28e:3d2...	fe80::b8b5:c1ff:fe...	DNS	87	Standard query 0x76bf A bbc.com
108	6.016494	fe80::b8b5:c1ff:fe...	fe80::c70:f28e:3d2...	DNS	240	Standard query response 0x76bf A bbc.com ...
109	6.017102	192.168.1.188	212.58.246.79	ICMP	98	Echo (ping) request id=0x821a, seq=0/0, ...
112	6.120310	212.58.246.79	192.168.1.188	ICMP	98	Echo (ping) reply id=0x821a, seq=0/0, ...
→ 128	7.020045	192.168.1.188	212.58.246.79	ICMP	98	Echo (ping) request id=0x821a, seq=1/256...
← 131	7.124623	212.58.246.79	192.168.1.188	ICMP	98	Echo (ping) reply id=0x821a, seq=1/256...

- 3) The IP address of bbc.com is 212.58.246.79 as seen in the ping request. Refer to the image in Q4 the line "destination" is where the ping request is being routed to.
- 4) The IP address of my personal computer is 192.168.1.188 as seen in the ping request. Refer to the image... the line "source" is where the ping request is being sent from (my computer).

```

▶ Ethernet II, Src: Apple_11:22:dc (80:e6:50:11:22:dc), Dst: ba:b5:c1:9d:9e:b4 (ba:b5:c1:9d:9e:b4)
▼ Internet Protocol Version 4, Src: 192.168.1.188, Dst: 212.58.246.79
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xbdfa (48634)
    ▶ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x2fc0 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.188
    Destination: 212.58.246.79
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
    ▶ Internet Control Message Protocol
0000  ba b5 c1 9d 9e b4 80 e6 50 11 22 dc 08 00 45 00  .... P."...E.
0010  00 54 bd fa 00 00 40 01 2f c0 c0 a8 01 bc d4 3a  .T...@. /.....:
0020  f6 4f 08 00 d0 ec 82 1a 00 01 59 dd 81 ba 00 0a  .O..... ..Y.....
0030  de 52 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .R..... .....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ..... .. !"#$$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,- ./012345
0060  36 37 67

```