Robert Crimi
Problem Set 2

1

a) 10.72.0.255/255.255.254.0 is not a valid address. It should be changed to 10.72.0.255/23

b) 10.72.0.0/16

5 networks 4096 hosts:
10.72.240.0/20
10.72.224.0/20
10.72.192.0/20
10.72.160.0/20
10.72.128.0/20

8 networks 1024 hosts
10.72.252.0/22
10.72.248.0/22
10.72.240.0/22
10.72.224.0/22
10.72.192.0/22
10.72.176.0/22
10.72.160.0/22
10.72.128.0/22

10 networks 64 hosts
10.72.255.192/26
10.72.255.128/26
10.72.255.0/26
10.72.254.0/26
10.72.252.0/26
10.72.248.0/26
10.72.240.0/26
10.72.224.0/26
10.72.192.0/26
10.72.128.0/26

c) 192.168.2/23 and 192.168.3/23 are the same as the subnet mask ends at the 2 bit. Ex 0000001|0 for 192.168.2 and 0000001|1 for 192.168.3

23 bit 23 bit

2

212.1.1/23

A: 150 hosts

B: 120 hosts

C: 55 hosts

D: 35 hosts

256

a) A: 212.1.0.0/24

B: 212.1.0.128/25

C: 212.1.0.192/26

D: 212.1.0.128/26

b) They do not need to anything as I organized department D to hold $2^6 = 64$ addresses

3

a)

	A	B	C	D	E	F
A	0	5	2	∞	∞	∞
B	5	0	∞	2	∞	∞
C	2	∞	0	3	2	∞
D	∞	2	3	0	∞	4
E	∞	∞	2	∞	0	4
F	∞	∞	∞	4	4	0

b)

	A	B	C	D	E	F
A	0	5	2	5	4	∞
B	5	0	5	2	∞	6
C	2	5	0	3	2	6
D	5	2	3	0	5	4
E	4	∞	2	5	0	4
F	∞	6	6	4	4	0

c)

	A	B	C	D	E	F
A	0	5	2	5	4	8
B	5	0	5	2	7	6
C	2	5	0	3	2	6
D	5	2	3	0	5	4
E	4	7	2	5	0	4
F	8	6	6	4	4	0

4

D	Confirmed	Tentative
1	(D,0,-)	
2	(D,0,-)	(B,2,B) (C,3,C), (F,4,F)
3	(D,0,-) (B,2,B)	(C,3,C) (F,4,F)
4	(D,0,-) (B,2,B)	(C,3,C) (F,4,F) (A,5,C) (E,5,C)
5	(D,0,-) (B,2,B) (C,3,C)	(F,4,F) (A,5,C) (E,5,C)
6	Previous + (F,4,F)	(A,5,C) (E,5,C)
7	Previous + (A,5,C)	(E,5,C)
8	Previous + (E,5,C)	

5

a) CSMA/CD is a collition detection algorithm while CSMA/CA is a collision avoidance algorithm, CA does not deal with recovery after a collision as it preempively checks if the medium is in use.

b) It is not possible for the transmitter to detect whether a collision occured or not in a wireless network. Because of this, wireless uses CA instead of CD

c) A sender can send a RTS and waits for a CTS. If sender sees CTS, it can send. If other node sees CTS, it waits for a specified period. All nodes will see a ACK when transmission is complete, allowing all nodes to send a RTS

6

a) H1

255.255.255.128 & 128.96.34.126 = 192.96.34.0

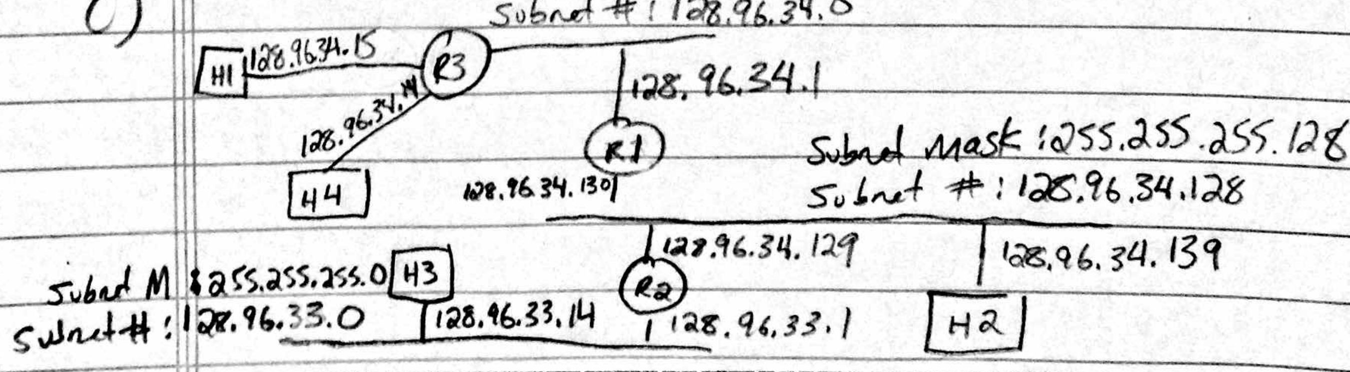
H1 sees that 192.96.34.0 is its network so it sends an ARP broadcast on network to find MAC address of destination. Upon a response, it sends a message to 128.96.34.126

b) H3

255.255.255.128 & 128.96.34.250 = 192.96.34.0

H3 sends ARP to find out which routers know of subnet 192.96.34.0. R2 responds saying it has a path, so H3 sends message to R2. R2 then forwards the message to the outgoing interface it knows has the subnet. This is 128.96.34.129 to 128.96.34.130. Once R1 receives the packet, assuming R1 knows all hosts on subnet, it can forward the message to 128.96.34.250

c)



7

a)

	src MAC	dst MAC	src IP	dst IP
B → R3	00:23:42:AC:EE:FC	AF:0C:EF:11:22:11	22.1.23.45	33.2.2.18
R3 → R2	AA:BB:CC:AF:2F:11	AF:CC:1F:11:EE:10	22.1.23.45	33.2.2.18
R2 → R4	0C:32:56:90:09:EF	89:98:AF:FF:FF:0C	22.1.23.45	33.2.2.18
R4 → C	00:1C:AA:FE:F1:1A	00:0C:EA:17:98:32	22.1.23.45	33.2.2.18

b) R1 returns 0C:AF:10:00:1E:1C

c)

	Interface MAC	Interface IP	Next Hop
22.1.0.0/16	AF:0C:EF:11:22:11	22.1.1.1	–
123.123.123.0/24	AA:BB:CC:AF:2F:11	20.11.33.254	R2
33.2.2.0/24	AA:BB:CC:AF:2F:11	20.11.33.254	R2

8

Distance to

Step	V3	V1	V6	V4	V2	V7	V5	Visited
1	0	∞	∞	∞	∞	∞	∞	{ }
2	0	2	3	∞	∞	∞	∞	{V3}
3	0	2	3	6	∞	∞	∞	{V3, V1}
4	0	2	3	6	∞	∞	∞	{V3, V1, V6}
5	0	2	3	6	8	∞	∞	{V3, V1, V6}
6	0	2	3	6	8	∞	12	{V3, V1, V6, V2}

In step 4, V1 is already visited so it never gets updated to the actual shortest path (V3→V6→V1). Dijkstra's algorithm fails to find the shortest path. It finds V3→V1→V4→V2→V5 where V3→V6→V1→V4→V2→V5 is actually the shortest path

9

a) $\overset{A=0\times T}{\frac{1}{2}} \cdot \overset{B\geq 1\times T}{\frac{3}{4}} + \overset{A=1\times T}{\frac{1}{2}} \times \overset{B\geq 1\times T}{\frac{1}{2}} = \frac{5}{8} = 62.5\%$

b) $\overset{A=0\times T}{\frac{1}{2}} \cdot \overset{B>0\times T}{\frac{7}{8}} + \overset{A\geq 1\times T}{\frac{3}{8}} = \frac{13}{16}$

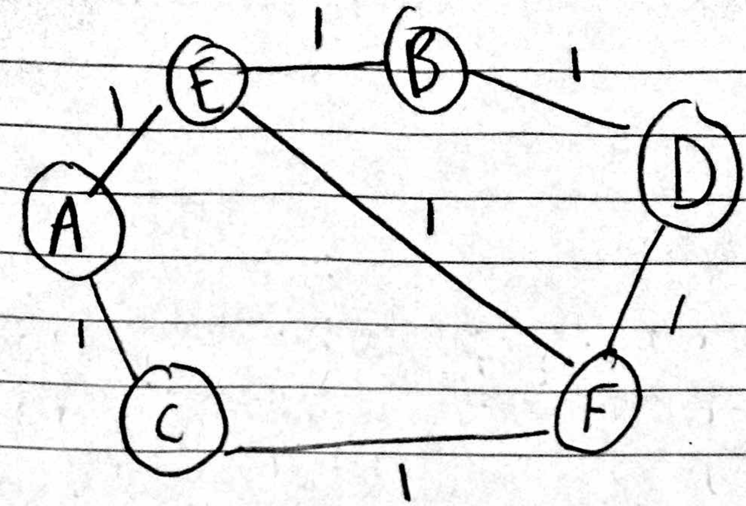
c) $P[A_4] \times P[A_5] \times P[A_6] \times \ldots \times P[A_{10}]$

$= \frac{29}{32} \times \frac{61}{64} \times \frac{125}{128} \times \frac{253}{256} \times \frac{509}{512} \times \frac{1021}{1024} \times \frac{2045}{2048}$

82%

d) B1 only gets delivered with probability = 18%. A takes over the link and B rarely wins any race

10



11

a)

Y	Via X	Via Z
X	60	8

Z	Via X	Via Y
X	50	7

b)

Y	Via X	Via Z
X	60	10

Z	Via X	Via Y
X	50	9

c) 27 updates

12

Before each request a RTS is sent over wifi and wait for CTS

1 laptop broadcasts

srcMAC = 2:3:4:5, IP = none, dstMAC = ff:ff:ff:ff, IP = none

DHCP Discover

2 DHCP -> laptop

srcMAC = 10:9:8:7, IP = 141.9.8.7, dstMAC = 2:3:4:5, IP = none

laptopIP = 141.1.1.1, DNS_IP = 141.1.1.0, routerIP = 141.1.1.2,

LAN_mask = 141.1.1.0/24

3 laptop broadcasts same as step 1

4 DHCP -> Laptop

DHCP ACK

5 laptop broadcasts

ARP 141.1.1.0

6 DNS -> Laptop

srcMAC = 6:6:6:6, IP = none, dstMAC = 2:3:4:5, IP = none

141.1.1.0 MAC = 6:6:6:6

7 laptop -> DNS

srcMAC = 2:3:4:5, IP = 141.1.1.1, dstMAC: 6:6:6:6, IP = 141.1.1.0

DNS lookup www.yahoo.com

8 DNS → laptop
srcMAC = 6:6:6:6, IP = 141.1.1.0, dstMAC = 2:3:4:5, IP = 141.1.1.1
www.yahoo.com IP = 192.40.0.1

9 laptop broadcasts
ARP 141.1.1.2

10 DNS → laptop
srcMAC = 5:4:3:2, IP = none, dstMAC = 2:3:4:5, IP = none
141.1.1.2 MAC = 5:4:3:2

11 laptop → yahoo.com
srcMAC = 2:3:4:5, IP = 141.1.1.1, dstMAC = 5:4:3:2, IP = 192.40.0.1
TCP SYN port 80

12 yahoo.com → laptop
srcMAC = 5:4:3:2, IP = 192.40.0.1, dstMAC = 2:3:4:5, IP = 141.1.1.1
TCP SYN ACK

13 laptop → yahoo.com (same header)
TCP ACK SYN ACK

14 laptop → yahoo.com (same header
TCP "HTTP GET"

15 yahoo.com → laptop
TCP "HTTP REPLY"

16 laptop → yahoo.com, TCP ACK "HTTP REPLY"