Wyatt Kormick
CSCI 4211
Take Home Quiz 2
11/27/17

1.
   a. 128.101.0.0
   b. The requesting client machine will be looking for offer messages that have the same transaction id as the discover message that it sent. If the transaction id is different, the message isn't for the client.
   c. We do not know. MAC addresses are assigned by the manufacturer to the adapter upon being built, and have nothing to do with the actual physical or virtual location of the adapter.
   d. The maximum amount of data that a link-layer frame can carry. It places a hard limit on the length of an IP datagram.
   e. With the given information, we cannot tell if they are from the same original IP datagram, we need to see the identifier field, but it is possible they could be because of the math for the lengths and the offsets work out. Assuming 20 bit headers, the actual length of the original data payload is 1580 (1600 including the header). (1500-20 + 120-20 (+20)).]

2.
   a. ACK packets tell their receiver what their sender's current window size is. One host (A) sends a packet that it knows will fit inside the buffer of the other host (B). Host B will send an ACK packet back to host A saying what the new current window size is after receiving the new packet and maybe processing some. Once again, host A will send a data packet that it knows will fit inside host B's buffer. If the buffer is full, the ACK packet will say the window size is 0. Host A will see that its full, and will send a packet of just 1 byte, for the sole purpose of eliciting a new ACK packet, with hopefully a new window size. It does this until an ACK packet says the buffer can hold more data, and continues the process.
   b. The only way we can have the numbers that we have is that we are in slow start. So for each ACK that we receive while congWin is less than threshold, we add our maximum segment size of 1500 bytes. We receive 3 ACKs, so our final CongWin is 13.5KB.
   c. Threshold = 6.75KB (half of old congestion window) CongWin=1.5KB (1MSS)
   d. We are in congestion avoidance, so for each new ACK, we add MSS*(MSS/Congwin) (150 in this case). CongWin = 15000+(150*5) = 15.75KB

3.
   a.

| Location | Router 3 | Router 4 | Host D |
|---|---|---|---|
| VCI | 2 | 2 | 2 |

Host D is the destination

B.

Router 1

| Input Port | Input VCI | Output Port | Output VCI |
|---|---|---|---|
| 0 | 4 | 1 | 2 |
| 0 | 3 | 3 | 2 |
| 0 | 5 | 1 | 3 |

Router 2

| Input Port | Input VCI | Output Port | Output VCI |
|---|---|---|---|
| 0 | 2 | 1 | 1 |
| 0 | 3 | 2 | 2 |
| 1 | 1 | 3 | 3 |

Router 3

| Input Port | Input VCI | Output Port | Output VCI |
|---|---|---|---|
| 0 | 2 | 2 | 1 |
| 3 | 2 | 2 | 2 |
| 1 | 1 | 3 | 3 |
| 1 | 2 | 3 | 4 |

Router 4

| Input Port | Input VCI | Output Port | Output VCI |
|---|---|---|---|
| 3 | 2 | 2 | 2 |
| 3 | 1 | 2 | 3 |
| 1 | 2 | 2 | 1 |
| 2 | 1 | 0 | 4 |

Router 5

| Input Port | Input VCI | Output Port | Output VCI |
|---|---|---|---|
| 1 | 3 | 3 | 1 |
| 2 | 4 | 3 | 2 |

C.
I. ppp. Matched with 0.0.0.0, because it doesn't match any prefix
II. eth0. Matches both 162.11.128.0 prefixes, but goes with the longer match for the network max, 255.255.192.0
III. eth1. Matches both 94.25.80.0 and 94.25.0.0, but has a longer prefix match with the first.
IV. eth1. Matches both 94.25.80.0 and 94.25.0.0, but has a longer prefix match with the first.
V. ppp. Matches with both 162.11.128.0 prefixes, but only matches with the 255.255.128.0 mask.

4.
A.

I. S3 will receive the broadcast frame because S4 broadcasted it as well. S3 will broadcast the broadcast frame because the destination IP is on the same subnet as S3. The MAC address for host G will be added to S3's forwarding table for interface 3.
II. R2 will receive the broadcast frame because S3 broadcast it. R2 will not forward the frame any further because the destination IP's subnet is on the adapter that the packet came from.

III.No. F knows the destination MAC address (G's) of the response because it is the same as the source MAC address of the ARP request message.
IV. It will send the ARP response to G because the destination MAC address is in S4's forwarding table from when it received the the request message.

B.
Since C already knows the MAC and IP of D, it can directly send its packets to host D (via S1) with D's MAC (retrieved from C's ARP cache) and IP as the destination addresses, and its own IP and MAC as the source addresses

C.

I. It would not perform an ARP query. Since H is not on the same subnet, C knows to send its message to R2, and the MAC of R2 is already in C's ARP cache, so it would use that as the destination MAc.
II. The source IP and MAC addresses of the packet would be C's IP and MAC. The destination IP would be H's IP, and the destination MAC would be R2's MAC.

III. No. It has to send it out interface 1. The packet from C has a specific IP address destination, and that matches in R2's ARP cache with H's First MAC address, which R2 knows is on interface 1.

IV. Source MAC will be R2's interface 1 MAC address. Destination MAC will be H's MAC Address 1.

D.

No. H knows the IP of R1, and can therefore determine which subnet it needs to send its packet out on. It doesn't need to broadcast, or send to R1 via S3 at all.

E.

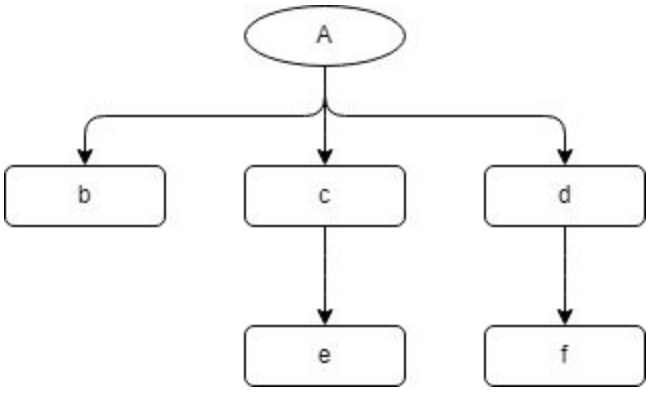
I. It will send out on interface 1, since the destination MAC is for R2.

II. H will send the packet back out on interface 1. This is because when R2 forwards the packet, it changes the source MAC address to its own, so H will set its destination MAC as R2's MAC and send it our there.

5.

| Step | N | D(B),p(B) | D(C),p(C) | D(D),p(D) | D(E),p(E) | D(F),p(F) |
|---|---|---|---|---|---|---|
| 0 | a | 3,a | 2,a | 4,a | inf | inf |
| 1 | ac | 3,a | | 4,a | 5,c | inf |
| 2 | acb | | | 4,a | 5,c | inf |
| 3 | acbd | | | | 5,c | 7,d |
| 4 | acbde | | | | | 7,d |
| 5 | acbdef | | | | | |

B.



C.

| Destination | Next |
|---|---|
| B | B |
| C | C |
| D | D |
| E | C |
| F | D |

D.
We'd need to include the previous node for each node, so that we can construct a path by following them.

| Destination | Next | Previous |
|---|---|---|
| B | B | A |
| C | C | A |
| D | D | A |
| E | C | C |
| F | D | D |

E.
Installing in A
Destination IP address == F ⇒ forward to B
Installing in B
Destination IP address == F ⇒ forward to D
Installing in D
Destination IP address == F ⇒ forward to F