Computer Networks Homework 3 12/3/2016

Username: Ronair

Ch4 P11.

- Interface 0: 0000 0000 – 0011 1111
  - Since the router only matches the prefix of the 8-bit host address. First 2 bits are fixed 00 and rest 6 bits are available, hence addresses = 2^6 = 64
- Interface 1: 0100 0000 – 0101 1111
  - Addresses = 2^5 = 32 (First 3 bits are fixed)
- Interface 2: 0110 0000 – 0111 1111 (First 2 and First 3 bits are fixed respectively)
  - Addresses = 2^6 + 2^5 = 64 + 32 = 96
- Interface 2: 1000 0000 – 1011 1111
  - Addresses Combined above
- Interface 3: 1100 0000 – 1111 1111
  - Addresses = 2^6 = 64

Ch4 P13.

We first fix the prefix 223.1.17 for all 3 subnets as per the requirement (first 24 bits)
(rounding to next power of 2. For instance, 60 becomes 64 etc.)

- We need 6 bits for at least 60 addresses, hence subnet 1 = 223.1.17.0/26
- We need 7 bits for at least 90 addresses, hence subnet 2 = 223.1.17.128/25
- We need 4 bits for at least 60 addresses, hence subnet 3 = 223.1.17.192/28

Ch4 P21.

a. Home network interfaces can have addresses 192.168.1.1, 192.168.1.2, 192.168.1.3 and the router can have 192.168.1.4
b.

WAN	LAN
24.34.112.235, 4000	192.168.1.1, 3345
24.34.112.235, 4001	192.168.1.1, 3346
24.34.112.235, 4002	192.168.1.2, 3345
24.34.112.235, 4003	192.168.1.2, 3346
24.34.112.235, 4004	192.168.1.3, 3345
24.34.112.235, 4005	192.168.1.3, 3346

Ch4 P26.

		t		u		v		w		y		z	
		Distance	Source	Distance	Source	Distance	Source	Distance	Source	Distance	Source	Distance	Source
init		inf	-	inf	-	inf	-	inf	-	inf	-	inf	-
0	x	inf	-	inf	-	3	x	6	x	6	x	8	x
1	xv	7	v	6	v	3	x	6	x	6	x	8	x
2	xvu	7	v	6	v	3	x	6	x	6	x	8	x
3	xvuw	7	v	6	v	3	x	6	x	6	x	8	x
4	xvuwy	7	v	6	v	3	x	6	x	6	x	8	x
5	xvuwyt	7	v	6	v	3	x	6	x	6	x	8	x
6	xvuwytz	7	v	6	v	3	x	6	x	6	x	8	x

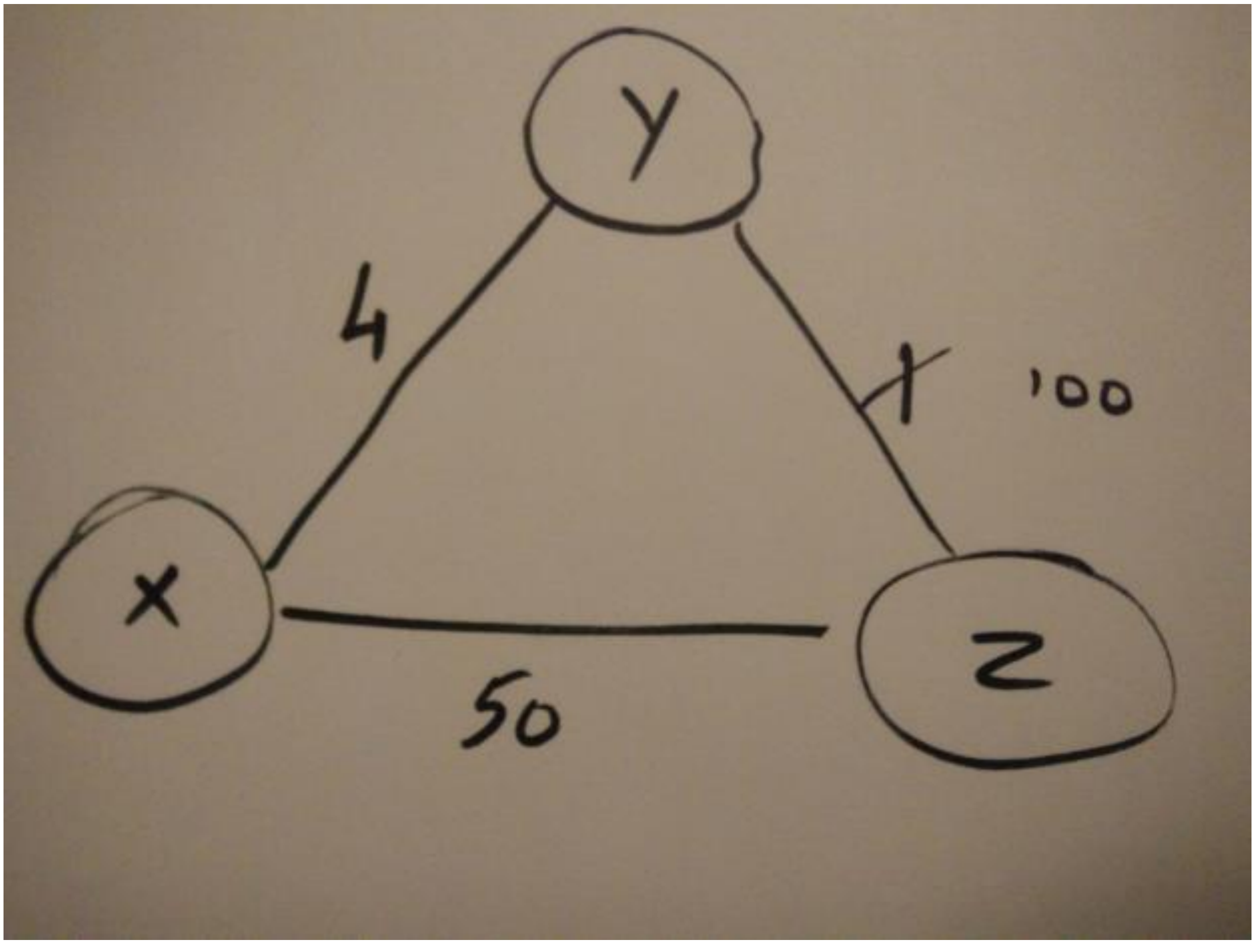
Additional Problem:

a. Count to Infinity Problem:

We have $D_y(z) = 1$, $D_x(z) = 5$.

1. y detects a link cost change from 1 to 100, so it computes a new minimum-cost path to z having a cost = $\min\{c(y,z) + D_z(z), c(y,x) + D_x(z)\} = \min\{100 + 0, 4 + 5\} = 9$. Hence the new $D_y(z)$ is 9
2. y informs x of its new distance vector.
3. x knows it can get to y with a cost of 4 and hence computes it new least cost to z of $D_x(z) = \min\{50+0, 4+9\} = 13$
4. x informs y of its new distance vector.
5. Now y calculates $D_y(z) = 17$ and informs x

This cycle goes on for around 12 iterations, until x sees that the cost to z via y is > 50, so it chooses the other path.



b. Using poisoned reverse:
   1. y detects a change in the link cost but x continues to advertise its cost to z as infinity. Thus, y will continue to reach z directly (cost = 100 now)
   2. y informs x of its new cost to reach z.
   3. x will now reach z via direct link of 50 instead of routing through y since that's the path with lesser cost.
   4. x now informs y about this new cost to reach z.
   5. y sees that this new path with cost 54 is cheaper than its current entry of 100 to reach z. So, $D_y(z)$ becomes 54.

And thus, the count to infinity problem is addressed.

Ch5 P2.

(Assuming 0 based indexing)

True matrix:

0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	1	1
0	0	1	1	1	1
0	0	0	0	0	0

Single bit error in row 2, column 2:

0	0	0	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	1	1	1	1
0	0	1	1	1	1
0	0	0	0	0	0

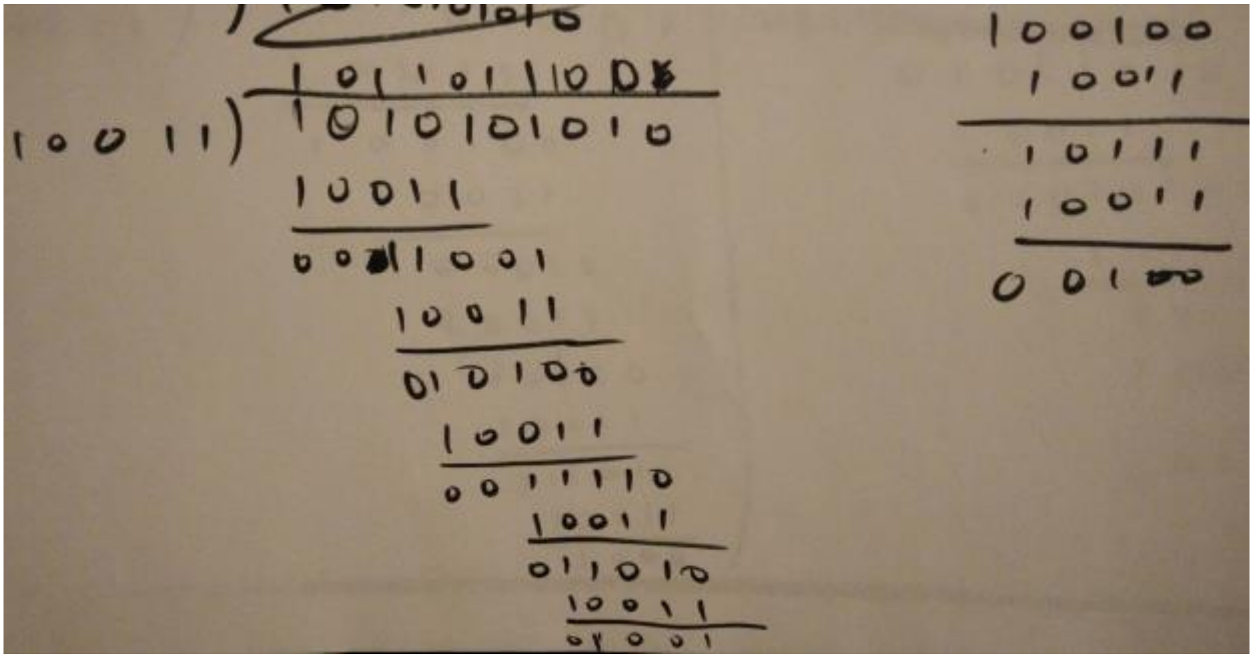
We detect that there's an error and know exactly which bit

Double bit error in row 3, columns 0 and 1:

0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	0	1	1
0	0	1	1	1	1
0	0	1	1	1	1
0	0	0	0	0	0

We detect that there's an error but don't know exactly which row since the 2 bit errors cancelled each other as far as that row's parity is concerned.

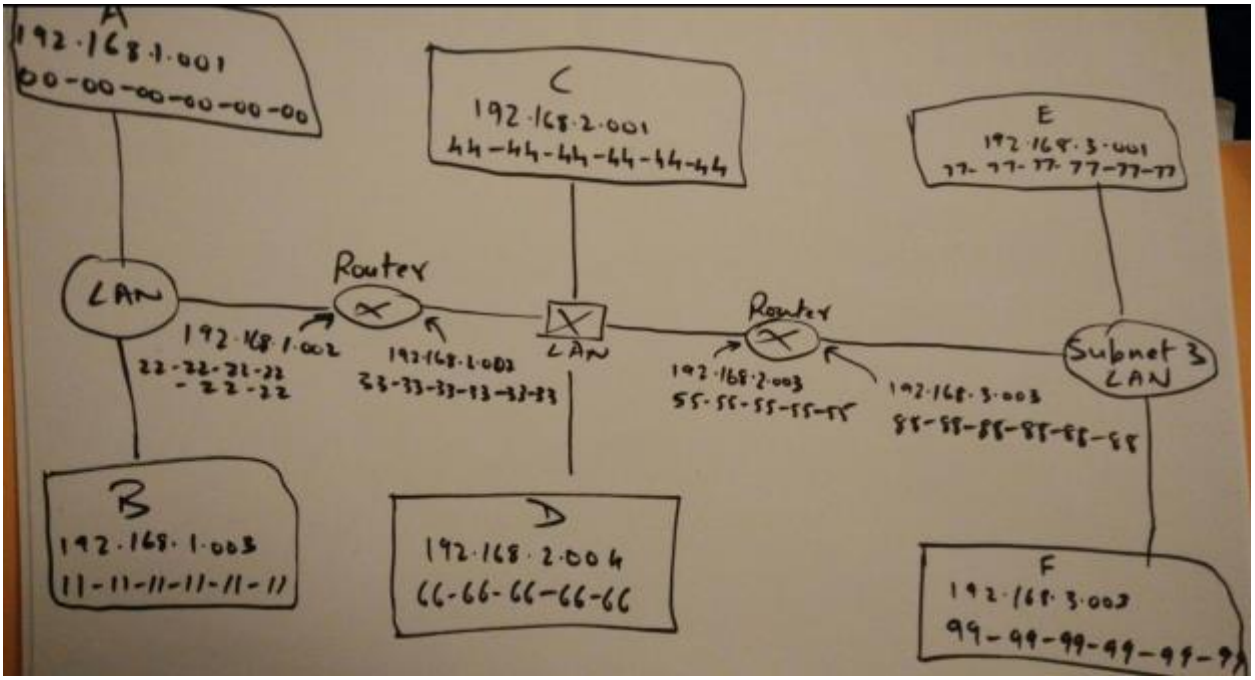
Ch5 P5.



From the above calculation, we see that the quotient is 1011011100 and R = 100

Ch5 P14.

a and b.



c.

We're given that the ARP tables are up to date, so host E to host B will be as follows:

- E knows from its forwarding table that the datagram needs to be routed to interface 192.168.3.002
- The adapter sends an Ethernet packet with destination 88-88-88-88-88-88, which is nothing but the right-hand side router
- The router extracts the datagram
- It knows from its forwarding table that the datagram needs to be routed to 192.168.2.002
- It sends an Ethernet packet with destination 33-33-33-33-33-33 and source IP 192.168.2.003 and source address of 55-55-55-55-55-55
- This is the left-hand side router. It knows from its forwarding table that the datagram needs to be routed to 192.168.1.003
- It sends an Ethernet packet with destination 11-11-11-11-11-11 and source IP 192.168.1.002 and source address of 22-22-22-22-22-22
- The packet is at the destination.

d.

- E's ARP must know what the MAC address of 192.168.3.002 is
- It sends a broadcasts frame with an ARP query packet within.
- The right-hand side router catches hold of this frame.
- It sends an ARP response packet with destination 77-77-77-77-77-77 to E.
- Since other tables are up to date, we're good.
- The process is same as above after this.

Ch5 P15

a. No
b. No
c. Yes

Explanations below:

a. Hosts E and F are in the same LAN. So, it will not require sending the packet to router R1, it can send the packet directly.
   Ethernet frame from Host E to Host F will have the below addresses:
   - Ethernet frame Source IP : E's IP address
   - Ethernet frame Source MAC : E's MAC address
   - Ethernet frame Destination IP : F's IP address
   - Ethernet frame Destination MAC : F's MAC address

b. Host E and B are in different LANs, hence E will not perform an ARP query to find the B's MAC address.
   - Ethernet frame Source IP : E's IP address
   - Ethernet frame Source MAC : E's MAC address
   - Ethernet frame Destination IP : B's IP address
   - Ethernet frame Destination MAC : R1's MAC address (connected to subnet 3 interface)

c. A broadcasts an ARP request. Switch S1 receives it and learns that the destination address is the broadcast address. Thus, S1 will broadcast the frame on all its interfaces.

S1 learns that A in on subnet 1 at interface connecting S1 to subnet. S1's forwarding table will now have an entry for A.

Yes, R1 will receive the ARP request message but it will not forward it to Subnet 3 since they're in the same LAN

Host B learns Host A's MAC address from the ARP request message. It sends an ARP response message. It won't need to send an ARP request message since it knows A's MAC address from the request message it received.

S1 sees this response and adds an entry for Host B into its forwarding table

Now, since Host A and B are on the same interface, it will drop the frame.

Ch5 P18

Time t = 0 : A starts transmitting
Time t = 324 : B starts transmitting
Time t = 576 : A finishes transmitting
(So there's a collision)

But given that there's a propagation delay between A and B which is 325 bit times, B's first bit arrives at A at time 324 + 325 = 649.
At A, since this is after it has finished transmitting, it thinks that there's no collision.

Ch5 P26

Initially, the switch (let's call it S) has an empty forwarding table

i. B sends frame to E
S adds an entry to its forwarding table for Host B because it learns the interface corresponding to MAC address B. S forwards the packet to all the others since it does not know the interface corresponding to the MAC address of E

ii. E replies with frame to B
S adds an entry to its table for Host E because it learns the interface corresponding to MAC address E. S forwards the frame to Host A directly as it knows the address of Host A from its forwarding table.

iii. A sends frame to B
S adds an entry to its forwarding table for Host A because it learns the interface corresponding to MAC address A. S forwards the frame to Host B directly as it knows the address for Host B from its forwarding table.

iv. B replies with frame to A
The frame is forwarded to A since switch already knows the address for Host A from its forwarding table which contains entries for both Host A and B already. Nothing new to update in the forwarding table.