CENG460 Assignment 3

Student ID:

1. Consider IP address X: 193.101.50.44/26

   (a) Which of the IP addresses below is on the same network as X:

   193.101.50.44/26:

net_add	11000001 01100101 00110010 00101100
&net_mask	11111111 11111111 11111111 11000000
=	11000001 01100101 00110010 00000000

   193.101.50.10/26:
   net_address &net_mask = 11000001 01100101 00110010 00000000
   193.100.50.11/26:
   net_address &net_mask = 11000001 0110010**0** 00110010 00000000
   193.101.50.65/26:
   net_address &net_mask = 11000001 01100100 00110010 0**1**000000

IP address	On the same network as X (Yes or No)?
193.101.50.10	Yes
193.**100**.50.11	No
193.101.50.**65**	No

   (b) With subnet mask 255.255.255.192, what is the maximum number of hosts on the subnet?

   Since the subnet mask has 26 bits "1", the address space for hosts is $2^6 = 64$. In addition, two addresses are normally not used for hosts: subnet identifier (all zeros) and the broadcast address (all ones); thus, the max number of hosts is $64 - 2 = 62$.

2. An IP packet has the following information in its header arrives at a WLAN:

...	length	ID	fragflag	offset	...
	5000	x	0	0	

   Since the maximum transmission unit (MTU) of the WLAN is 2308 bytes, the packet will be fragmented into how many packets? What will be the length, ID, fragflag and offset values in their IP headers?

   Solution: An IP packet (datagram) contains the IP header (20 bytes by default) and a data portion. MTU is the maximum length allowed for each datagram. The "length" field in a datagram identifies the total datagram length, including both the data portion and the IP header. The "offset" equals the number of 8-byte blocks between the beginning of the current datagram and the beginning of the original datagram. The "fragflag" bit shows whether there are more datagram to come to reassemble the original datagram.

Since $2308 - 20 = 2288$ bytes.

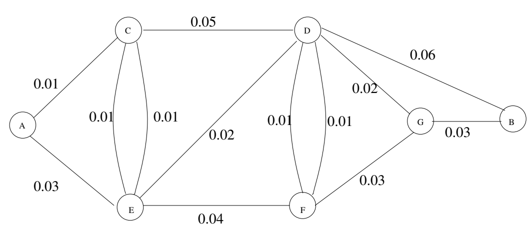
The original IP packet will be fragmented to three datagrams:

...	length	ID	fragflag	offset	...
	2308	x	1	0	

...	length	ID	fragflag	offset	...
	2308	x	1	286	

...	length	ID	fragflag	offset	...
	424	x	0	572	

3. The number shown in the following figure is the probability of the link failing. It is assumed that links fail independently of each other.



(a) Find the most reliable path from A to B, i.e., the path for which the probability that all links stay intact is maximal. [Hint: for link i with failing probability $p_i << 1$ and link j with failing probability $p_j << 1$, Pr{fail of the path through link i and link j} $= 1-(1-p_i)(1-p_j) = p_i + p_j - p_i p_j \approx p_i + p_j$]

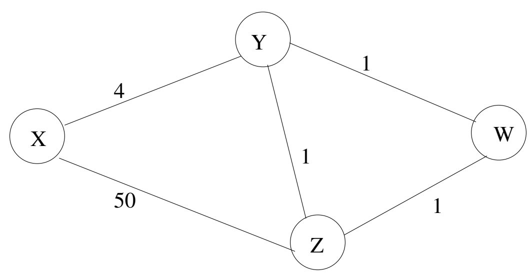
Iterations	A	B	C	D	E	F	G
Initially	(0, A)	$(\infty,\ .)$	(0.01, A)	$(\infty,\ .)$	(0.03, A)	$(\infty,\ .)$	$(\infty,\ .)$
1	(0, A)	$(\infty,\ .)$	(0.01, A)	(0.06, C)	(0.02, C)	$(\infty,\ .)$	$(\infty,\ .)$
2	(0, A)	$(\infty,\ .)$	(0.01, A)	(0.04, E)	(0.02, C)	(0.06, E)	$(\infty,\ .)$
3	(0, A)	(0.1, D)	(0.01, A)	(0.04,E)	(0.02, C)	(0.05, D)	(0.06, D)
4	(0, A)	(0.1, D)	(0.01, A)	(0.04,E)	(0.02, C)	(0.05, D)	(0.06, D)
5	(0, A)	(0.09, G)	(0.01, A)	(0.04,E)	(0.02, C)	(0.05, D)	(0.06, D)

The most reliable path from A to B is "A C E D G B".

(b) Find the second most reliable path from A to B which does not share any link belonging to the path found in (a).

Remove all the links used in (a), the most reliable path from A to B is "AEFDB".

4. In the network shown in the following figure, a routing protocol based on distance vector algorithm is used. Routers exchange distance vectors every 30 seconds. During each round, the sequence that the routers send out distance vectors is X, Y, Z, W. The number shown in the figure is the cost of each link, and routers try to use the path with the minimum cost to deliver packets. Assume at time t_0, the cost between X and Y is increased from 4 to 60.



(a) If **no** poisoned reverse is used, what will be the distance vectors from Y to X after one, two, and three distance vectors exchanges between neighbors?

The answer depends on the sequence of the distance vector updates from Y, Z, and W. Assume that, in each round, Y sends its distance vector to W and Z first, followed by Z, and followed by W.

In the first round exchange:
Y sends $\min(Y\ to\ X) = 6$ to Z and W;
Z sends $\min(Z\ to\ X) = 6$ to Y and W;
W sends $\min(W\ to\ X) = 7$ to Z and Y.

In the 2nd round:
Y sends $\min(Y\ to\ X) = 7$ to W and Z;
Z sends $\min(Z\ to\ X) = 8$ to Y and W;
W sends $\min(W\ to\ X) = 8$ to Z and Y.

In the 3rd round:
Y sends $\min(Y\ to\ X) = 9$ to W and Z;
Z sends $\min(Z\ to\ X) = 9$ to Y and W;
W sends $\min(W\ to\ X) = 10$ to Z and Y.

From Y to X thru	neighbor X	neighbor Z	neighbor W
initially	$4 \to 60$	6	6
after 1st exchange	60	7	8
after 2nd exchange	60	9	9
after 3rd exchange	60	10	11

From Z to X thru	neighbor X	neighbor Y	neighbor W
initially	50	5	6
after 1st exchange	50	7	8
after 2nd exchange	50	8	9
after 3rd exchange	50	10	11

From W to X thru	neighbor Y	neighbor Z
initially	5	6
after 1st exchange	7	7
after 2nd exchange	8	9
after 3rd exchange	10	10

The results show that the routers are very slow to realize the "bad news", and cannot choose the optimal paths promptly. This is the so called "count-to-infinity" problem. For instance, even after a few rounds of message exchange, packets from Z to X will be bounced between Z, Y and W according to their routing tables.

(b) If poisoned reverse is used, what will be the distance vectors from Y to X after one, two, and three distance vectors exchanges between neighbors?

In each round, Y sends its distance vector to W and Z first, followed by Z, and followed by W.

In the first round exchange:
Y sends $\min(Y\ to\ X) = 60$ to Z and W;
Z sends $\min(Z\ to\ X) = 6$ to Y, and $\min(Z\ to\ X) = \infty$ to W;
W sends $\min(W\ to\ X) = 61$ to Z, and $\min(W\ to\ X) = \infty$ to Y.

In the 2nd round:
Y sends $\min(Y\ to\ X) = 7$ to W, and $\min(Y\ to\ X) = \infty$ to Z;
Z sends $\min(Z\ to\ X) = 50$ to Y and W;
W sends $\min(W\ to\ X) = 9$ to Z, and $\min(W\ to\ X) = \infty$ to Y.

In the 3rd round:
Y sends $\min(Y\ to\ X) = 51$ to W, and $\min(Y\ to\ X) = \infty$ to Z;
Z sends $\min(Z\ to\ X) = 9$ to Y, and $\min(Z\ to\ X) = \infty$ to W;
W sends $\min(W\ to\ X) = 52$ to Z, and $\min(W\ to\ X) = \infty$ to Y.

From Y to X	neighbor X	neighbor W	neighbor Z
initially	$4 \rightarrow 60$	∞	∞
after 1st exchange	60	∞	7
after 2nd exchange	60	∞	51
after 3rd exchange	60	∞	10

From Z to X	neighbor X	neighbor Y	neighbor W
initially	50	5	6
after 1st exchange	50	61	62
after 2nd exchange	50	∞	9
after 3rd exchange	50	∞	53

From W to X	neighbor Y	neighbor Z
initially	5	6
after 1st exchange	61	∞
after 2nd exchange	8	51
after 3rd exchange	52	8

The results show that, even the Poisoned Reverse scheme is used, the routers cannot figure out the "bad news" quickly, and it may take a long time for them to finally find out their best paths to X (YZX, ZX, and WZX, respectively). Before that, packets to X might be delivered through a sub-optimal path (e.g., W will choose Y instead of Z for packets to X). Since the distance vector algorithm requires the routers to exchange information locally only (with their neighbors), the slow response to “bad news” is a problem that has not be effectively solved yet.