EN.601.713. Future Network

Week01 Homework

1. (10 points) An IP datagram using the Strict source routing option has to be fragmented. Do you think the option is copied into each fragment, or is it sufficient to just put it in the first fragment? Explain your answer

I think the option **needs** to be copied into each fragment. Because the Strict source routing option requires these fragmented datagrams to follow an exact specific route. If we only set for the first fragment, then the other fragments can be sent by different routes, leading to the final destination can not reassembled.

2. (10 points) A class B network on the Internet has a subnet mask of 255.255.248.0, what is the maximum number of hosts per subnet?

Since 255.255.248.0 in decimal is equivalent to 11111111.11111111.11111000.00000000 in binary, identifying 11 bits for the host identifier. Since we will need one each for the network address and the broadcast address, the maximum number of hosts per subnet is: 2^11-2=**2046**.

3. (10 points) IPv6 uses 16-byte addresses. If a block of 1 million addresses is allocated every picosecond, how long will the addresses last?

Since IPv6 uses 16-byte addresses, we can number a total of: 2^(16*8)=2^128 addresses.

On the other hand: since a block of 1 million addresses is allocated every picosecond, assume 1 year last 365 days, then this equivalent we can address: (10^6 / 10^-12) * 60 * 60 * 24 * 365 = 3.1536*10^25 address every year.

As a result, the IPv6 addresses last: (2^128)/(3.1536*10^25) ~= **1.079*10^13 years**.

4. (10 points) A new virtual circuit is being set up in an ATM network. Between the source and destination hosts lie three ATM switches. How many messages (including acknowledgments) will be sent to establish the circuit?

Let the source and destination hosts be "S" and "D", and three ATM switches be "S1", "S2", and "S3" respectively. Because this is a virtual circuit network, it requires a connection service along the source-to-dest path. So to establish the circuit, it will need:

(1) A message from S to S1 (init request)
(2) A message from S1 to S2 (routed request)
(3) A message from S2 to S3 (routed request)
(4) A message from S4 to D (acknowledgments)

As a consequence, we will need a total of **4 messages**.

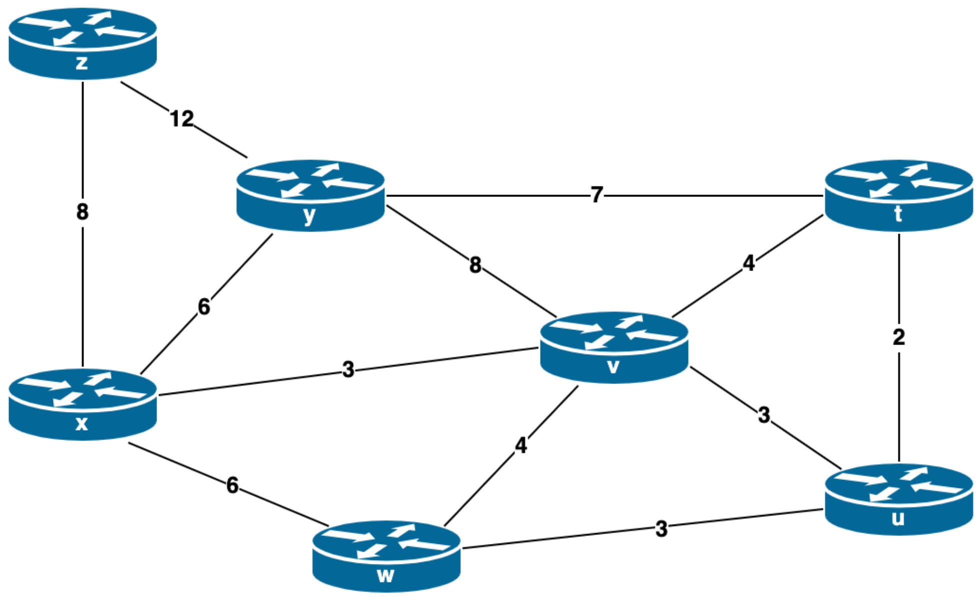
5. (10 points) Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

Since we have a 2400-byte datagram and MTU = 700 bytes, due to the 20-byte IP header at the head, there will be: int((2400-20)/(700-20)) = int(3.5) = **4 fragments generated**.

By assuming the ID number of the original datagram = 422, we have the various fields in the IP datagram(s) generated related to fragmentation **Table as follows**:

	length = 700	ID = 422	fragflag = 1	offset = 0	
	length = 700	ID = 422	fragflag = 1	offset = (680/8) = 85	
	length = 700	ID = 422	fragflag = 1	offset = (1360/8) = 170	
	length = (2380-2040) = 340		fragflag = 0	offset = (2040/8) = 255	

6. (30 points) Consider the network shown below.



Using Dijkstra's algorithm and showing your work. do the following:

a. Compute the shortest path from *t* to all network nodes.
b. Compute the shortest path from *u* to all network nodes.
c. Compute the shortest path from *v* to all network nodes.
d. Compute the shortest path from *w* to all network nodes.
e. Compute the shortest path from *y* to all network nodes.
f. Compute the shortest path from *z* to all network nodes.

Apply Dijkstra's algorithm, we have the Tables for each case as follows

a. The shortest path from *t* to all network nodes

Step	N'	D(u)	D(v)	D(y)	D(x)	D(w)	D(z)
0	t	**2,t**	4,t	7,t	inf	inf	inf
2	tu		**4,t**	7,t	inf	5,u	inf
3	tuv			7,t	7,v	**5,u**	inf
4	tuvw			**7,t**	7,v		inf
5	tuvwy				**7,v**		19,y
6	tuvwyx						**15,x**

b. The shortest path from *u* to all network nodes

Step	N’	D(t)	D(v)	D(w)	D(x)	D(y)	D(z)
0	u	**2,u**	3,u	3,u	inf	inf	inf
1	ut		**3,u**	3,u	inf	9,t	inf
2	utv			**3,u**	6,v	9,t	inf
3	utvw				**6,v**	9,t	inf
4	utvwx					**9,t**	14,x
5	utvwxy						**14,x**

c. The shortest path from *v* to all network nodes

Step	N’	D(t)	D(u)	D(w)	D(x)	D(y)	D(z)
0	v	4,v	**3,v**	4,v	3,v	8,v	inf
1	vu	4,v		4,v	**3,v**	8,v	inf
2	vux	**4,v**		4,v		8,v	11,x
3	vyxt			**4,v**		8,v	11,x
4	vyxtw					**8,v**	11,x
5	vyxtwy						**11,x**

d. The shortest path from *w* to all network nodes

Step	N’	D(u)	D(v)	D(x)	D(t)	D(y)	D(z)
0	w	**3,w**	4,w	6,w	inf	inf	inf
1	wu		**4,w**	6,w	5,u	inf	inf
2	wuv			6,w	**5,u**	12,v	inf
3	wuvt			**6,w**		12,v	inf
4	wuvtx					**12,v**	14,x
5	wuvtxy						**14,x**

e. The shortest path from *y* to all network nodes

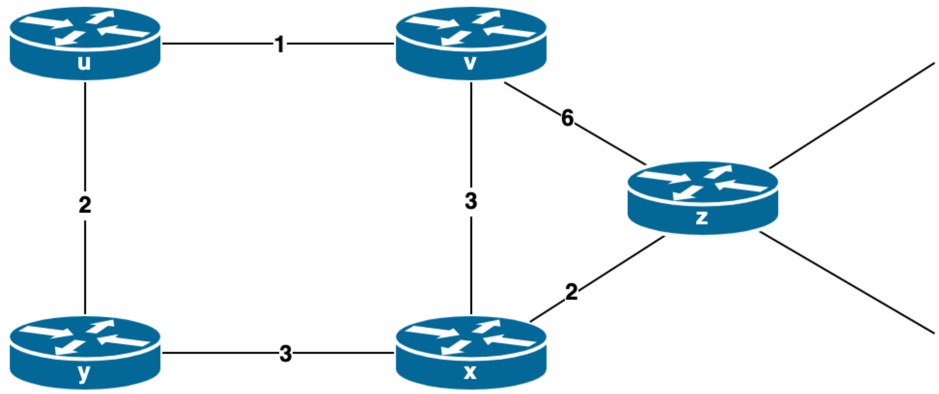
Step	N’	D(t)	D(v)	D(x)	D(z)	D(u)	D(w)
0	y	7,y	8,y	**6,y**	12,y	inf	inf
1	yx	**7,y**	8,y		12,y	inf	12,x

2	yxt		**8,y**		12,y	9,t	12,x
3	yxtv				12,y	**9,t**	12,x
4	yxtvu				**12,y**		12,x
5	yxtvuz						**12,x**

f. The shortest path from *z* to all network nodes

Step	N’	D(x)	D(y)	D(t)	D(v)	D(w)	D(u)
0	z	**8,z**	12,z	inf	inf	inf	inf
1	zx		12,z	inf	**11,x**	14,x	inf
2	zxv		**12,z**	15,v		14,x	14,v
3	zxvy			15,v		**14,x**	14,v
4	zxvyw			15,v			**14,v**
5	zxvywu			**15,v**			

7. (20 points) Consider the network shown below



Assume that each node initially knows the costs to each of its neighbors. Consider the distance-vector algorithm and show the distance table entries at node z.

Step 1: Note z table, init:

From/Cost to	u	v	x	y	z
v	inf	inf	inf	inf	inf
x	inf	inf	inf	inf	inf
z	inf	6	2	inf	0

Step 2: Apply DV algorithm, and update cost from initial-tables of other neighbours:

- D_z(u) = min{c(z,v) + D_v(u)} = min{6+1} = 7
- D_z(v) = min{c(z,x) + D_x(v), c(z,v) + D_v(v)} = min{2+3, 6} = 5
- D_z(x) = min{c(z,v) + D_v(x), c(z,x) + D_x(x)} = min{6+3, 2} = 2
- D_z(y) = min{c(z,x) + D_x(y)} = min{2+3} = 5

Since D_z(v) and D_z(y) changes, we need to re-update:

- D_z(u) = min{c(z,v) + D_v(u)} = min{5+1} = 6
- D_z(x) = min{c(z,v) + D_v(x), c(z,x) + D_x(x)} = min{5+3, 2} = 2

So, we obtain the new note z table as follows

From/Cost to	u	v	x	y	z
v	1	0	3	inf	6
x	inf	3	0	3	2
z	6	5	2	5	0

Step 3: Update cost form updated-tables of other neighbours after applied DV algorithm:

- D_v(x) = min{c(v,z) + D_z(x), c(v,x) + D_x(x)} = min{6+2, 3} = 3
- D_v(y) = min{c(v,u) + D_u(y), c(v,x) + D_x(y)} = min{1+2, 3+3} = 3
- D_v(z) = min{c(v,x) + D_x(z), c(v,z) + D_z(z)} = min{3+2, 6} = 5
- D_x(u) = min{c(x,v) + D_v(u), c(x,y) + D_y(u)} = min{1+3, 2+3} = 4
- D_x(v) = min{c(x,z) + D_z(v), c(x,v) + D_v(v)} = min{2+6, 3} = 3
- D_x(z) = min{c(x,v) + D_v(z), c(x,z) + D_z(z)} = min{3+6, 2} = 2

As a consequence, we obtain the final note z table as follows

From/Cost to	u	v	x	y	z
v	**1**	**0**	**3**	**3**	**5**
x	**4**	**3**	**0**	**3**	**2**
z	**6**	**5**	**2**	**5**	**0**