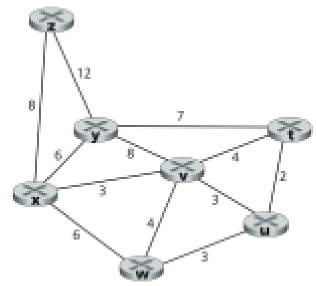
5장 과제

수학과 202021224 주민찬

P3. Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.



step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	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

P4. Consider the network shown in Problem P3. Using Dijkstra's algorithm, and showing your work using a table similar to Table 5.1, do the following:

a. Compute the shortest path from t to all network nodes.

step	N'	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	t	2,t	4,t	∞	∞	7,t	∞
1	tu	2,t	4,t	5,u	∞	7,t	∞
2	tuv	2,t	4,t	5,u	7,v	7,t	∞
3	tuvw	2,t	4,t	5,u	7,v	7,t	∞
4	tuvwx	2,t	4,t	5,u	7,v	7,t	15,x
5	tuvwxy	2,t	4,t	5,u	7,v	7,t	15,x
6	tuvwxyz	2,t	4,t	5,u	7,v	7,t	15,x

b. Compute the shortest path from u to all network nodes.

step	N'	D(t),p(t)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	3,u	3,u	∞	∞	∞
1	ut	2,u	3,u	3,u	∞	9,t	∞
2	utv	2,u	3,u	3,u	6,v	9,t	∞
3	utvw	2,u	3,u	3,u	6,v	9,t	∞
4	utvwx	2,u	3,u	3,u	6,v	9,t	14,x
5	utvwxy	2,u	3,u	3,u	6,v	9,t	14,x
6	utvwxyz	2,u	3,u	3,u	6,v	9,t	14,x

c. Compute the shortest path from v to all network nodes.

step	N'	D(t),p(t)	D(u),p(u)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	v	4,v	3,v	4,v	3,v	8,v	∞
1	vx	4,v	3,v	4,v	3,v	8,v	11,x
2	vxu	4,v	3,v	4,v	3,v	8,v	11,x
3	vxut	4,v	3,v	4,v	3,v	8,v	11,x
4	vxutw	4,v	3,v	4,v	3,v	8,v	11,x
5	vxutwy	4,v	3,v	4,v	3,v	8,v	11,x
6	vxutwz	4,v	3,v	4,v	3,v	8,v	11,x

d. Compute the shortest path from w to all network nodes.

step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	w	∞	3,w	4,w	6,w	∞	∞
1	wu	5,u	3,w	4,w	6,w	∞	∞
2	wuv	5,u	3,w	4,w	6,w	12,v	∞
3	wuvt	5,u	3,w	4,w	6,w	12,v	∞
4	wuvtx	5,u	3,w	4,w	6,w	12,v	14,x
5	wuvtxy	5,u	3,w	4,w	6,w	12,v	14,x
6	wuvtxyz	5,u	3,w	4,w	6,w	12,v	14,x

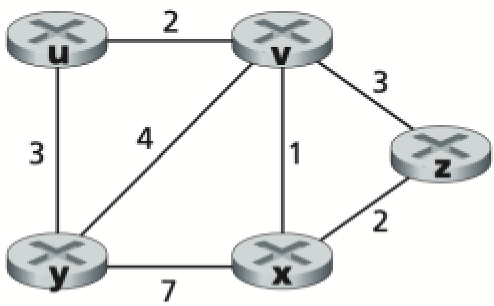
e. Compute the shortest path from y to all network nodes.

step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(z),p(z)
0	y	7,y	∞	8,y	∞	6,y	12,y
1	yx	7,y	∞	8,y	12,x	6,y	12,y
2	yxt	7,y	9,t	8,y	12,x	6,y	12,y
3	yxtv	7,y	9,t	8,y	12,x	6,y	12,y
4	yxtvu	7,y	9,t	8,y	12,x	6,y	12,y
5	yxtvuw	7,y	9,t	8,y	12,x	6,y	12,y
6	yxtvuwz	7,y	9,t	8,y	12,x	6,y	12,y

f. Compute the shortest path from z to all network nodes.

step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)
0	z	∞	∞	∞	∞	8,z	12,z
1	zx	∞	∞	11,x	14,x	8,z	12,z
2	zxv	15,v	14,v	11,x	14,x	8,z	12,z
3	zxvy	15,v	14,v	11,x	14,x	8,z	12,z
4	zxvyu	15,v	14,v	11,x	14,x	8,z	12,z
5	zxvyuw	15,v	14,v	11,x	14,x	8,z	12,z
6	zxvyuwt	15,v	14,v	11,x	14,x	8,z	12,z

P5. 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.

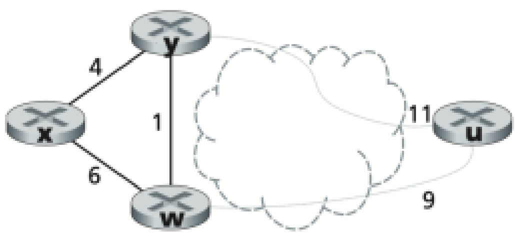


	u	v	x	y	z
z	∞	3	2	∞	0
x	∞	∞	∞	∞	∞
v	∞	∞	∞	∞	∞

	u	v	x	y	z
z	∞	3	2	∞	0
x	∞	3	2	9	0
v	∞	∞	∞	∞	∞

	u	v	x	y	z
z	∞	3	2	∞	0
x	∞	3	2	9	0
v	5	3	2	7	0

P7. Consider the network fragment shown below. x has only two attached neighbors, w and y. w has a minimum-cost path to destination u - (illustrated with the dotted line through the remaining network) of 9, and y has a minimum-cost path to u of 11. The complete paths from w and y to u (and between w and y) are pictured with dotted lines, as they are irrelevant to the solution.



a. Give x's distance vector for destinations w, y, and u.

- Dx(w) = 6, Dx(y) = 4, Dx(u) = 14

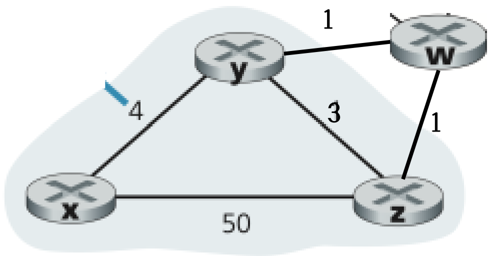
b. Give a link-cost change for either c(x,w) or c(x,y) such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

- c(x,w)를 감소 예를 들어 c(x,w)를 4로 감소하면 Dx(u) = 13으로 줄어든다.

c. Give a link-cost change for either c(x,w) or c(x,y) such that x will not inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

- c(x,w) 증가 예를 들어 c(x,w)를 9로 증가시켜도 Dx(u)는 14로 동일하다.

P11. Consider Figure 5.7. Suppose there is another router w, connected to router y and z. The costs of all links are given as follows: c(x,y) = 4, c(x,z) = 50, c(y,w) = 1, c(z,w) = 1, c(y,z) = 3. Suppose that poisoned reverse is used in the distance-vector routing algorithm.



a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?

- Dw(x) = 5, Dy(x) = 4, Dz(x) = 6

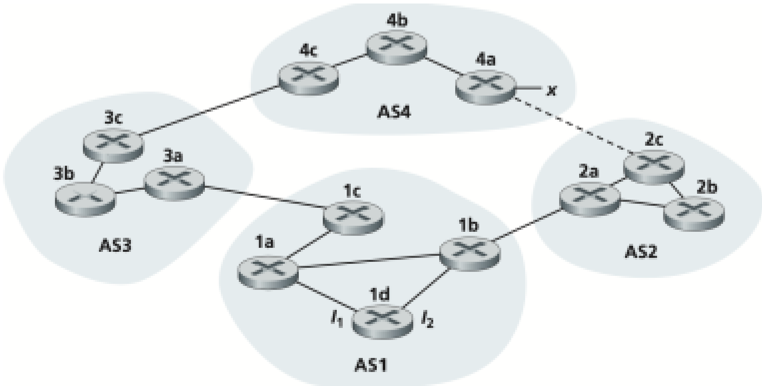
b. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.

- count-to-infinity problem이 발생할 수 있다. x와 y 사이의 링크가 60으로 증가했기 때문에, x와 y는 서로를 통해 계속 갱신하며 무한대로 증가하는 과정에 빠질 수 있다.

c. How do you modify c(y,z) such that there is no count-to-infinity problem at all if c(y,x) changes from 4 to 60?

- c(y,z)를 끊는다.

P14. Consider the network shown below. Suppose AS3 and AS2 are running OSPF for their intra-AS routing protocol. Suppose AS1 and AS4 are running RIP for their intra-AS routing protocol. Suppose eBGP and iBGP are used for the inter-AS routing protocol. Initially suppose there is no physical link between AS2 and AS4.



a. Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?

- eBGP

b. Router 3a learns about x from which routing protocol?

iBGP

c. Router 1c learns about x from which routing protocol?

- eBGP

d. Router 1d learns about x from which routing protocol?

- iBGP