

제24강 거리 벡터 라우팅

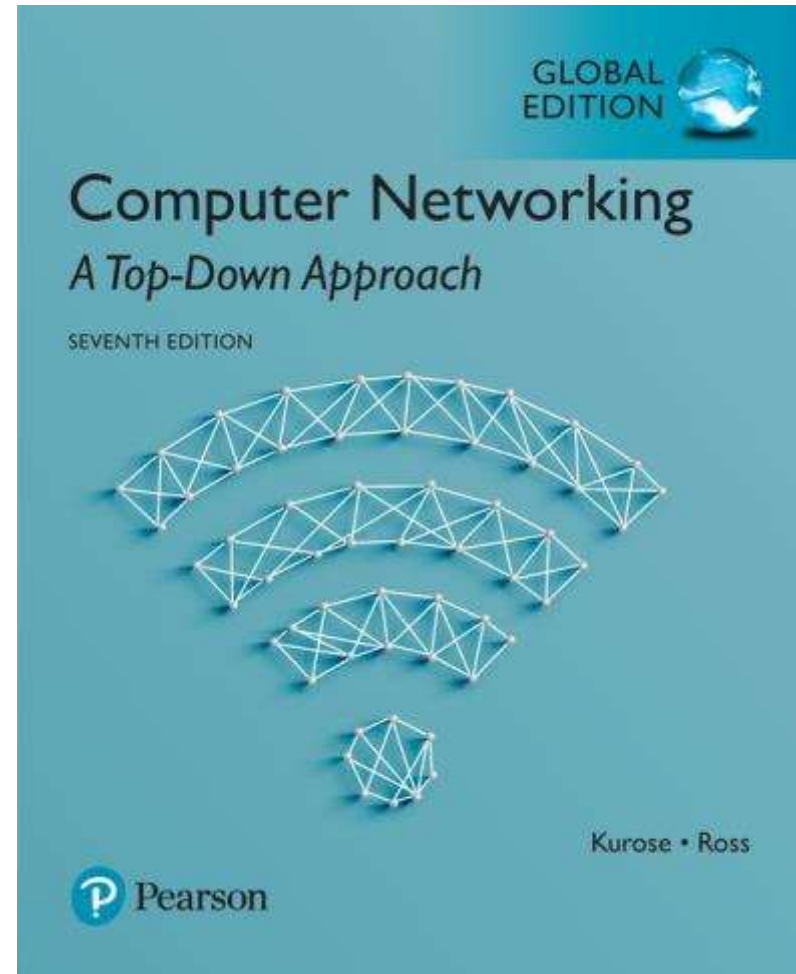
*Computer Networking: A
Top Down Approach*

컴퓨터 네트워크
(2019년 1학기)

박승철교수

한국기술교육대학교
컴퓨터공학부

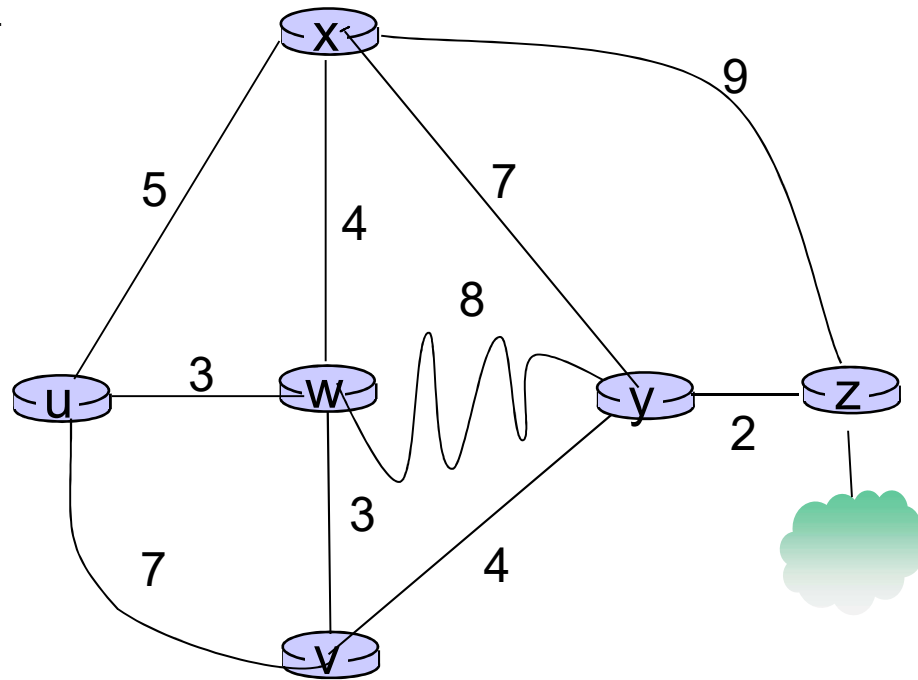
©



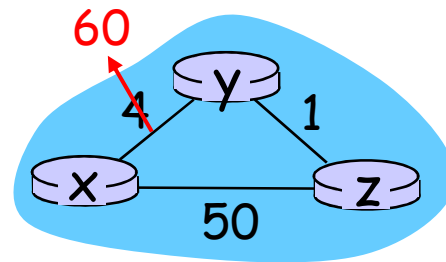
Pre-study Test :

(문제1) 거리 벡터(distance vector) 라우팅 알고리즘은 이웃 라우터가 특정 목적지에 대해 알려준 경로 비용과 이웃 라우터까지 링크 비용의 합으로 해당 목적지까지의 경로 비용을 산정한다.

다음 그림에서 목적지 Z에 대해 라우터 U에게 X는 9,
W는 9, V는 6이라 알려줄 때,
U의 라우팅 테이블에서 Z에 대한
다음 라우터는 무엇인가 ?



(문제2) 거리 벡터(distance vector) 라우팅 알고리즘은 목적지로의 최소 경로 비용에 변경이 발생하면 이웃 라우터에게 통보하고, 변경을 통보 받은 이웃 라우터는 목적지로의 최소 경로 비용을 재계산한다.
현재 라우터 z는 라우터 y에게 x까지 최소 비용을 5로 통보해둔 상태다.
x-y 링크 비용이 4 → 60으로 변경될 때 라우터 y에게 무슨 문제가 발생하는가?
이 문제를 어떻게 해결할 수 있는가?



Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let

$d_x(y) :=$ cost of least-cost path from x to y

then

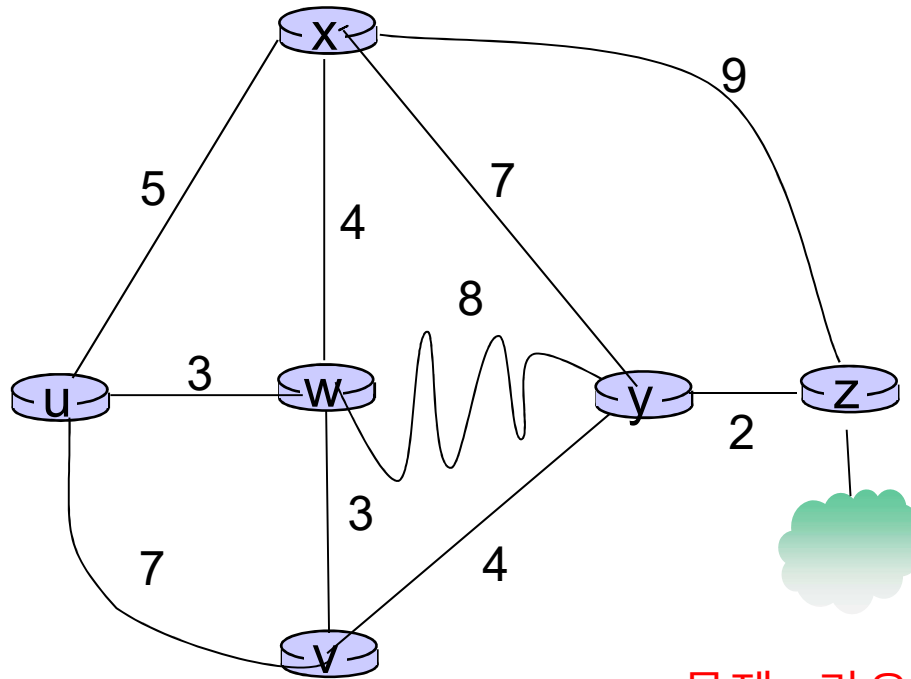
$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

cost from neighbor v to destination y

cost to neighbor v

\min taken over all neighbors v of x

Distance vector algorithm



$$d_x(z) = 9$$

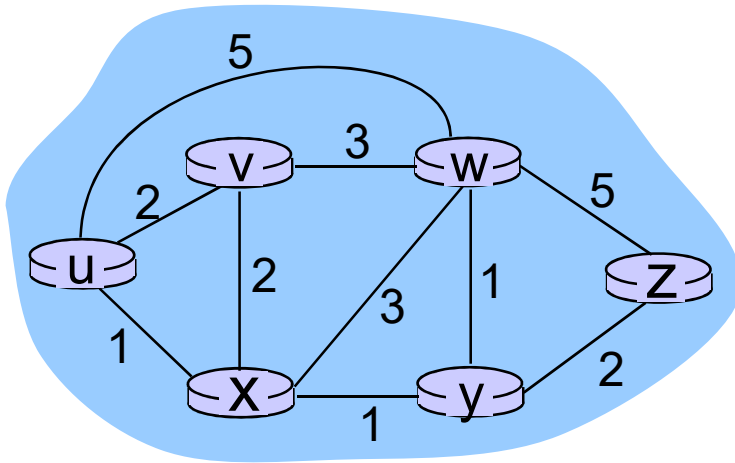
$$d_w(z) = 9$$

$$d_v(z) = 6$$

$$d_u(z) = \min\{c(u,x) + d_x(z), \\ c(u,w) + d_w(z), \\ c(u,v) + d_v(z)\} \\ = \min\{5+9, 3+9, 7+6\} = 12$$

문제 : 라우터 u에서 라우터z에 연결된
망 주소에 대한 라우팅 테이블의 다음 홉 주소는
무엇인가?

Bellman-Ford example



$d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$ 일 때
라우터 u에서 목적지 주소가
라우터 z에 연결된 망 주소에 대한
라우팅 테이블의 다음 홉 주소를
구하는 과정을 설명하라.

Distance vector algorithm

key idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

- ❖ under minor, natural conditions, the estimate $D_x(y)$ converge to the actual least cost $d_x(y)$

Distance vector algorithm

iterative, asynchronous:

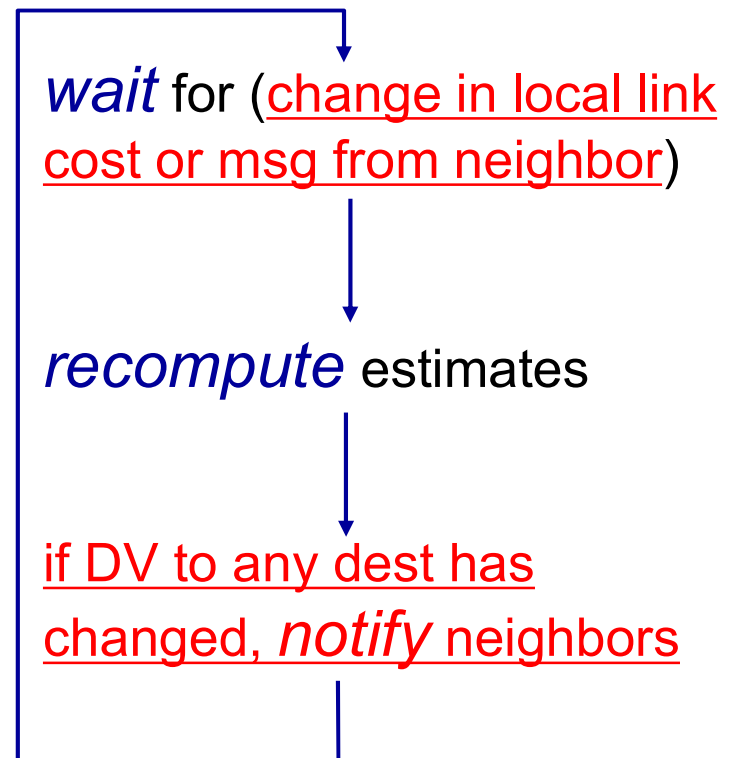
each local iteration
caused by:

- local link cost change
- DV update message from neighbor

distributed:

- each node notifies neighbors *only* when its DV changes
 - neighbors then notify their neighbors if necessary

each node:



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

**node x
table**

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

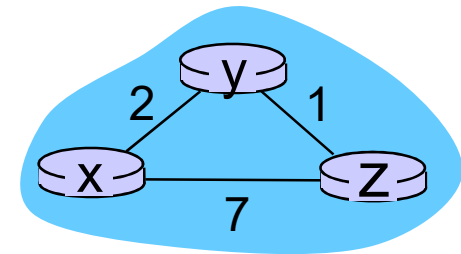
**node y
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

**node z
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	?
	y	2	0	1
	z	7	1	0



time

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

**node x
table**

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

**node y
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

**node z
table**

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

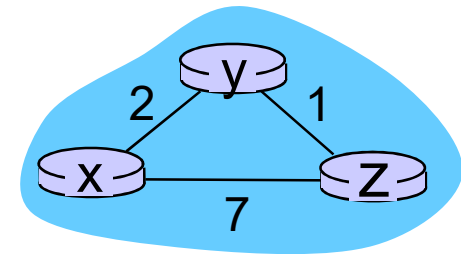
		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

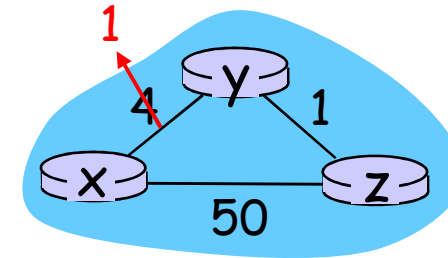


time →

Distance vector: link cost changes

link cost changes:

- ❖ node detects local link cost change
- ❖ updates routing info, recalculates distance vector
- ❖ if DV changes, notify neighbors



“good
news
travels
fast”

t_0 : y detects link-cost change, updates its DV, informs its neighbors. ($D_y(x) = 1 (\leftarrow 4)$)

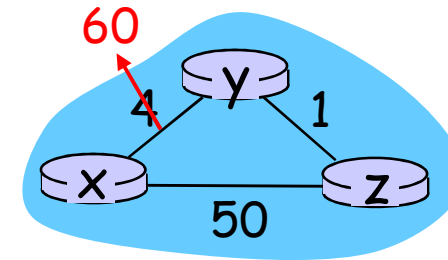
t_1 : z receives update from y ($D_y(x) = 1$), updates its table, computes new least cost to x ($c(z,y) + D_y(x)$), sends its neighbors its DV.

t_2 : y receives z's update, updates its distance table. y's least costs do *not* change ($\min\{D_y(x), c(y,z) + D_z(x)\} = 1$) so y *does not* send a message to z.

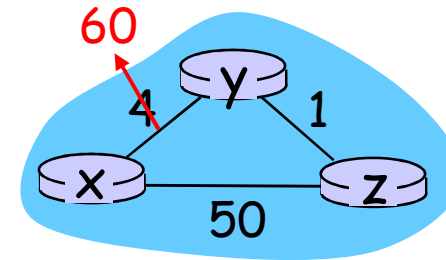
Distance vector: link cost changes

link cost changes:

- ❖ node detects local link cost change
($D_y(x) = 60(\leftarrow 4)$)
- ❖ 문제1 : (1) 라우터 z의 라우팅 테이블에서 라우터 x로의 경로는 ?
(2) 라우터 y의 라우팅 테이블에서 라우터 x로의 경로는 ?
- ❖ 문제2 : 라우터 z의 라우팅 테이블에서 라우터 x로의 정상적인 경로가 언제 결정될까?



Distance vector: link cost changes



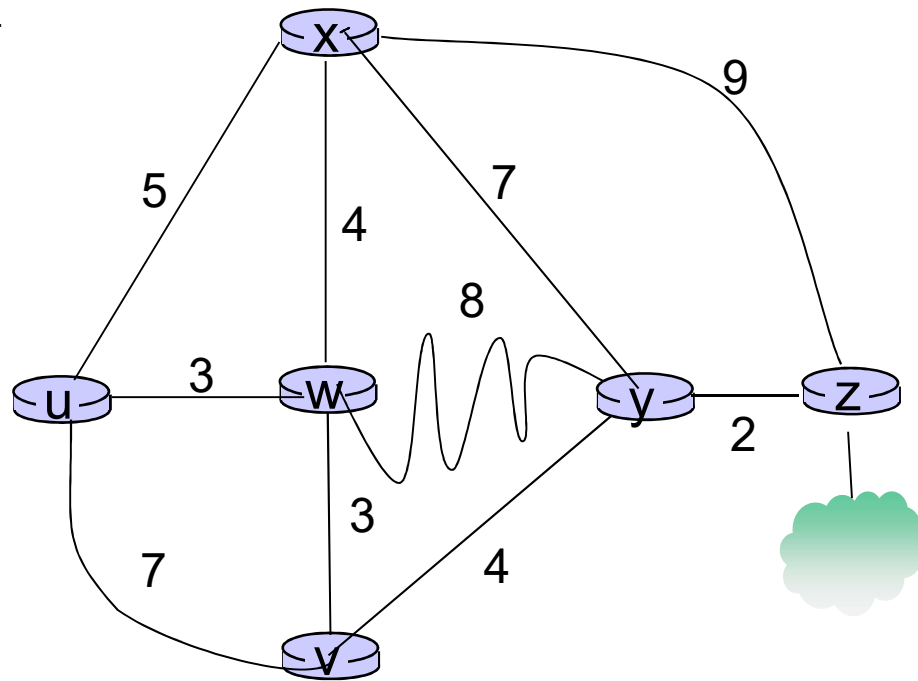
poisoned reverse:

- ❖ If Z routes through Y to get to X :
 - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z) ($D_Z(x) = \infty$ (not 5))
 - $D_Y(x) = 60$ ($\leftarrow ?$)

After-study Test :

(문제1) 거리 벡터(distance vector) 라우팅 알고리즘은 이웃 라우터가 특정 목적지에 대해 알려준 경로 비용과 이웃 라우터까지 링크 비용의 합으로 해당 목적지까지의 경로 비용을 산정한다.

다음 그림에서 목적지Z에 대해 라우터 U에게 X는 9,
W는 9, V는 6이라 알려줄 때,
U의 라우팅 테이블에서 Z에 대한
다음 라우터는 무엇인가 ?



(문제2) 거리 벡터(distance vector) 라우팅 알고리즘은 목적지로의 최소 경로 비용에 변경이 발생하면 이웃 라우터에게 통보하고, 변경을 통보 받은 이웃 라우터는 목적지로의 최소 경로 비용을 재계산한다.
현재 라우터 z는 라우터 y에게 x까지 최소 비용을 5로 통보해둔 상태다.
x-y 링크 비용이 4 → 60으로 변경될 때 라우터 y에게 무슨 문제가 발생하는가?
이 문제를 어떻게 해결할 수 있는가?

