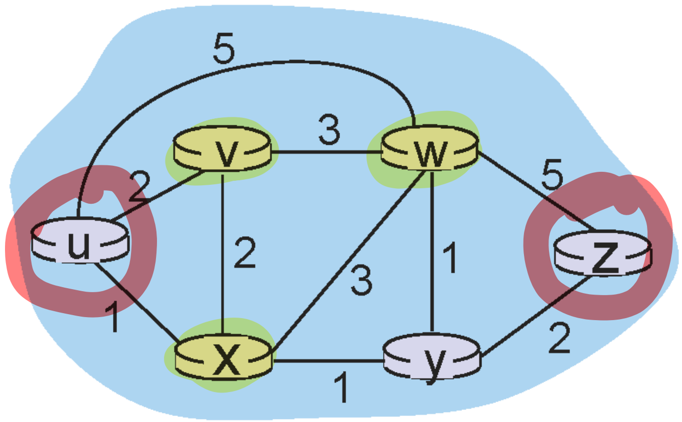
Bellman-Ford example

*자기자신의 Distance Vector값이 변경됐으면
주변에 전달한다!



clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

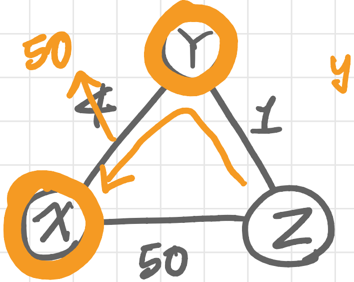
$$
\begin{aligned}
d_u(z) = \min \{ & c(u,v) + d_v(z), \\
& c(u,x) + d_x(z), \\
& c(u,w) + d_w(z) \} \\
= \min \{ & 2 + 5, \\
& 1 + 3, \\
& 5 + 3\} = 4
\end{aligned}
$$

distance 정보를
넘겨주기 때문에
distance 알고있습니다!

node achieving minimum is next
hop in shortest path, used in forwarding table

Distance vector algorithm

- $D_x(y)$ = estimate of least cost from x to y
  - x maintains distance vector $\mathbf{D_x} = [D_x(y): y \in N]$
- node x:
  - knows cost to each neighbor v: $c(x,v)$
  - maintains its neighbors' distance vectors. For each neighbor v, x maintains $\mathbf{D_v} = [D_v(y): y \in N]$



y한테는 Z→X로가는길이 ∞이라고 알려줘야한다??!

시간 →

$d_x(z) = \min(C(x,v) + d_v(z))$

$d_x(y) = \min(C(x,v) + d_v(y))$

C(x,y)=50

Link Cost에 변화 C(y,x)=50

X

	x	y	z
x	0	4	50
y			
z			

Y

	x	y	z
x			
y	4	0	1
z			

Z

	x	y	z
x			
y			
z	50	1	0

X:

	x	y	z
x	0	4	5
y	4	0	1
z	50	1	0

Y:

	x	y	z
x	0	4	50
y	4	0	1
z	50	1	0

Z:

	x	y	z
x	0	4	50
y	4	0	1
z	5	1	0

X:

	x	y	z
x	0	4	5
y	4	0	1
z	5	1	0

재계산필요!

$d_y(x) = \min[\overbrace{\underset{4}{C(y,x)} + \underset{0}{d_x(x)}}^{4}, \overbrace{\underset{1}{C(y,z)} + \underset{5}{d_z(x)}}^{6}]$ (4 vs 6)

Y:

	x	y	z
x	0	4	5
y	4	0	1
z	5	1	0

50

→ 6 0 1

8 0 1

⋮

Z:

	x	y	z
x	0	4	5
y	4	0	1
z	5	1	0

No update → Stop!

	x	y	z
x			
y			
z			

	x	y	z
x			
y			
z	7	1	0

9 1 0

⋮

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:

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

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

Distance vector algorithm $\rightarrow d_x(y) = \min_v [c(x,v) + d_v(y)]$

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:

wait for (change in local link cost or msg from neighbor)

recompute estimates

if DV to any dest has changed, *notify* neighbors

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\} = \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\} = \min\{2+1\ ,\ 7+0\} = 3$$

node x table

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

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

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

node y table

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

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

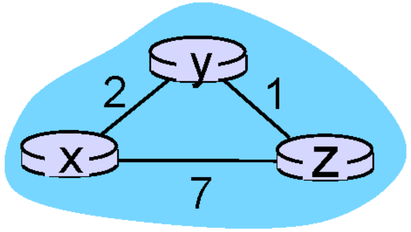
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

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

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

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



...

time