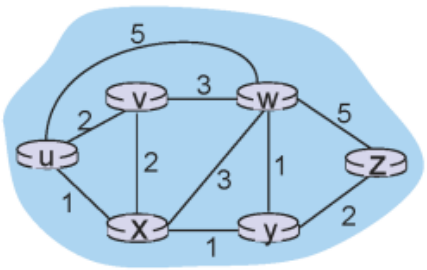
네트워크계층5

작성일시	@2022년 10월 31일 오후 8:47
강의날짜	@2022/10/31
편집일시	@2022년 10월 31일 오후 9:39
분야	네트워크
공부유형	스터디 그룹
복습	☐
태그	

Bellman-Ford example



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

B-F equation says:

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

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

Network Layer 4-85

거리를 구하는데 결국 recursive하게 구한다

직관적이지는 않음

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 $\mathbf{D}_v = [D_v(y): y \in N]$

Network Layer 4-86

각노드는 자기자신이 알고있는 벡터를 이웃하게 넘겨주고 그런 정보를 넘겨주면 또 이웃한테 전달

이러한 업데이트가 계속되다 보면 업데이트가 그만되고 stable해지는 상황이 옴

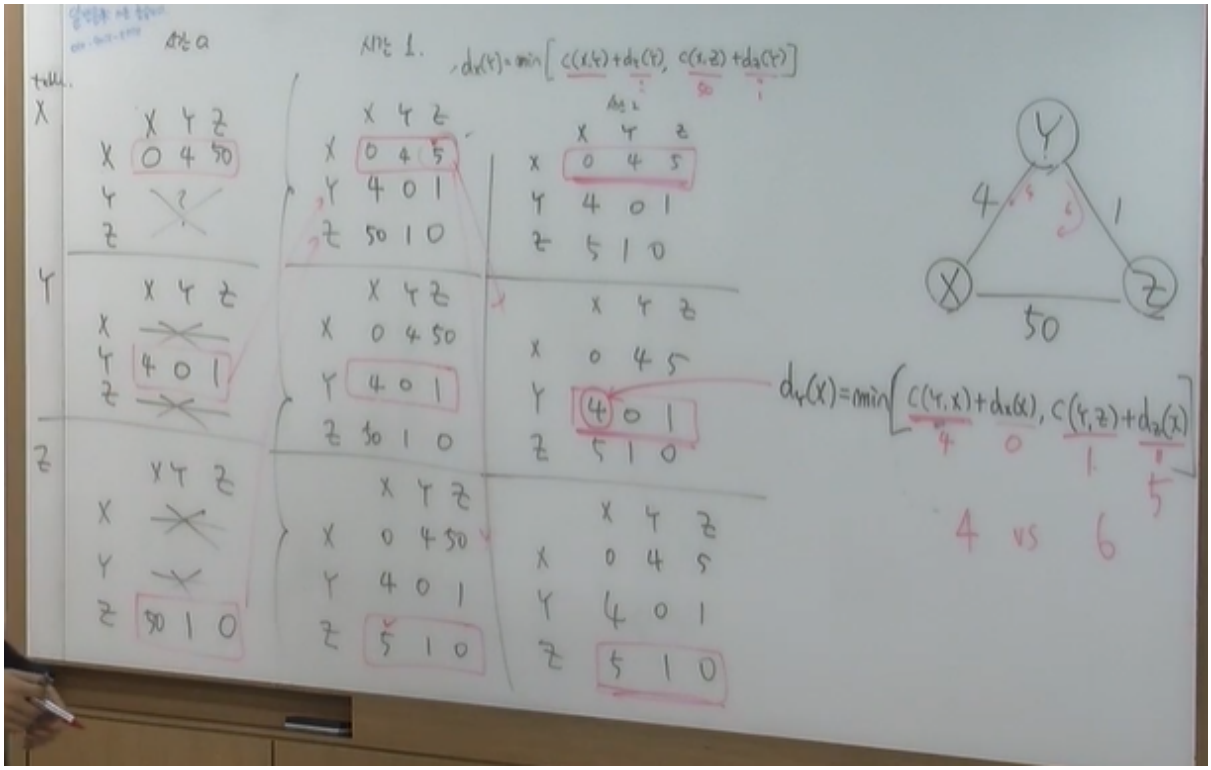
Distance vector algorithm

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

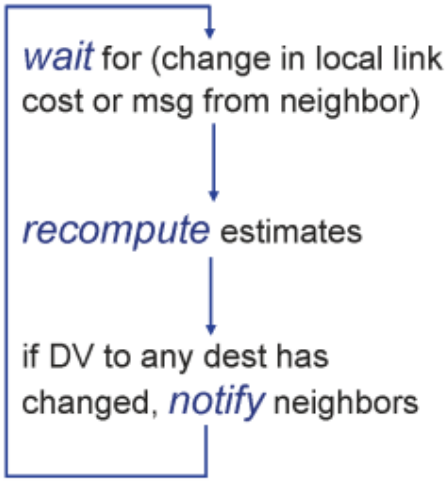
iterative, asynchronous:
each local iteration caused by:

- 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:

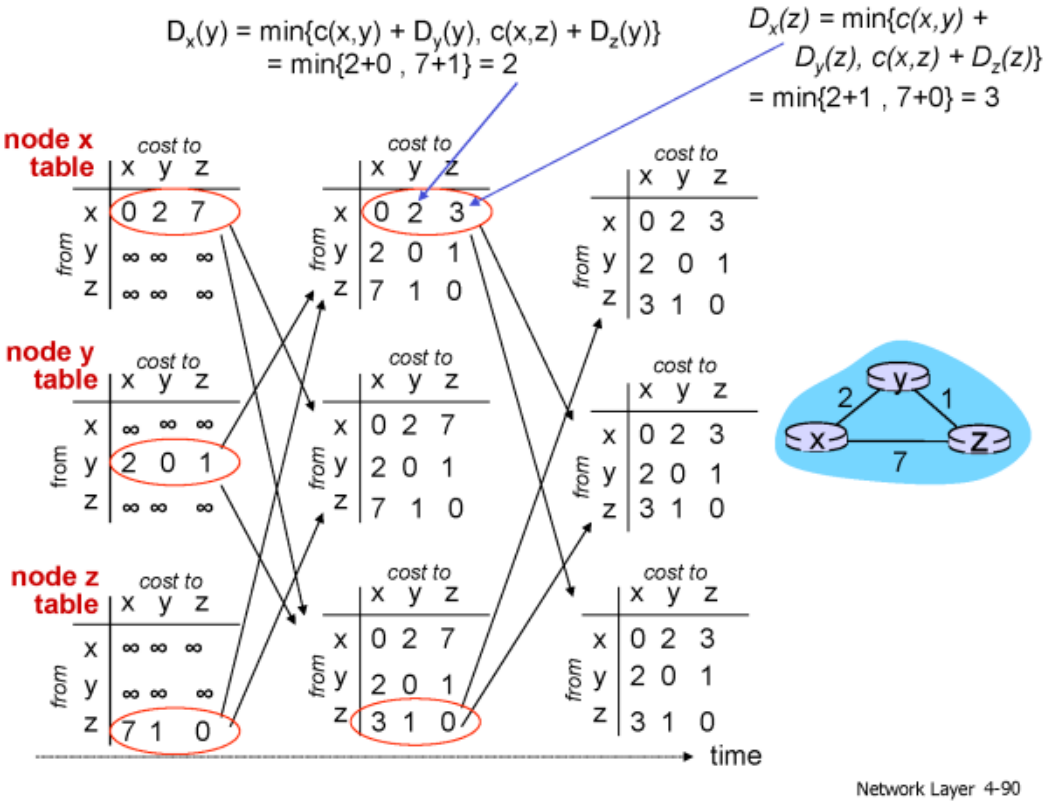


Network Layer 4-88

링크 cost가 낮아지면 바로 커트됨

count infinity : 링크 cost가 높아지면 핑퐁핑퐁 하나씩 높여가며 시간이 많이 걸림

따라서 Dx(Z)가 무한대라고 알려줘야함



하나의 네트워크에 국한되어서 그 내부의 브로드캐스트 되어서 내부의 그림을 알 때 사용