Distance-Vector Routing

COS 461: Computer Networks
Spring 2010 (MW 3:00-4:20 in COS 105)

Michael Freedman
http://www.cs.princeton.edu/courses/archive/spring10/cos461/
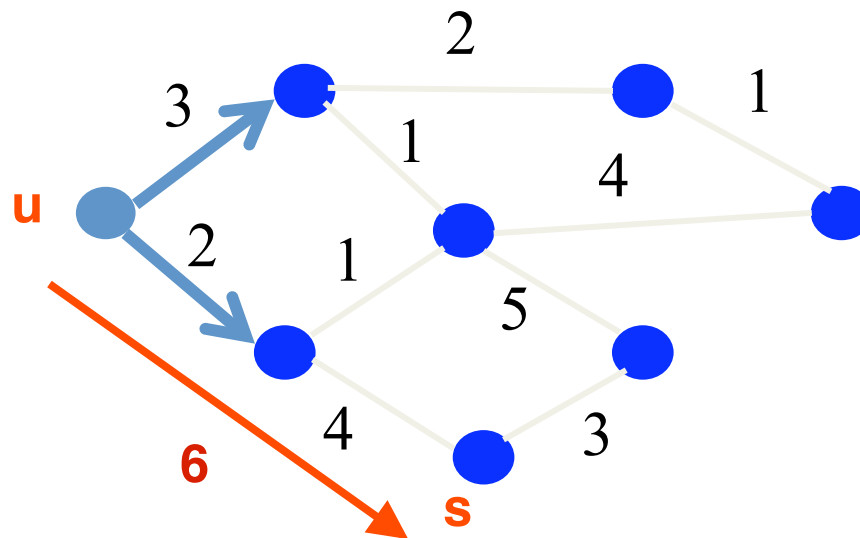
Shortest-Path Routing

- Path-selection model
  - Destination-based
  - Load-insensitive (e.g., static link weights)
 - Minimum hop count or sum of link weights
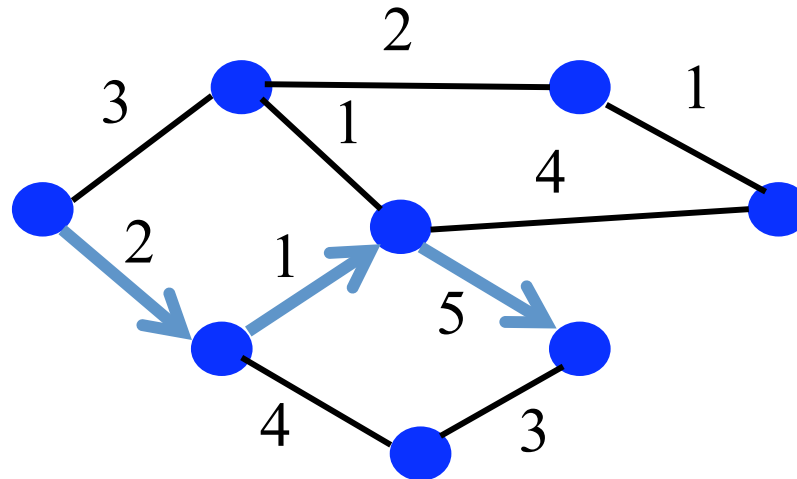
Shortest-Path Problem

- Compute: *path costs* to all nodes
  - From a given source u to all other nodes
  - Cost of the path through each outgoing link
  - Next hop along the least-cost path to s

Bellman-Ford Algorithm

- Define distances at each node x
  - $d_x(y)$ = cost of least-cost path from x to y
- Update distances based on neighbors
  - $d_x(y) = \min \{c(x,v) + d_v(y)\}$ over all neighbors v
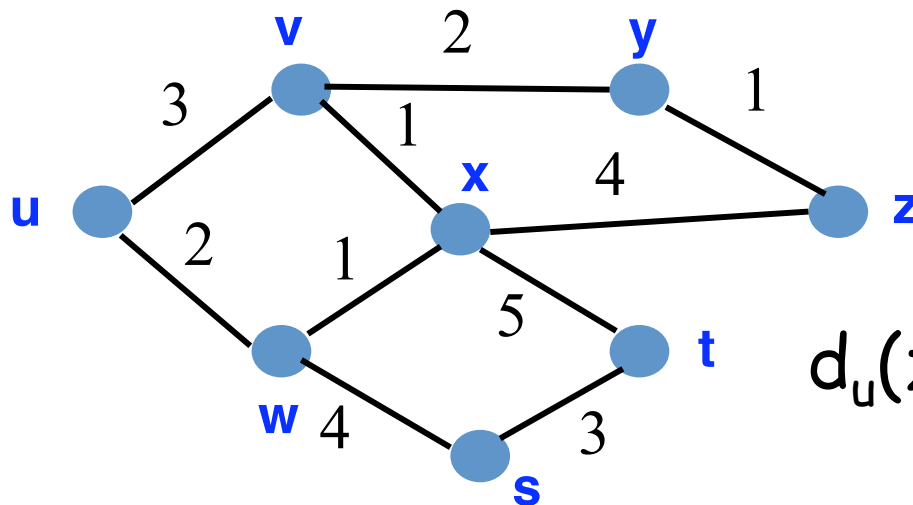
$$d_u(z) = \min\{c(u,v) + d_v(z),$$
$$c(u,w) + d_w(z)\}$$

Distance Vector Algorithm

- $c(x,v)$ = cost for direct link from x to v
  - Node x maintains costs of direct links $c(x,v)$
- $D_x(y)$ = estimate of least cost from x to y
  - Node x maintains distance vector $\mathbf{D_x} = [D_x(y): y \in N]$
- Node x maintains its neighbors' distance vectors
  - For each neighbor v, x maintains $\mathbf{D_v} = [D_v(y): y \in N]$
- Each node v periodically sends D_v to its neighbors
  - And neighbors update their own distance vectors
  - $D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\}$ for each node $y \in N$
- Over time, the distance vector D_x converges

Distance Vector Algorithm

Iterative, asynchronous:

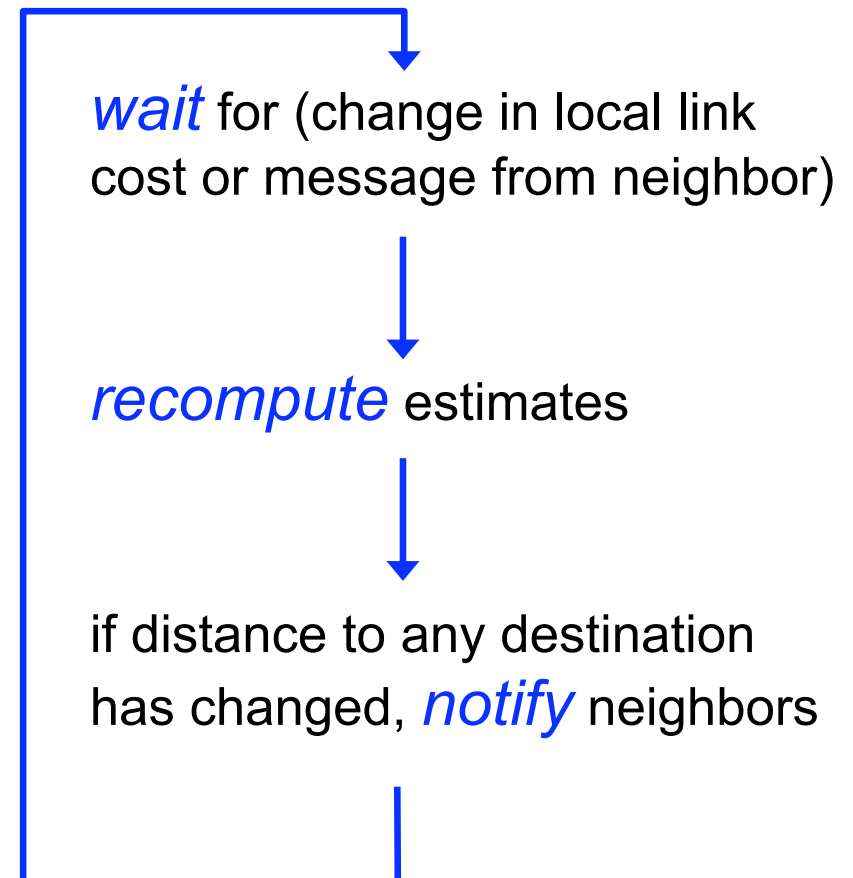each local iteration caused by:

- Local link cost change
- Distance vector 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 message from neighbor)

↓

recompute estimates

↓

if distance to any destination has changed, *notify* neighbors

Distance Vector Example: Step 1

Optimum 1-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	∞	–	C	∞	–
D	∞	–	D	3	D
E	2	E	E	∞	–
F	6	F	F	1	F
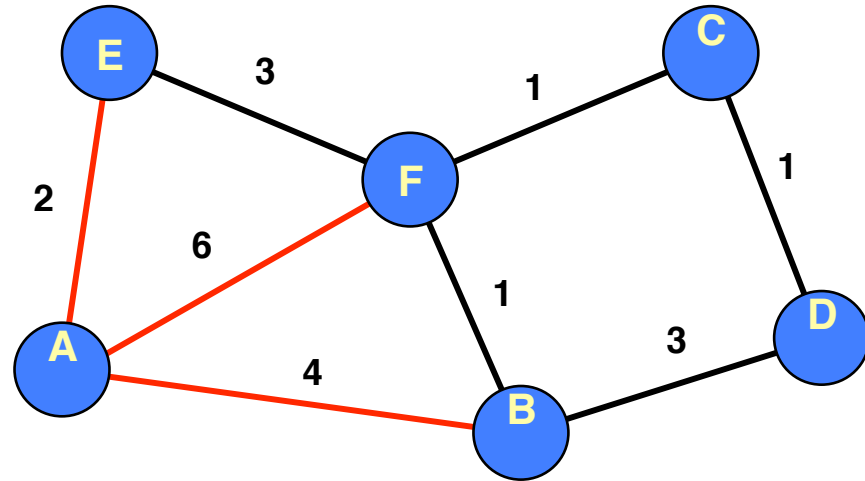
Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	∞	–	A	∞	–	A	2	A	A	6	A
B	∞	–	B	3	B	B	∞	–	B	1	B
C	0	C	C	1	C	C	∞	–	C	1	C
D	1	D	D	0	D	D	∞	–	D	∞	–
E	∞	–	E	∞	–	E	0	E	E	3	E
F	1	F	F	∞	–	F	3	F	F	0	F

7

Distance Vector Example: Step 2

Optimum 2-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	7	F	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F
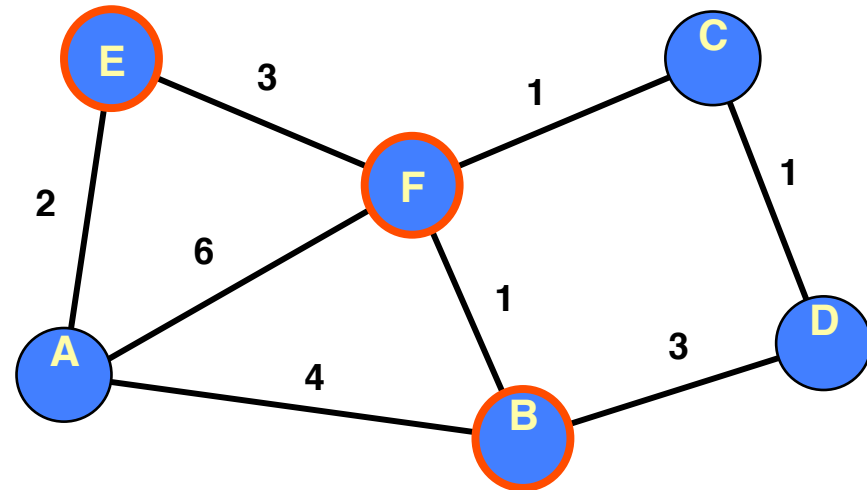
Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	7	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	∞	–	D	2	C
E	4	F	E	∞	–	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

Distance Vector Example: Step 3

Optimum 3-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	6	E	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F
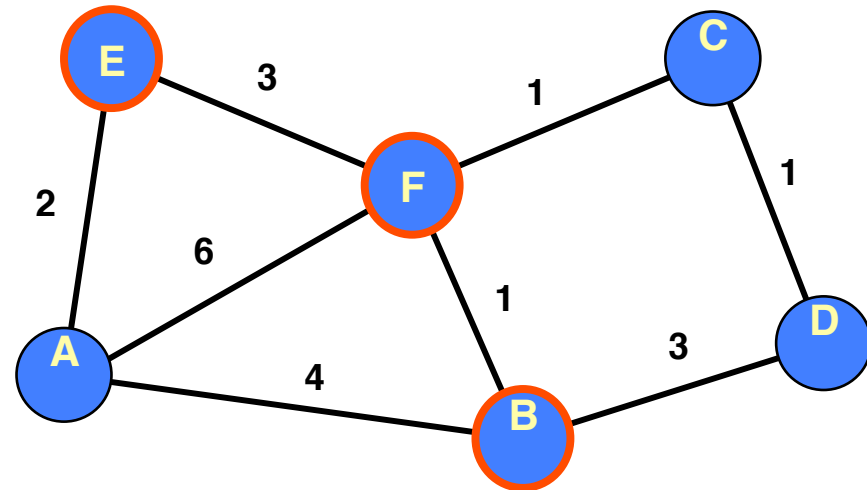
| Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	6	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	5	F	D	2	C
E	4	F	E	5	C	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

9

Distance Vector: Link Cost Changes

Link cost changes:

- Node detects local link cost change
- Updates the distance table
- If cost change in least cost path, notify neighbors

"good news travels fast"

D^Y	via X	Z
X	④	6

D^Z	via X	Y
X	50	⑤

D^Y	X	Z
X	①	6

D^Z	X	Y
X	50	⑤

D^Y	X	Z
X	①	6

D^Z	X	Y
X	50	②

D^Y	X	Z
X	①	3

D^Z	X	Y
X	50	②

algorithm terminates

c(X,Y) change

time t_0 t_1 t_2
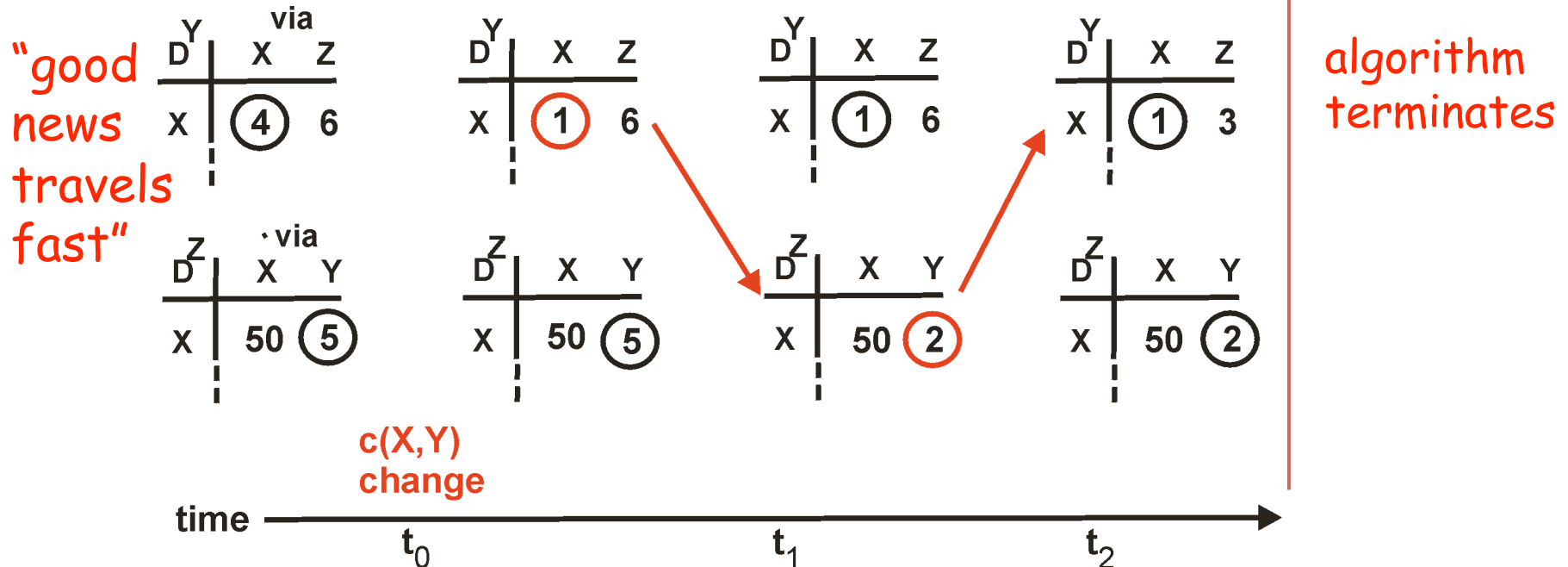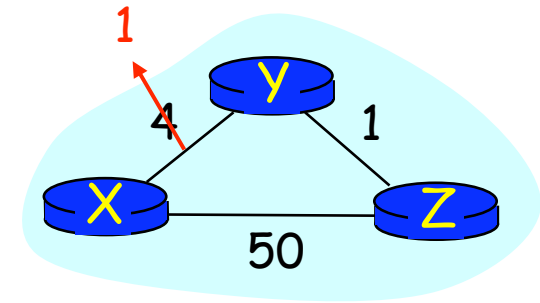
Distance Vector: Link Cost Changes
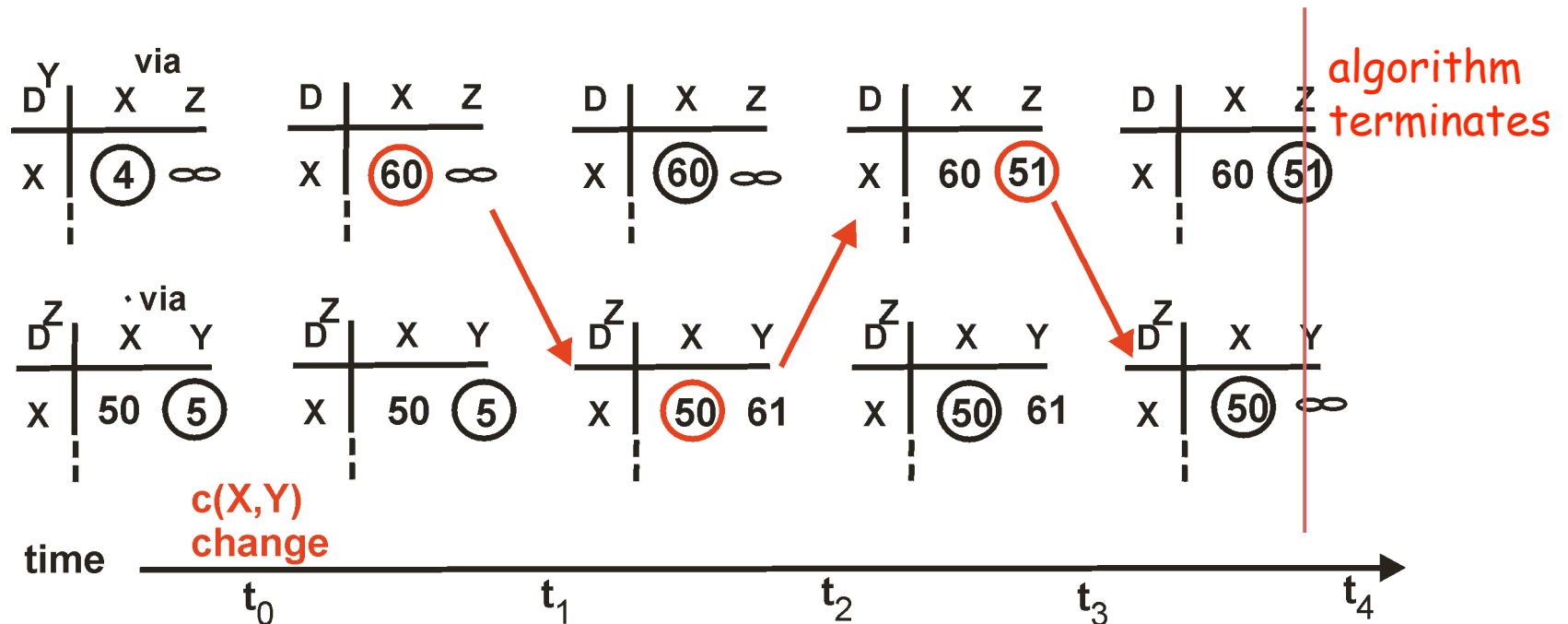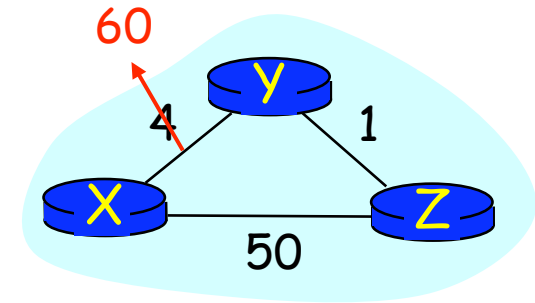
Link cost changes:

- Good news travels fast

- Bad news travels slow - "count to infinity" problem!
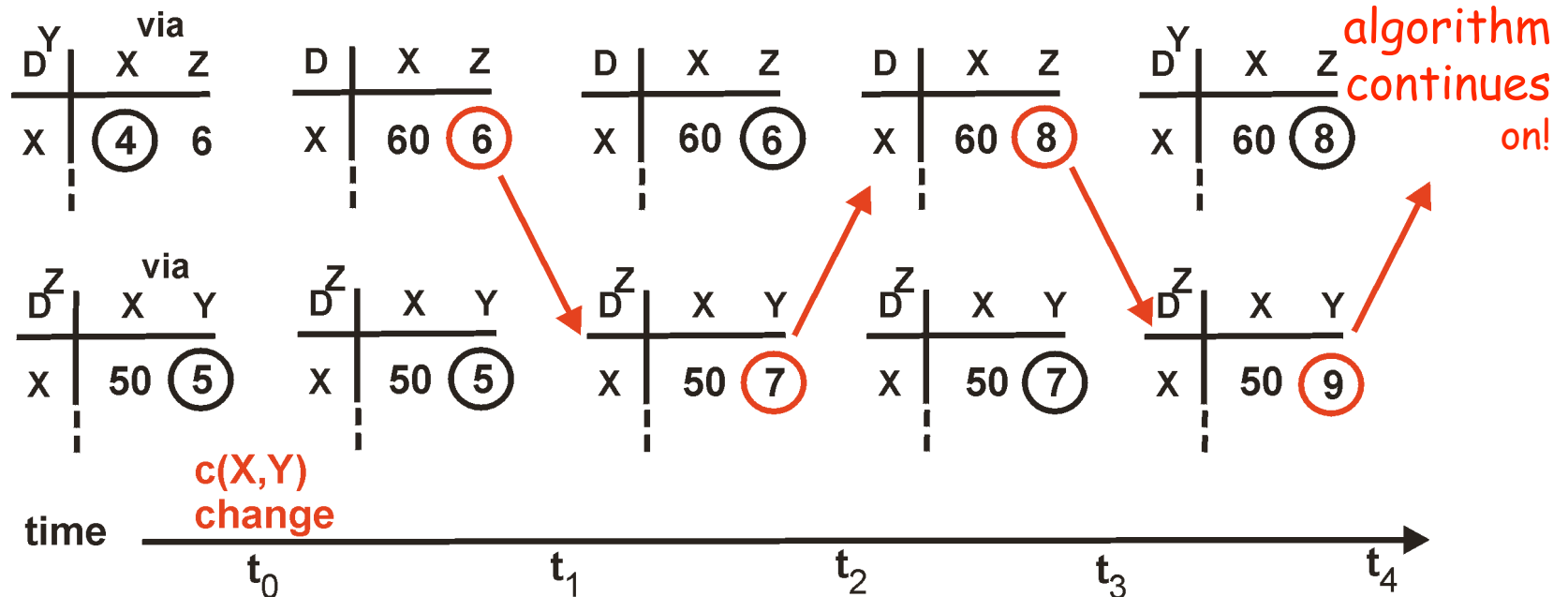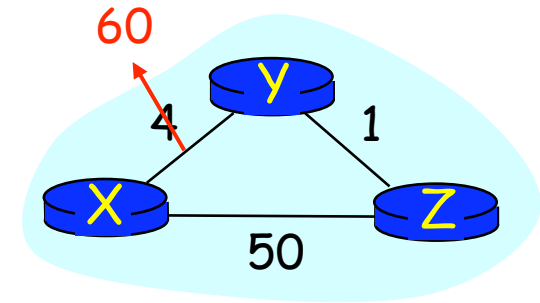
Distance Vector: Poison Reverse

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)

- Still, can have problems when more than 2 routers are involved

Routing Information Protocol (RIP)

- ## Distance vector protocol
  - Nodes send distance vectors every 30 seconds
  - ... or, when an update causes a change in routing

- ## Link costs in RIP
  - All links have cost 1
  - Valid distances of 1 through 15
  - ... with 16 representing infinity
  - Small "infinity" $\rightarrow$ smaller "counting to infinity" problem

- ## RIP is limited to fairly small networks
  - E.g., used in the Princeton campus network

Comparison of LS and DV Routing

Message complexity

- <u>LS</u>: with n nodes, E links, O(nE) messages sent
- <u>DV</u>: exchange between neighbors only

Speed of Convergence

- <u>LS</u>: relatively fast
- <u>DV</u>: convergence time varies
  - May be routing loops
  - Count-to-infinity problem

Robustness: what happens if router malfunctions?

<u>LS:</u>

- Node can advertise incorrect *link* cost
- Each node computes only its *own* table

<u>DV:</u>

- DV node can advertise incorrect *path* cost
- Each node's table used by others (error propagates)

Similarities of LS and DV Routing

- Shortest-path routing
  - Metric-based, using link weights
  - Routers share a common view of how good a path is

- As such, commonly used *inside* an organization
  - RIP and OSPF are mostly used as *intra*domain protocols
  - E.g., Princeton uses RIP, and AT&T uses OSPF

- But the Internet is a "network of networks"
  - How to stitch the many networks together?
  - When networks may not have common goals
  - … and may not want to share information