# EECS 489 Discussion 8

# Announcements

- Assignment 3 is due next week

# Q1 True or False

Link-State (LS) routing algorithm is a centralized routing algorithm, which requires global knowledge about the network.

**True**

# Q2 True or False

Both Link-State and Distance-Vector are shortest-path based routing.

**True**

# Distance-Vector in action

$D[x,y] = \min\{c(x,y) + D[y,y],\ c(x,z) + D[z,y]\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D[x,z] = \min\{c(x,y) + D[y,z],\ c(x,z) + D[z,z]\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

$$D_x(y) = \min_{v}\{c(x,v) + D_v(y)\}$$
$$v \in \mathbf{Neighbor}(x) \cup \{x\}$$

$x$'s 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 |

$y$'s 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 |

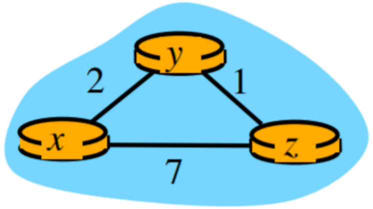
$z$'s table

| from \ cost to | $x$ | $y$ | $z$ |
|---|---|---|---|
| $x$ | ∞ | ∞ | ∞ |
| $y$ | ∞ | ∞ | ∞ |
| $z$ | 7 | 1 | 0 |

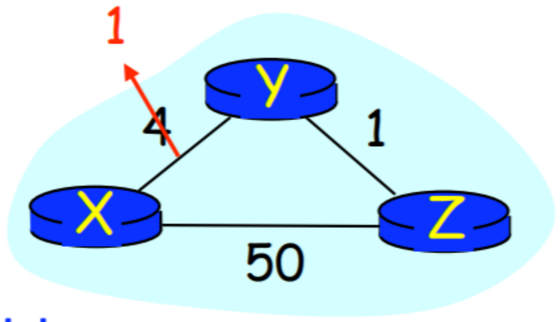
| from \ cost to | $x$ | $y$ | $z$ |
|---|---|---|---|
| $x$ | 0 | 2 | 7 |
| $y$ | 2 | 0 | 1 |
| $z$ | 3 | 1 | 0 |

| from \ cost to | $x$ | $y$ | $z$ |
|---|---|---|---|
| $x$ | 0 | 2 | 3 |
| $y$ | 2 | 0 | 1 |
| $z$ | 3 | 1 | 0 |

time



# Distance Vector: Link Cost Changes

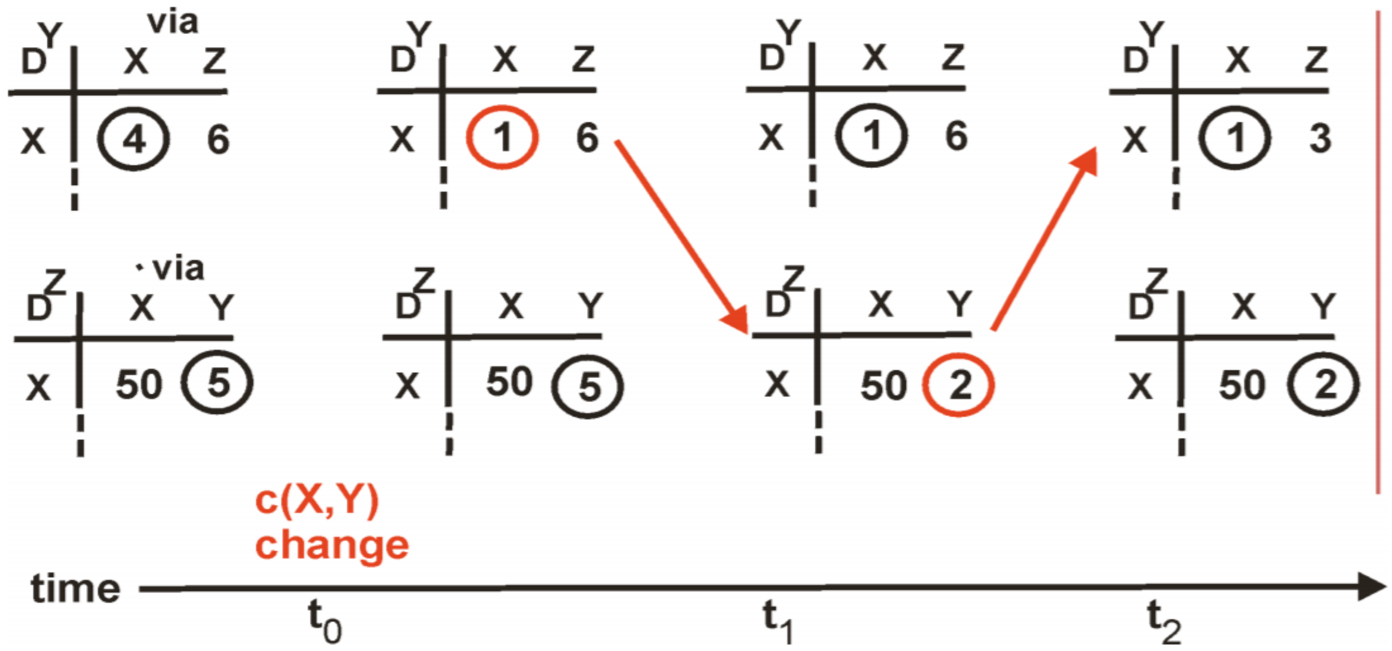



## Link cost changes:

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

## View of X (about neighbor y and z's routing tables)

"Good news travels fast"



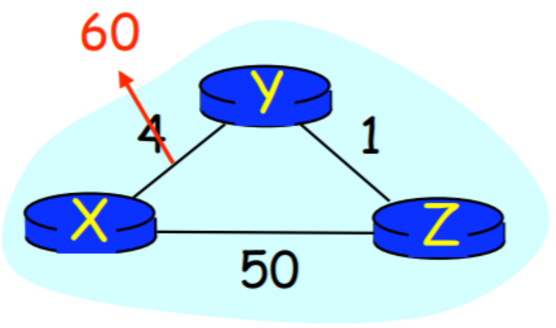

algorithm terminates

Circled entry is least cost

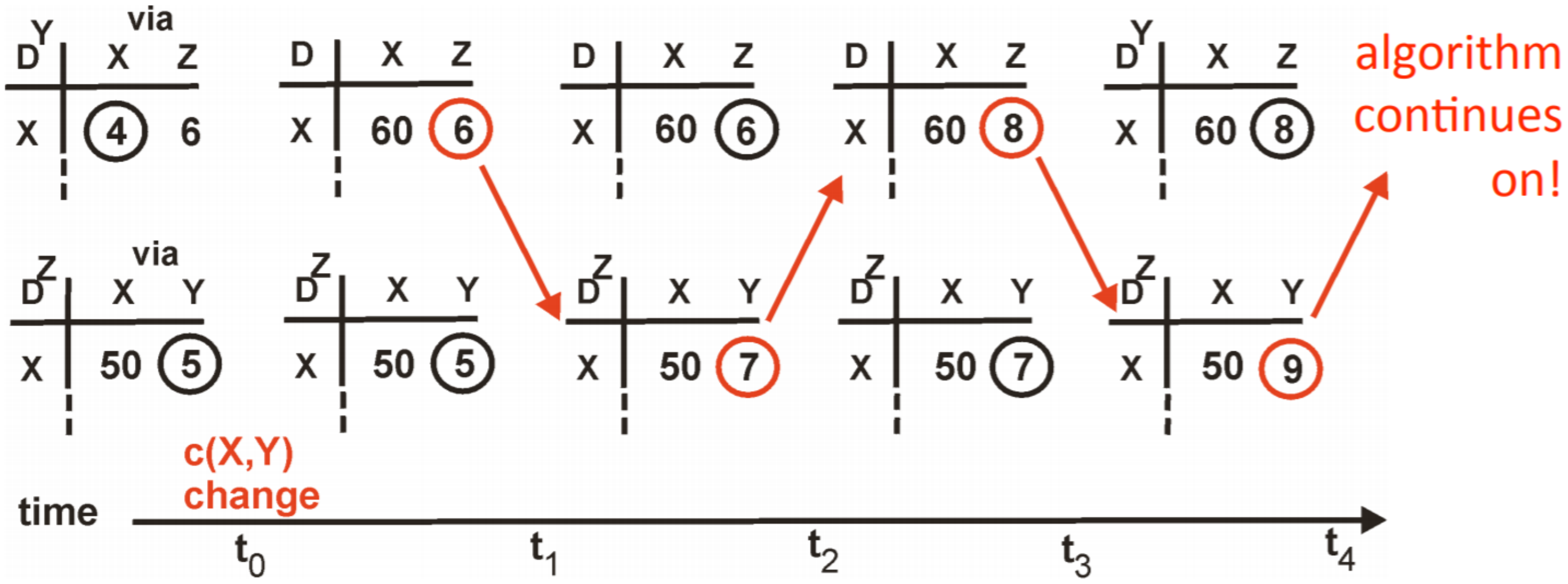
# Distance Vector: Link Cost Changes

## Link cost changes:

- Good news travels fast
- Bad news travels slow - “count to infinity” problem!




## View of X (about neighbor y and z’s routing tables)



# Q2.1

Consider the count-to-infinity problem in DV routing. Will the count-to-infinity problem occur if we **decrease the cost of a link**? Why?

**No.**

**Decreasing the cost of a link won’t cause a routing loop**

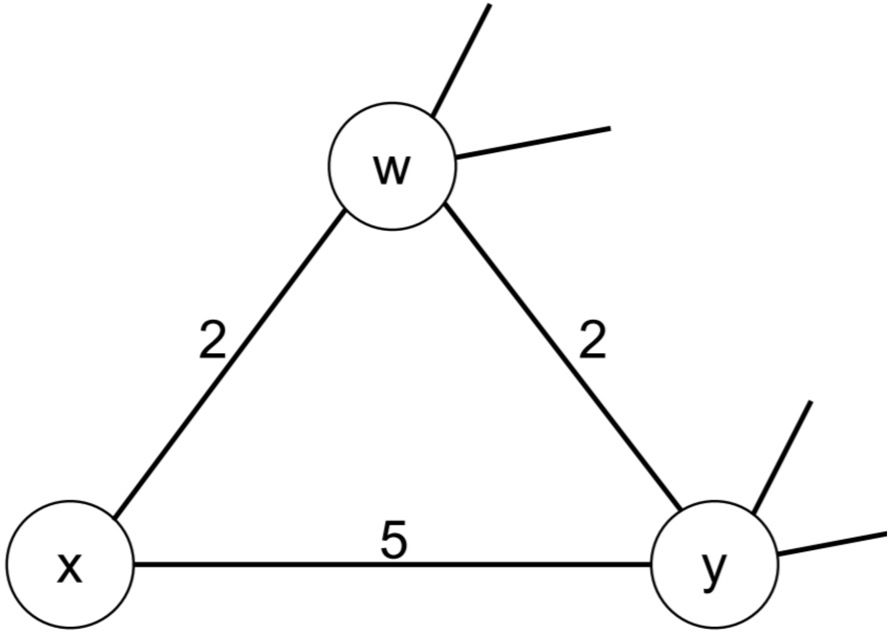
# Q2.2

Consider the count-to-infinity problem in DV routing. Will the count-to-infinity problem occur if **we connect two nodes which do not have a link**? Why?

**No**

# Q3

Consider the network fragment shown below. Node **w** has minimum-cost path to destination **u** (not shown) of **5**, and node **y** has a minimum-cost path to **u** of **6**. The complete paths from **w** and **y** to u are not shown. All link costs in the network have strictly positive integer values.

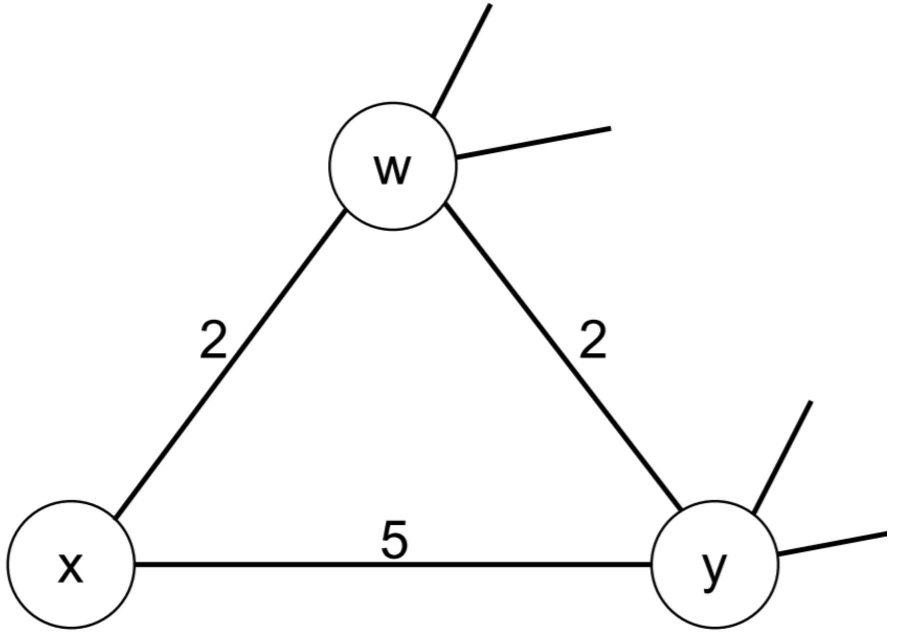



$$D_w(u) = 5$$
$$D_y(u) = 6$$

# Q3

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

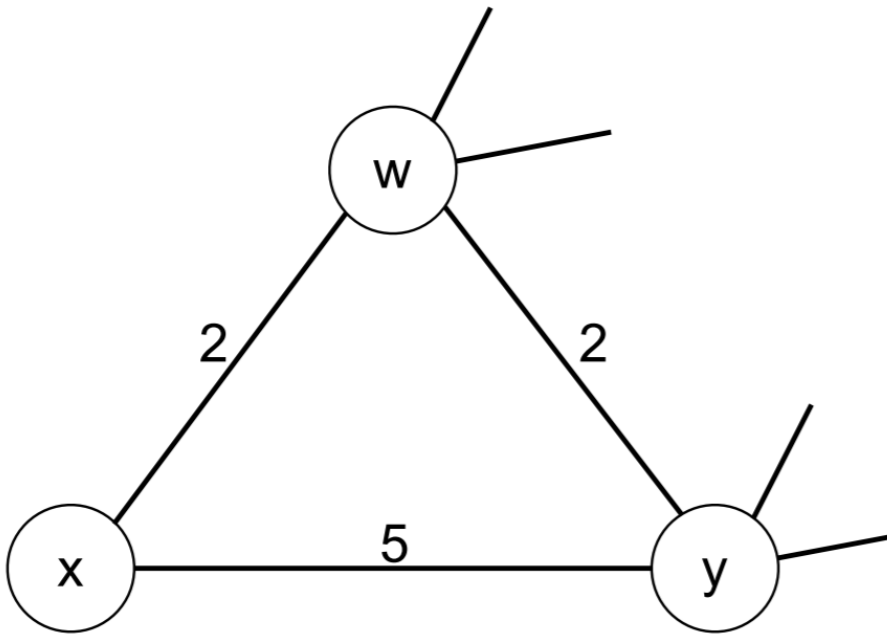
$$D_x(w) = 2$$
$$D_x(y) = 4$$
$$D_x(u) = 7$$




$$D_w(u) = 5$$
$$D_y(u) = 6$$

# Q3

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.

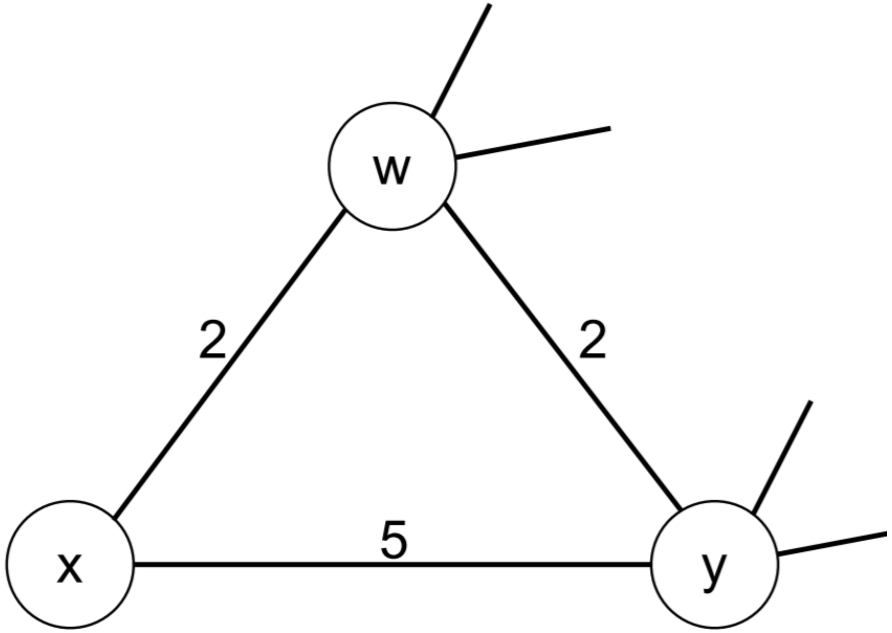



$$\boldsymbol{D_w(u) = 5}$$
$$\boldsymbol{D_y(u) = 6}$$

**Change $c(x, w)$, or**
**Make $c(x, y) < 1$ (non-integer link cost)**

# Q3

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.




$$D_w(u) = 5$$
$$D_y(u) = 6$$

**Make $c(x, y) > 1$**

# Q4

Consider the network shown below and assume the distance-vector protocol with **poisoned reverse** is used to compute the routing information. Further assume that after the routing information has stabilized, the link weight between nodes **A** and **B** changes suddenly from 1 to 200. **Will count-to-infinity occur?**

**If z routes to x through y, z advertises to y that c(z,x) is infinite.**