COMPUTER COMMUNICATION NETWORKS

M Rajasekar

Department of Electronics and Communication Engineering

COMPUTER COMMUNICATION NETWORKS

Routing algorithms: Bellman-Ford Algorithm

M Rajasekar

Department of Electronics and Communication Engineering

Routing algorithms: Bellman-Ford Algorithm

- **Distance vector algorithm (Bellman-Ford algorithm)**
  - Implemented in asynchronous manner
  - It is distributed and iterative $D_x = [D_x(y) \text{: } y \text{ in } N]$
  - For node x the distance vector is given by $D_v = [D_v(y) \text{: } y \text{ in } N]$
  - Neighbor v sends its distance vector to x
  - $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$ for each node y in N

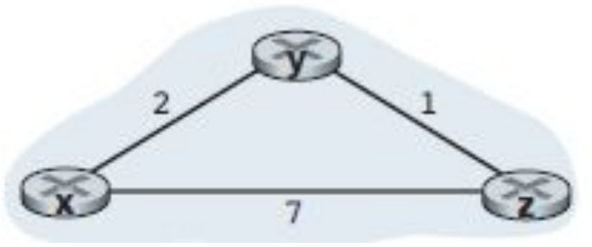
  - Node x updates all neighbors in *N*
  - Above process repeats till convergence

Distance vector algorithm (Bellman-Ford algorithm)

```
Initialization:
   for all destinations y in N:
      D_x(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
   for each neighbor w
      D_w(y) = ? for all destinations y in N
   for each neighbor w
      send distance vector D_x = [D_x(y): y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      D_x(y) = min_v{c(x,v) + D_v(y)}

   if D_x(y) changed for any destination y
      send distance vector D_x = [D_x(y): y in N] to all neighbors

forever
```

Distance vector algorithm (Bellman-Ford algorithm)

- Assuming synchronous update



For node x:

$$D_x(y) = \min\left[c(x,y) + D_y(y), c(x,z) + D_z(y)\right]$$

$$D_x(z) = \min\left[c(x,z) + D_z(z), c(x,y) + D_y(z)\right]$$

For node y:

$$D_y(x) = \min[c(y,x) + D_x(x), c(y,z) + D_z(x)]$$

$$D_y(z) = \min[c(y,x) + D_x(z), c(y,z) + D_z(z)]$$

For node z:

$$D_z(x) = \min\left[c(z,x) + D_x(x), c(z,y) + D_y(x)\right]$$

$$D_z(y) = \min\left[c(z,x) + D_x(y), c(z,y) + D_y(z)\right]$$

Routing algorithms: Bellman-Ford Algorithm

Distance vector algorithm (Bellman-Ford algorithm)

- Assuming synchronous update

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

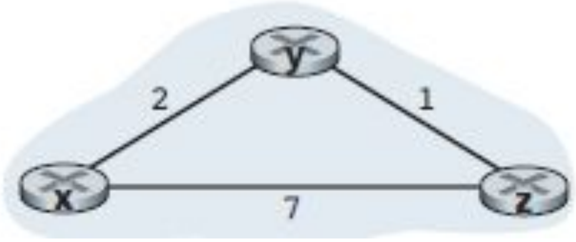
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

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



$$c(x,y) = c(y,x) = 2$$

$$c(y,z) = c(z,y) = 1$$

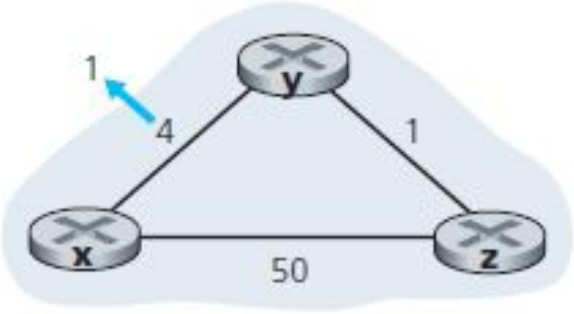
$$c(z,x) = c(x,z) = 7$$

Use the expressions given in previous slide

Routing algorithms: Bellman-Ford Algorithm

Effect of changes in link cost

Initial distance vectors will be the same for x, y and z



	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	4	0	1
$D_z(\cdot)$	5	1	0

$$c(x,y) = c(y,x) = 1$$

$$c(y,z) = c(z,y) = 1$$

$$c(z,x) = c(x,z) = 50$$

Suppose link cost c(x,y) changes to 1 and is detected by y

The new table for y is given by

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	5	1	0

$$D_y(x) = \min[c(y,x) + D_x(x), c(y,z) + D_z(x)] = \min[1 + 0, 1 + 5] = 1$$

$$D_y(z) = \min[c(y,x) + D_x(z), c(y,z) + D_z(z)] = \min[1 + 5, 1 + 0] = 1$$

Routing algorithms: Bellman-Ford Algorithm

Effect of changes in link cost

Based on the broadcast from $D_y()$, x and z update as follows

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	5	1	0

	x	y	z
$D_x(\cdot)$	0	1	2
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	5	1	0

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	2	1	0

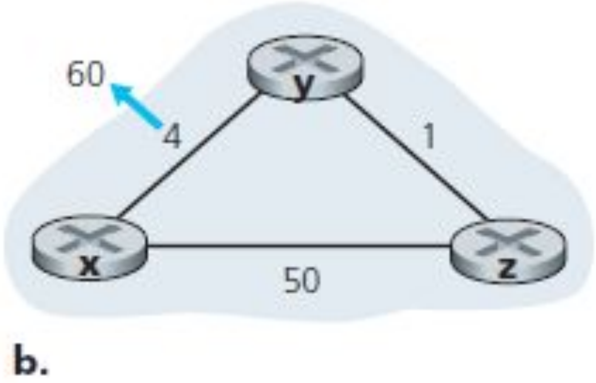
	x	y	z
$D_x(\cdot)$	0	1	2
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	2	1	0

	x	y	z
$D_x(\cdot)$	0	1	2
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	2	1	0

	x	y	z
$D_x(\cdot)$	0	1	2
$D_y(\cdot)$	1	0	1
$D_z(\cdot)$	2	1	0

Effect of changes in link cost

- Initial distance vectors will be the same for x, y and z



	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	4	0	1
$D_z(\cdot)$	5	1	0

$$c(x,y) = c(y,x) = 60$$

$$c(y,z) = c(z,y) = 1$$

$$c(z,x) = c(x,z) = 50$$

- Suppose link cost c(x,y) changes to 60 and is detected by y
- The new table for y is given by

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	5	1	0

$$D_y(x) = \min[c(y,x) + D_x(x), c(y,z) + D_z(x)] = \min[60 + 0, 1 + 5] = 6$$

$$D_y(z) = \min[c(y,x) + D_x(z), c(y,z) + D_z(z)] = \min[60 + 5, 1 + 0] = 1$$

Routing algorithms: Bellman-Ford Algorithm

- Based on the broadcast from $D_y()$, x and z update as follows
- Based on $D_x()$ and $D_z()$, x, y and z update

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	5	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	5	1	0

	x	y	z
$D_x(\cdot)$	0	4	5
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	7	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	7	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	8	0	1
$D_z(\cdot)$	7	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	6	0	1
$D_z(\cdot)$	7	1	0

COMPUTER COMMUNICATION NETWORKS

Routing algorithms: Bellman-Ford Algorithm

- Only y’s distance vector changed, so broadcast D_y to x and z
- Based on $D_x()$ and $D_z()$, x, y and z update
- As only z’s distance vector changed, z broadcasts to both x and y

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	8	0	1
$D_z(\cdot)$	7	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	8	0	1
$D_z(\cdot)$	7	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	8	0	1
$D_z(\cdot)$	9	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	8	0	1
$D_z(\cdot)$	9	1	0

	x	y	z
$D_x(\cdot)$	0	51	50
$D_y(\cdot)$	10	0	1
$D_z(\cdot)$	9	1	0

So, the nodes y and z alternatively broadcast and keep updating their distance vectors. In total, it takes 44 iterations to finally converge

Routing algorithms: Bellman-Ford Algorithm

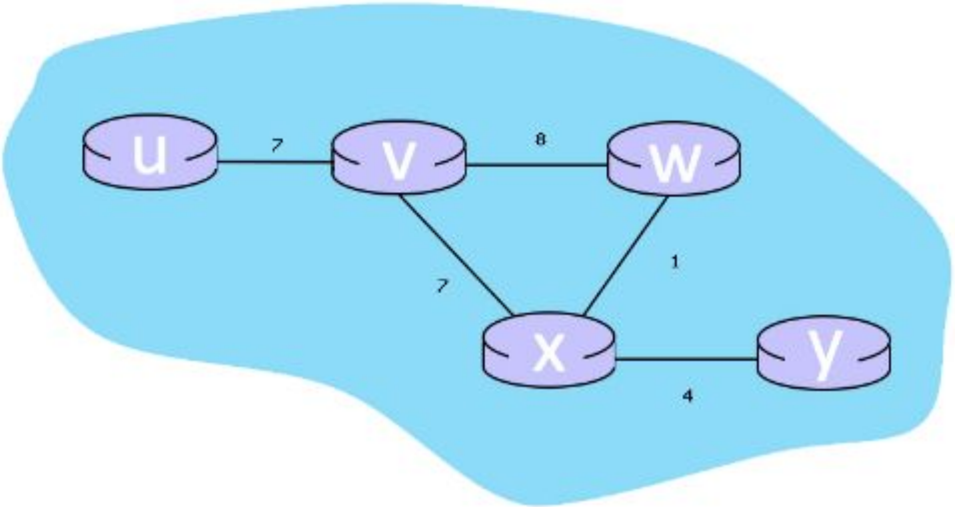
Numerical 1:

Consider the 6-node network shown below, with the given link costs:

1. When the algorithm converges, what are the distance vectors from router 'V' to all routers? Write your answer as u,v,w,x,y

2. What are the initial distance vectors for router 'U'? Write your answer as u,v,w,x,y and if a distance is ∞, write 'x'

3. The phrase 'Good news travels fast' is very applicable to distance vector routing when link costs decrease; what is the name of the problem that can occur when link costs increase?

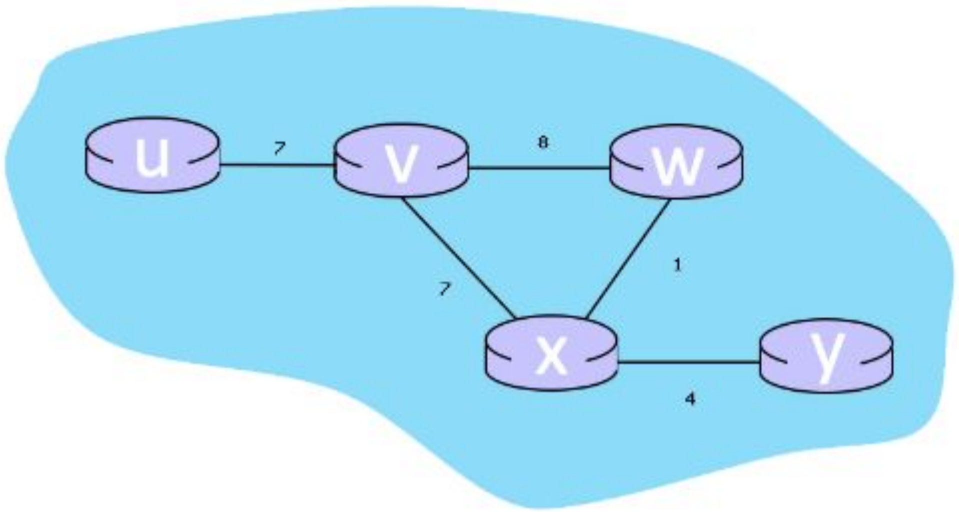


SOLUTION

1. When the algorithm converges, router V has distance vectors (u,v,w,x,y) = (7,0,8,7,11)

2. The initial distance vectors of router U are: (u,v,w,x,y) = (0,7,x,x,x) where x is ∞

3. It is called the 'Count to Infinity' problem.



THANK YOU

M Rajasekar
Department of Electronics and Communication Engineering
rajasekarmohan@pes.edu