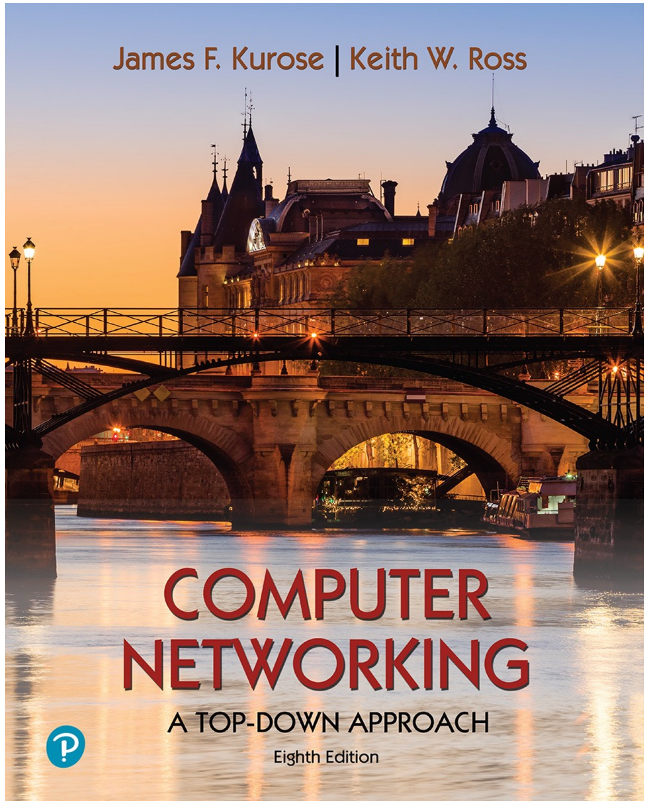
Chapter 5
Network Layer: Control Plane

Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

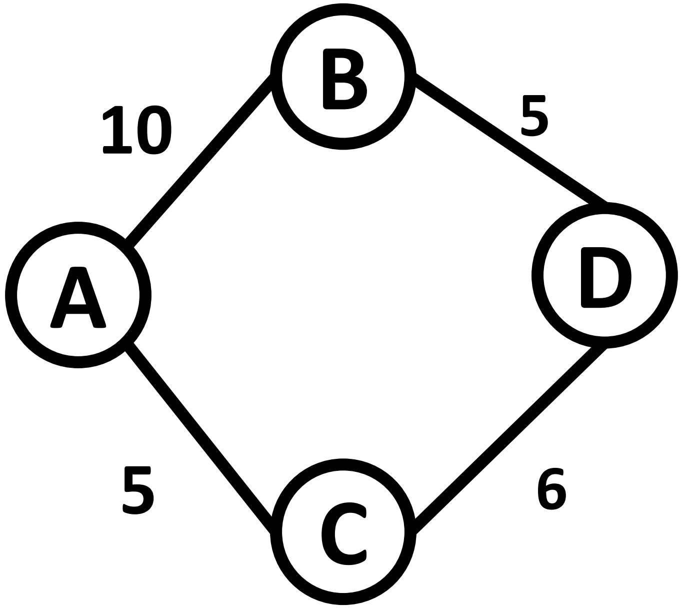
Adapted from the slides of the book's authors

Network layer: “control plane” roadmap

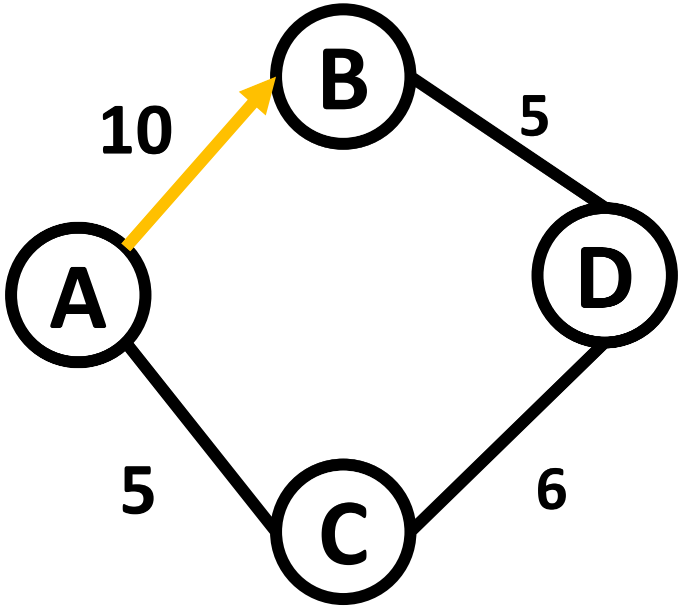
- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

- network management, configuration
  - SNMP
  - NETCONF/YANG

The Key Insight: routing by rumor



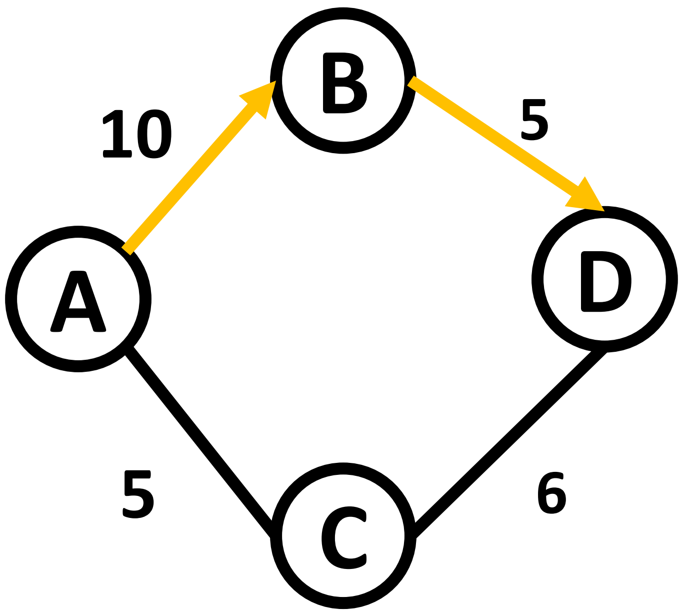
The Key Insight: routing by rumor



A ask B: can you reach D? If yes, at what cost?

B answers: yes and the cost is 5.

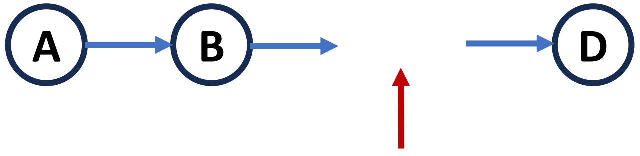
The Key Insight: routing by rumor



A ask B: can you reach D? If yes, at what cost?

B answers: yes and the cost is 5.

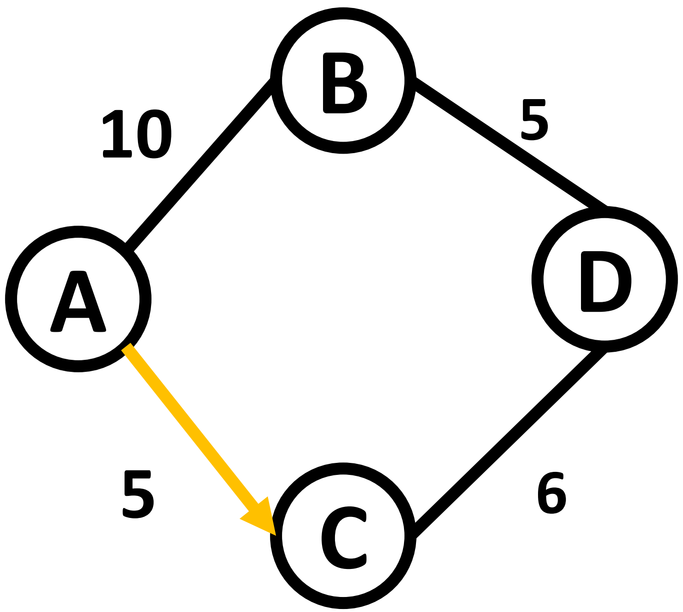
One path:



Cost: 10 + 5 =**15**

A does not know B can directly reach D or via other nodes

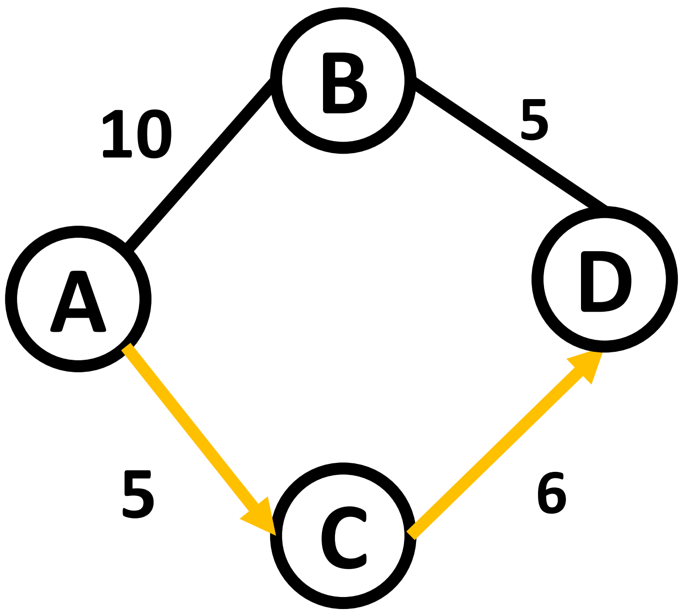
The Key Insight: routing by rumor



A ask C: can you reach D? If yes, at what cost?

C answers: yes and the cost is 6.

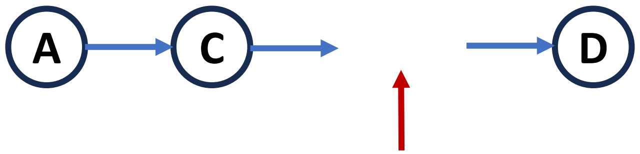
The Key Insight: routing by rumor



A ask C: can you reach D? If yes, at what cost?

C answers: yes and the cost is 6.

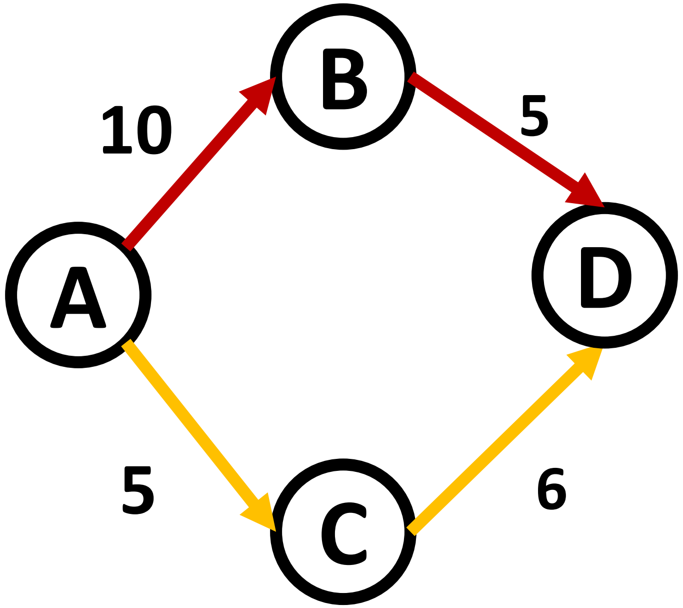
One path:



Cost: 5 + 6=**11**

A does not know C can directly reach D or via other nodes

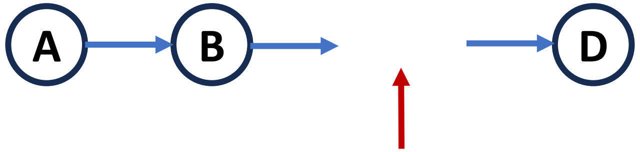
The Key Insight: routing by rumor



A ask B: can you reach D? If yes, at what cost?

B answers: yes and the cost is 5.

One path:



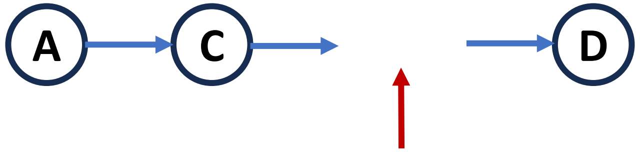
Cost: 10 + 5 =**15**

A does not know B can directly reach D or via other nodes

A ask C: can you reach D? If yes, at what cost?

C answers: yes and the cost is 6.

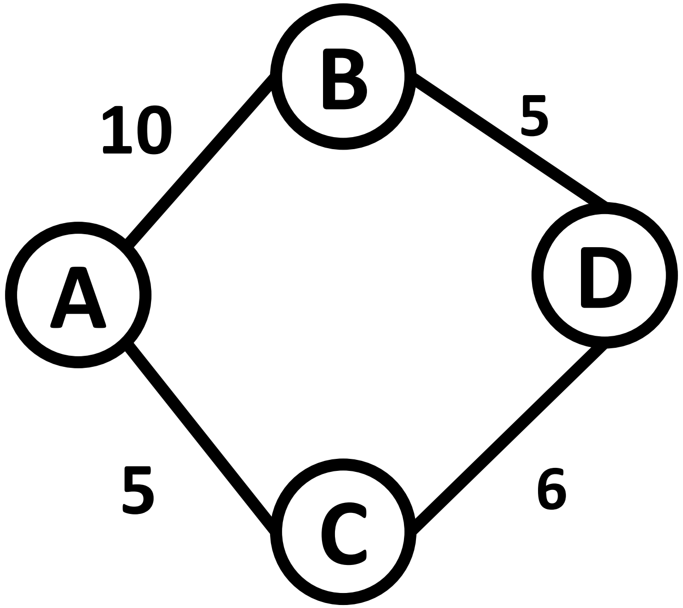
One path:

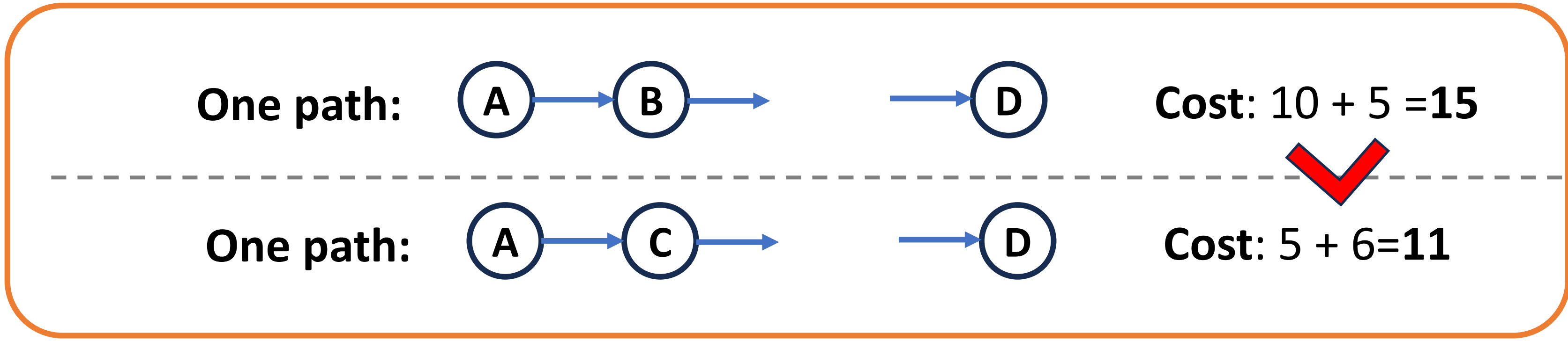


Cost: 5 + 6=**11**

A does not know C can directly reach D or via other nodes

The Key Insight: routing by rumor

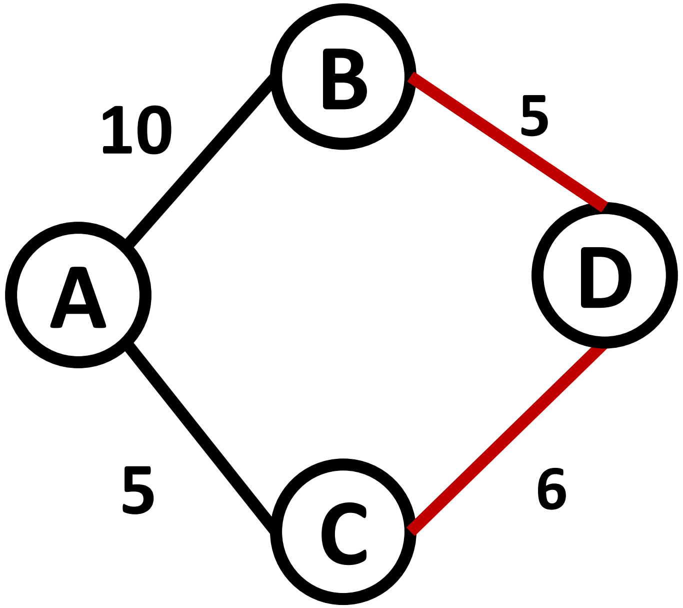


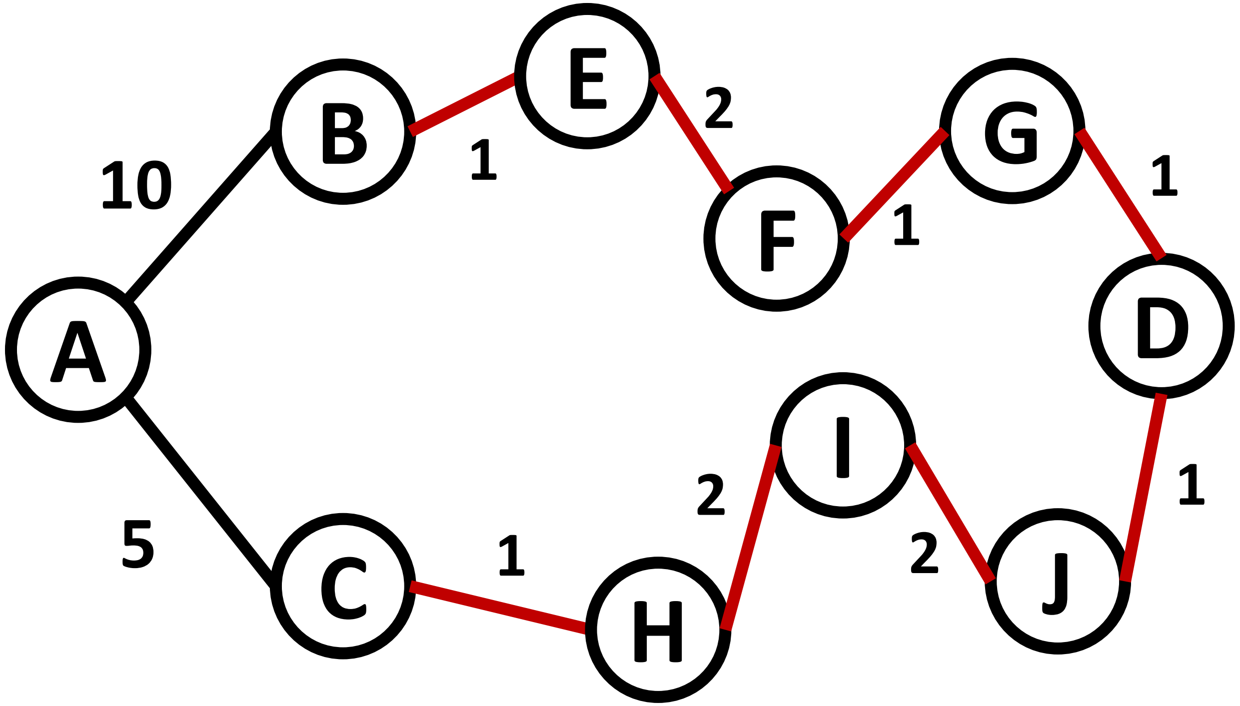


<u>**Conclusion:**</u> For destination D, node A selects C as the optimal next hop

The Key Insight: routing by rumor

B answers: yes and the cost is 5.

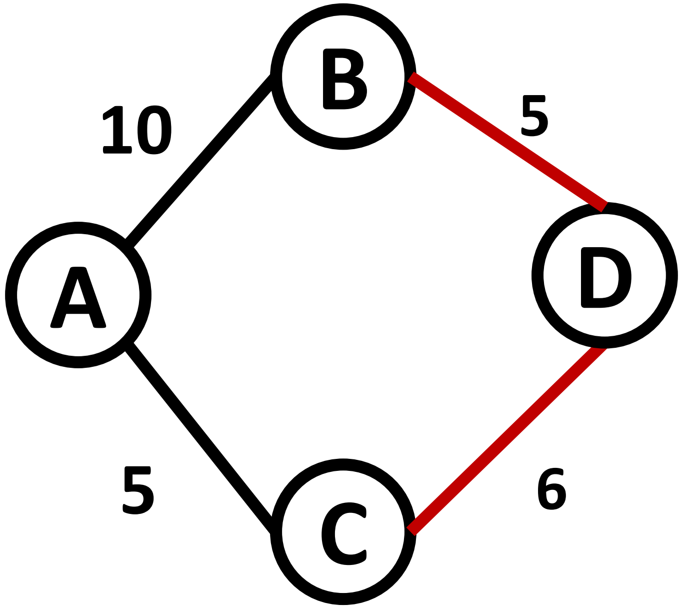




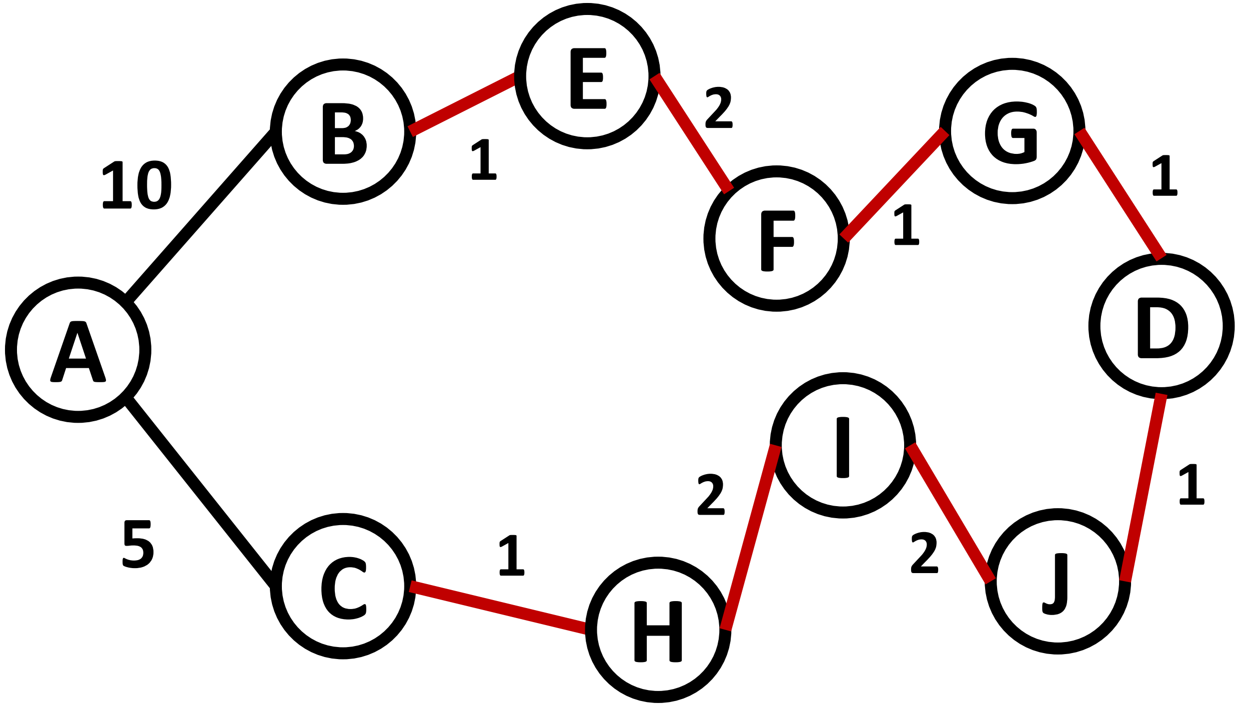
C answers: yes and the cost is 6.

Conclusion: For destination D, node A selects C as the optimal next hop

The Key Insight: routing by rumor



B answers: yes and the cost is 5.

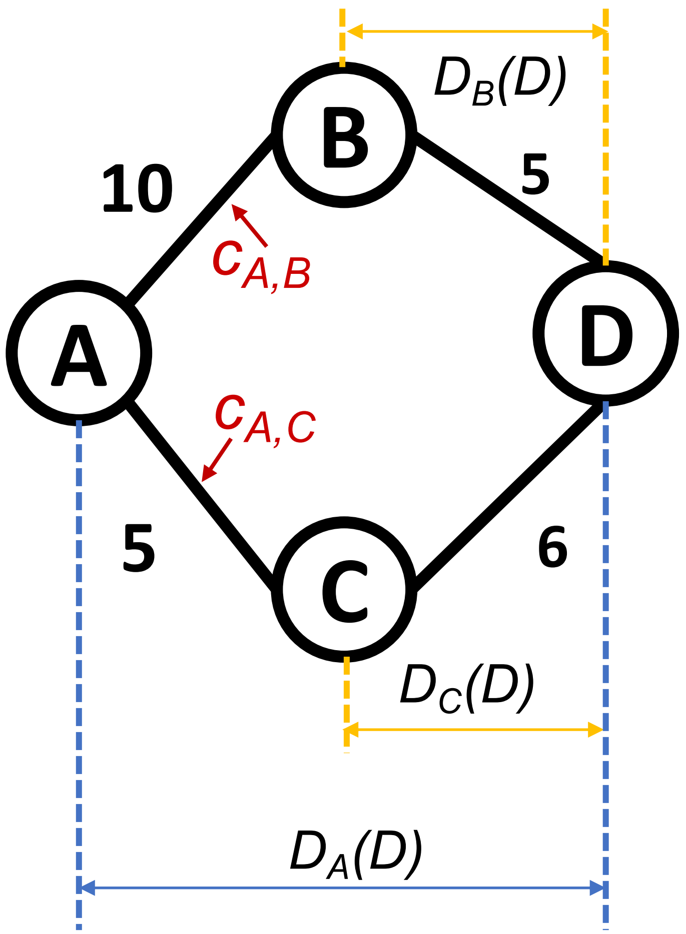


C answers: yes and the cost is 6.

B -> D, direct connection or indirect connection doesn’t mater, as long as the cost to D is correct.

But, we need to be able to trust the neighbors, B and C!

Routing by rumor: Math behind it

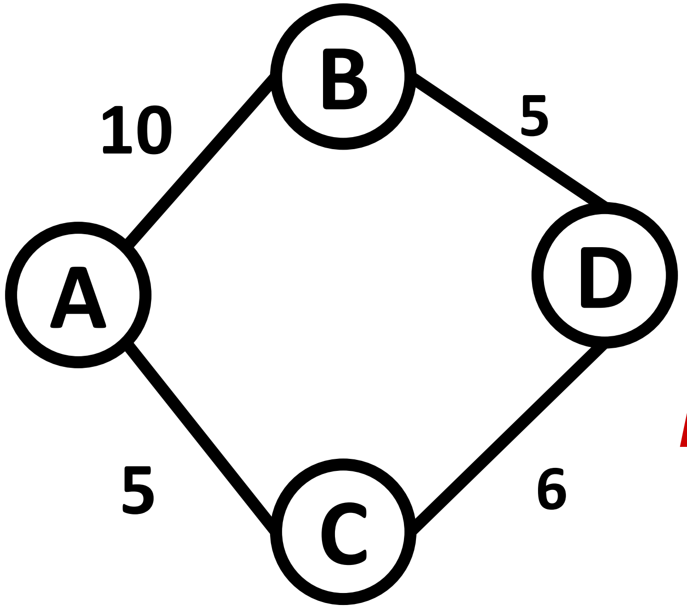


Let $D_x(y)$: cost of least-cost path from x to y

$c_{x,v}$ Cost of **direct link** from x to its neighbor v

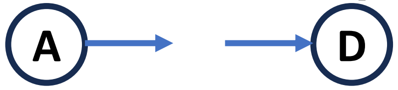
$$D_A(D) = \min_v \{c_{A,B} + D_B(D),\ c_{A,C} + D_C(D)\}$$

Routing by rumor: Math behind it



Let $D_x(y)$: cost of least-cost path from x to y

Example:



$c_{x,v}$ Cost of direct link from x to its neighbor v

Example:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

v's estimated least-cost-path cost to y

min taken over all neighbors v of x

direct cost of link from x to v

Routing by rumor: Math behind it

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.
Then:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

v's estimated least-cost-path cost to y

min taken over all neighbors v of x

direct cost of link from x to v

Distance vector algorithm

Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let $D_x(y)$: cost of least-cost path from x to y.

Then:

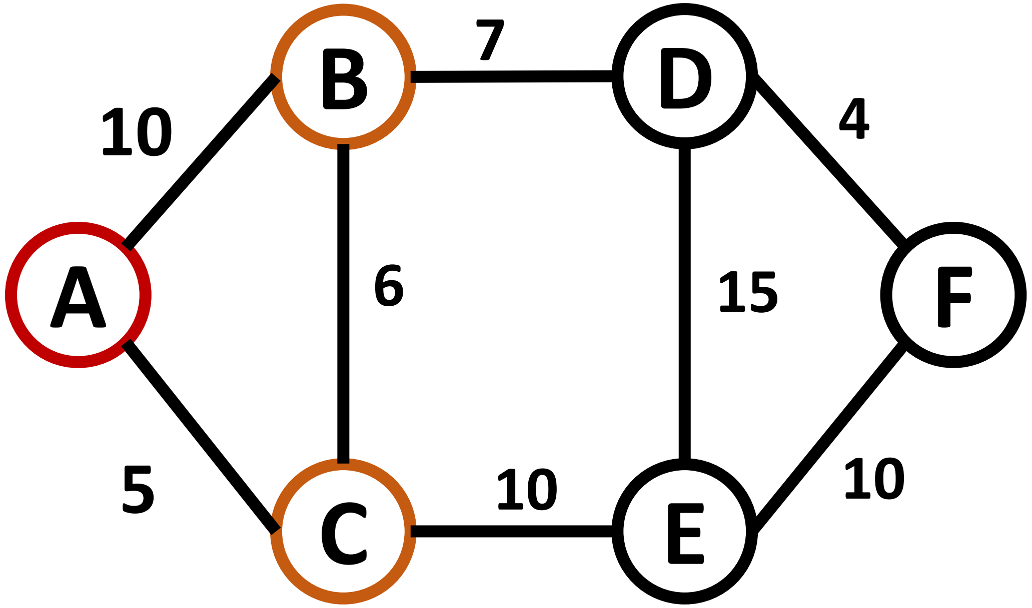
$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

v's estimated least-cost-path cost to y

min taken over all neighbors v of x

direct cost of link from x to v

Bellman-Ford Example



Q: Find the shortest distance A -> F

A has two neighbors: B and C

Shortest distance:

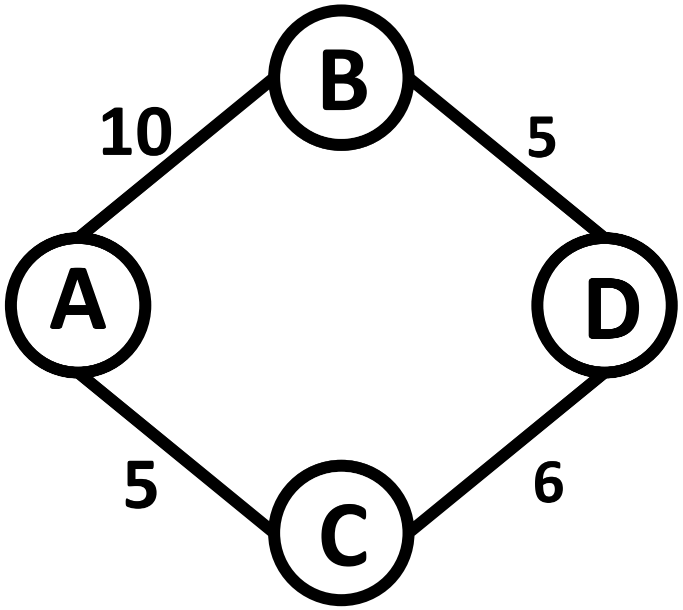
B->F : 11 C->F : 20

A->B : 10 A->C : 5

A->B->F : 21 < A->C->F : 25

Shortest Path: A->B->F Next Hop: B Total Cost: 21

Detailed Steps of Distance Vector Algorithm

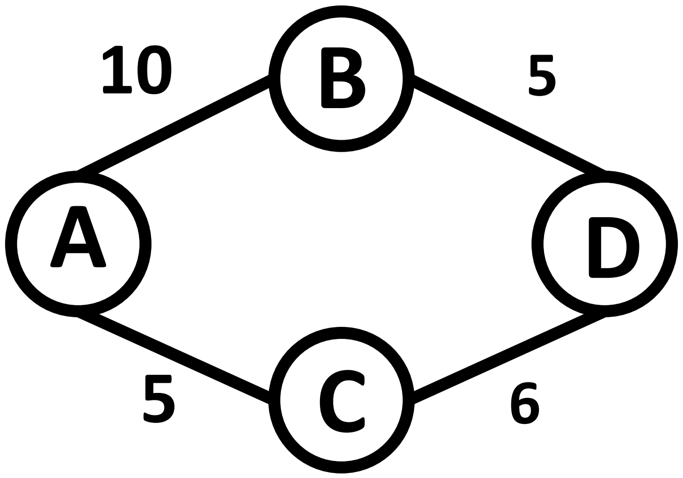


The first step, nodes send hello messages to discover neighbors, and determine the link cost

Detailed Steps of Distance Vector Algorithm

Generating the Distance Vectors

Dest	Cost	Next Hop
A	10	A
C	∞	
D	5	D



Distance Vector

Dest	Cost	Next Hop
B	10	B
C	5	C
D	∞	

Neighbors

Unknown Destination

Dest	Cost	Next Hop
A	∞	
B	5	B
C	6	C

Dest	Cost	Next Hop
A	5	A
B	∞	
D	6	D

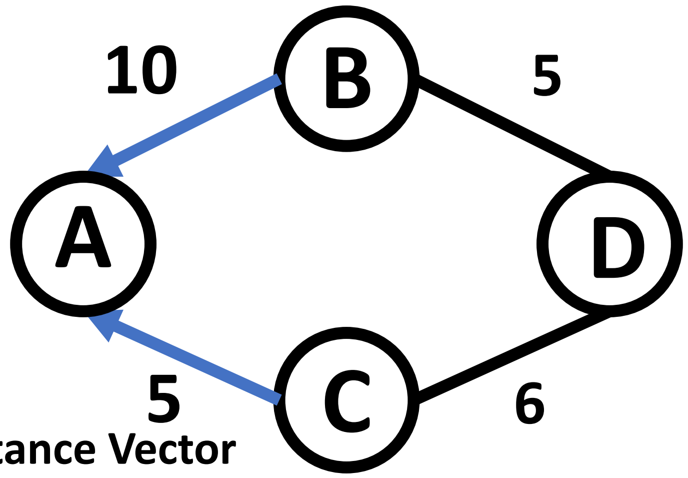
Detailed Steps of Distance Vector Algorithm

Exchanging the Distance Vectors with Neighbors

Distance Vector

Neighbors

Dest	Cost	Next Hop
B	10	B
C	5	C
D	∞	



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	∞

Distance Vector of Node B

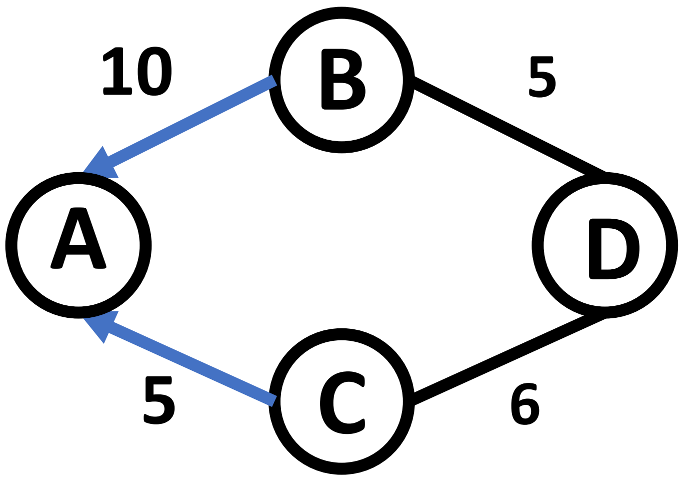
Dest	Cost
A	10
C	∞
D	5

Distance Vector of Node C

Dest	Cost
A	5
B	∞
D	6

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node B

Dest	Cost
A	10
C	∞
D	5

Distance Vector of Node A

Dest	Cost
B	10
C	5
D	∞

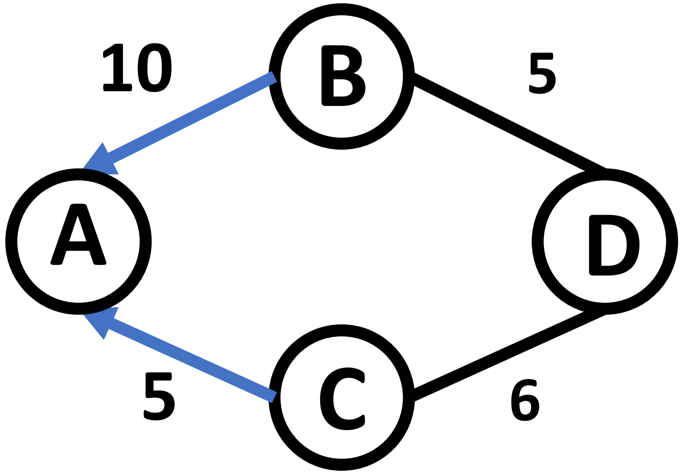
Distance Vector of Node C

Dest	Cost
A	5
B	∞
D	6

C can not reach B

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	∞

Distance Vector of Node B

Dest	Cost
A	10
C	∞
D	5

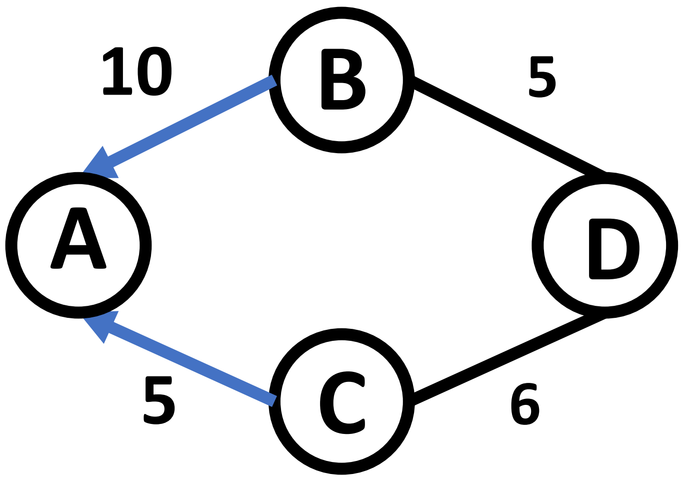
B can not reach C

Distance Vector of Node C

Dest	Cost
A	5
B	∞
D	6

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	∞

Distance Vector of Node B

Dest	Cost
A	10
C	∞
D	5

A->B->D Cost: 15

B can reach D

Distance Vector of Node C

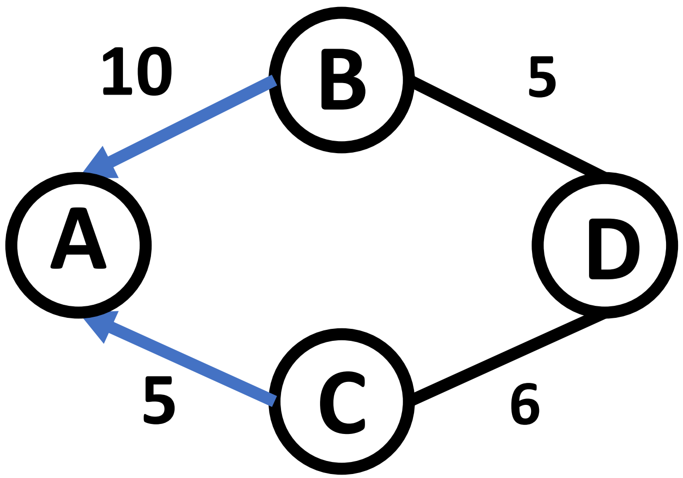
Dest	Cost
A	5
B	∞
D	6

A->C->D Cost: 11

C can reach D

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node A

Dest	Cost	Next Hop
B	10	B
C	5	C
D	11	C

Distance Vector of Node B

Dest	Cost
A	10
C	∞
D	5

A->B->D Cost: 15

Distance Vector of Node C

Dest	Cost
A	5
B	∞
D	6

A->C->D Cost: 11

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors

Dest	Cost	Next Hop
A	10	A
C	11	D
D	5	D

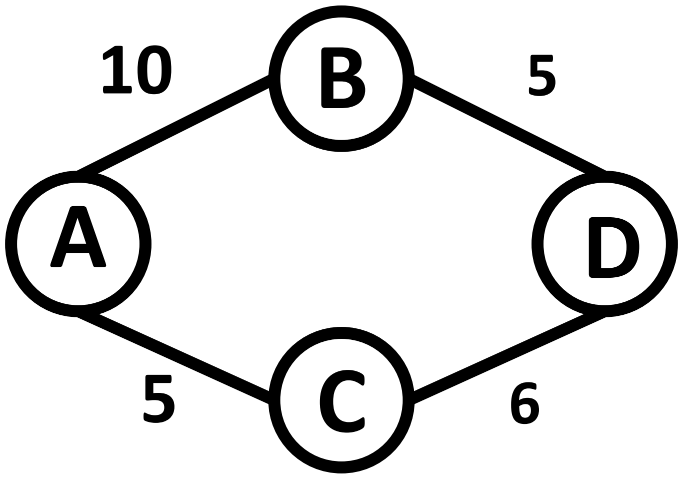
DV has updated!

Distance Vector

Dest	Cost	Next Hop
B	10	B
C	5	C
D	11	C

Dest	Cost	Next Hop
B	10	B
C	5	C
D	∞	

DV has updated!



Dest	Cost	Next Hop
A	11	C
B	5	B
C	6	C

DV has updated!

Dest	Cost	Next Hop
A	5	A
B	11	D
D	6	D

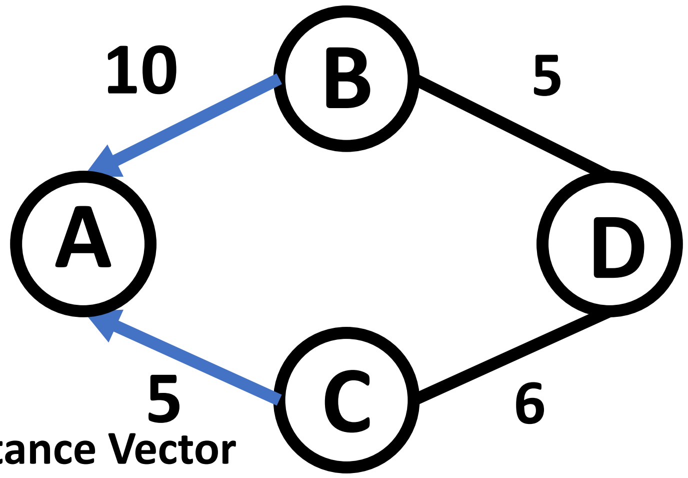
DV has updated!

Detailed Steps of Distance Vector Algorithm

Exchanging the Distance Vectors with Neighbors

Distance Vector

Dest	Cost	Next Hop
B	10	B
C	5	C
D	11	C



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	11

Distance Vector of Node B

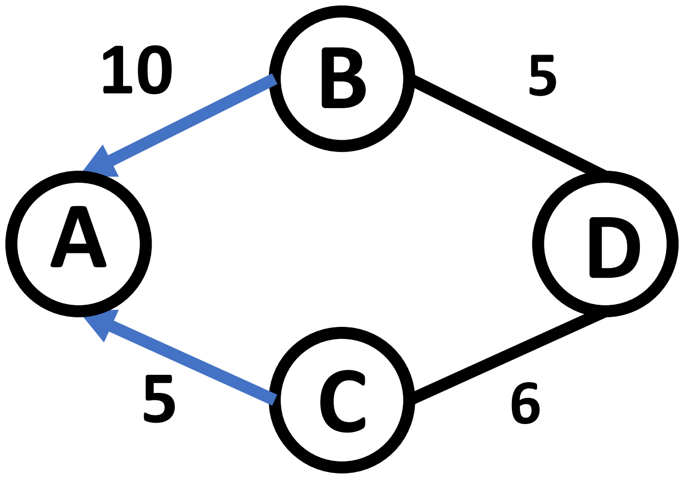
Dest	Cost
A	10
C	11
D	5

Distance Vector of Node C

Dest	Cost
A	5
B	11
D	6

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node B

Dest	Cost
A	10
C	11
D	5

Distance Vector of Node A

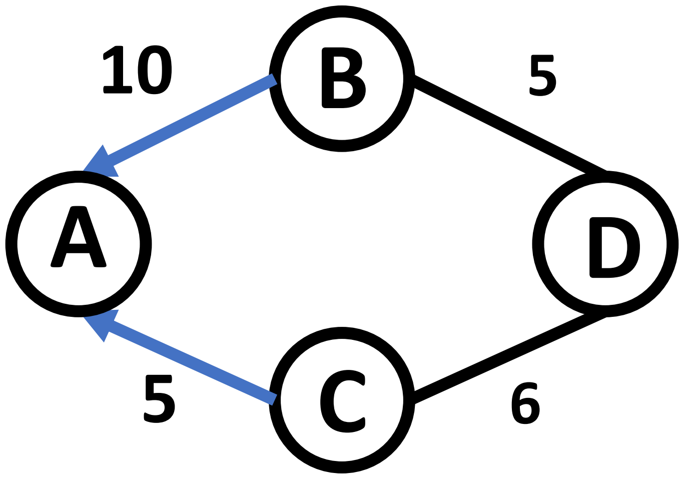
Dest	Cost
B	10
C	5
D	11

Distance Vector of Node C

Dest	Cost
A	5
B	11
D	6

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	11

Distance Vector of Node B

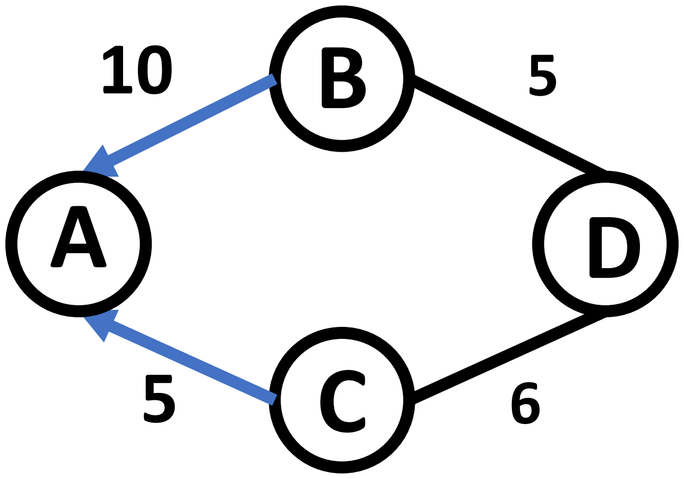
Dest	Cost
A	10
C	11
D	5

Distance Vector of Node C

Dest	Cost
A	5
B	11
D	6

Detailed Steps of Distance Vector Algorithm

Recomputing the Distance Vectors



Distance Vector of Node A

Dest	Cost
B	10
C	5
D	11

Distance Vector of Node B

Dest	Cost
A	10
C	11
D	5

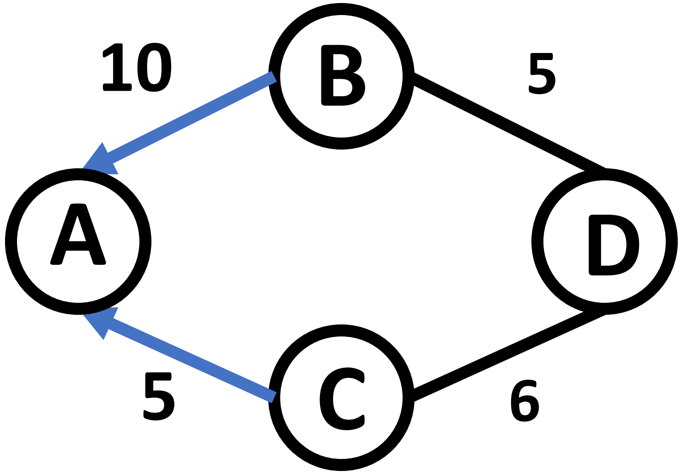
Distance Vector of Node C

Dest	Cost
A	5
B	11
D	6

The algorithm has converaged!

A larger network takes longer to converage

Distance vector algorithm:



each node:

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

recompute DV estimates using DV received from neighbor

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

Distance vector algorithm:

each node:

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

recompute DV estimates using DV received from neighbor

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

iterative, asynchronous: each local iteration caused by:

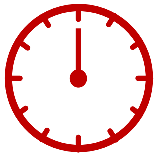
- local link cost change
- DV update message from neighbor

distributed, self-stopping: each node notifies neighbors *only* when its DV changes

- neighbors then notify their neighbors – *only if necessary*
- no notification received, no actions taken!

A larger network takes longer to converage

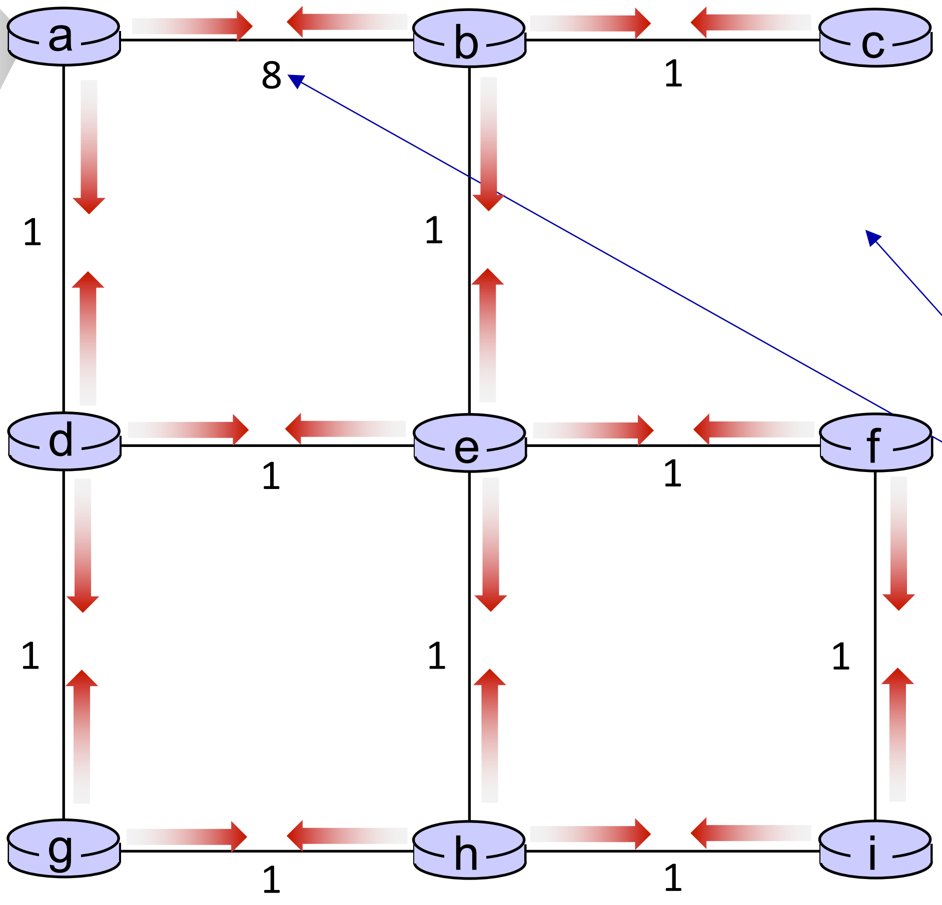
Distance vector: example



t=0

- All nodes have distance estimates to nearest neighbors (only)
- All nodes send their local distance vector to their neighbors

DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

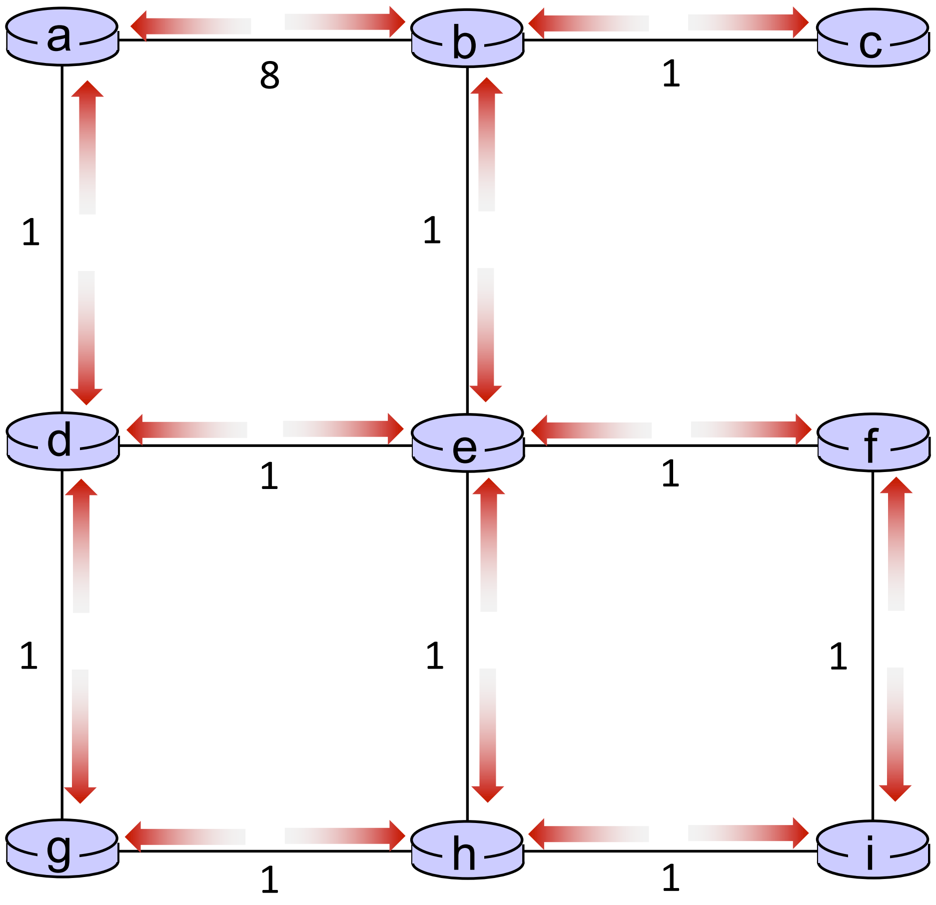


A few asymmetries:
- missing link
- larger cost

Distance vector example: iteration

t=1

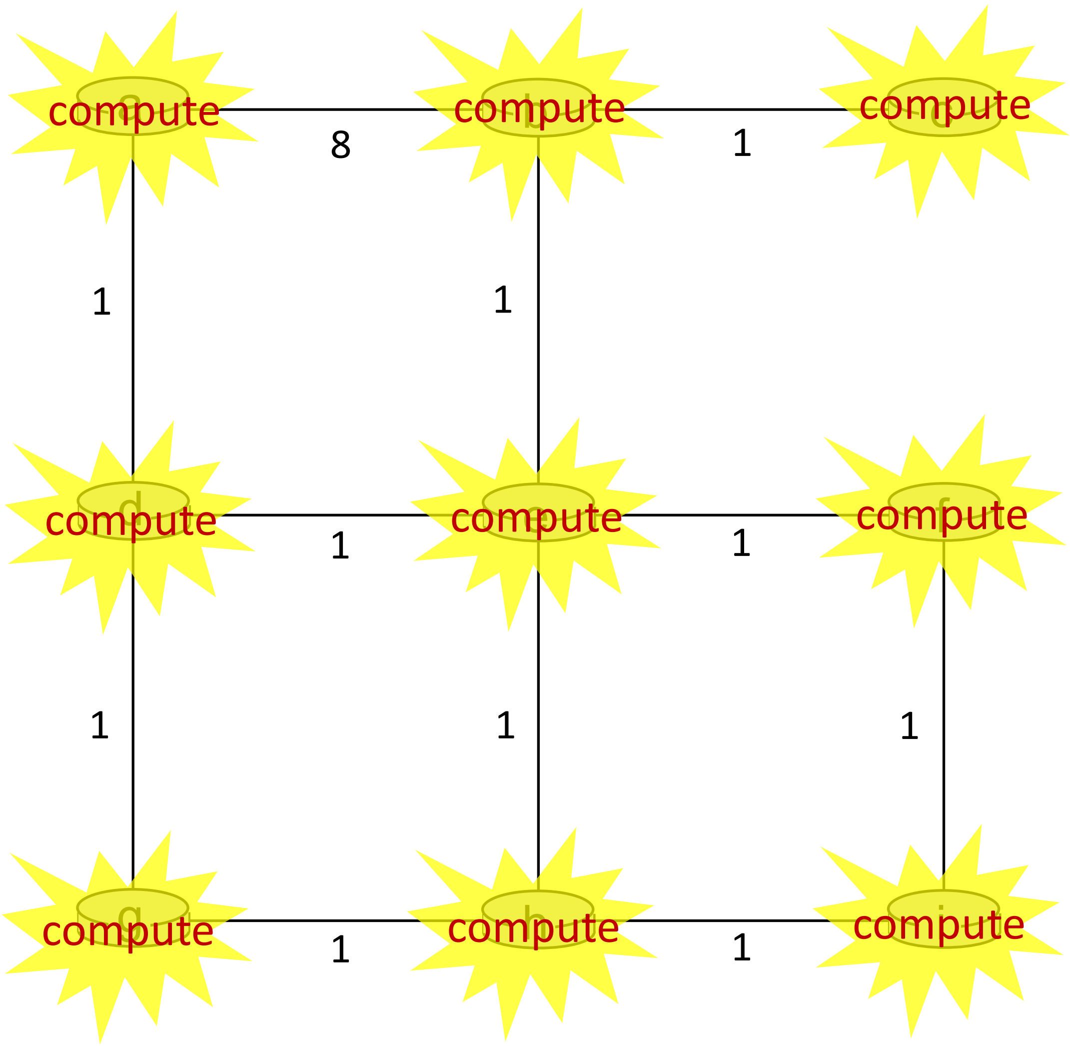
All nodes:
- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



Distance vector example: iteration

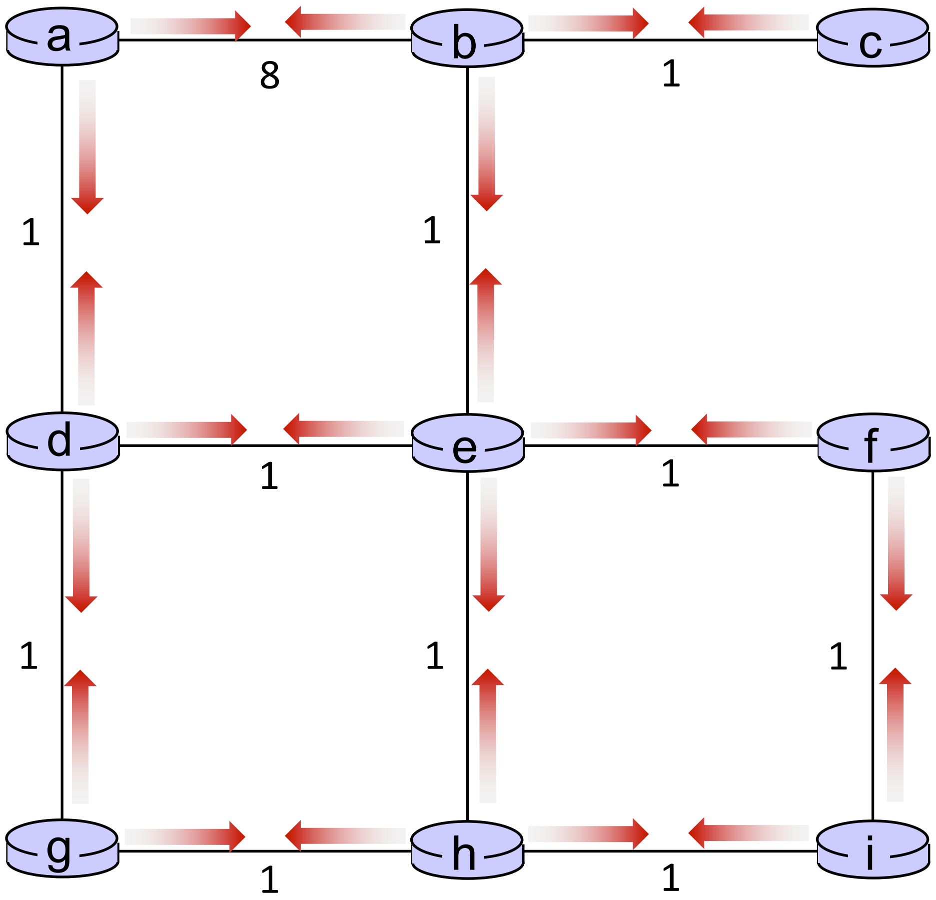
t=1

All nodes:
- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

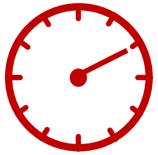


Distance vector example: iteration

All nodes:
- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



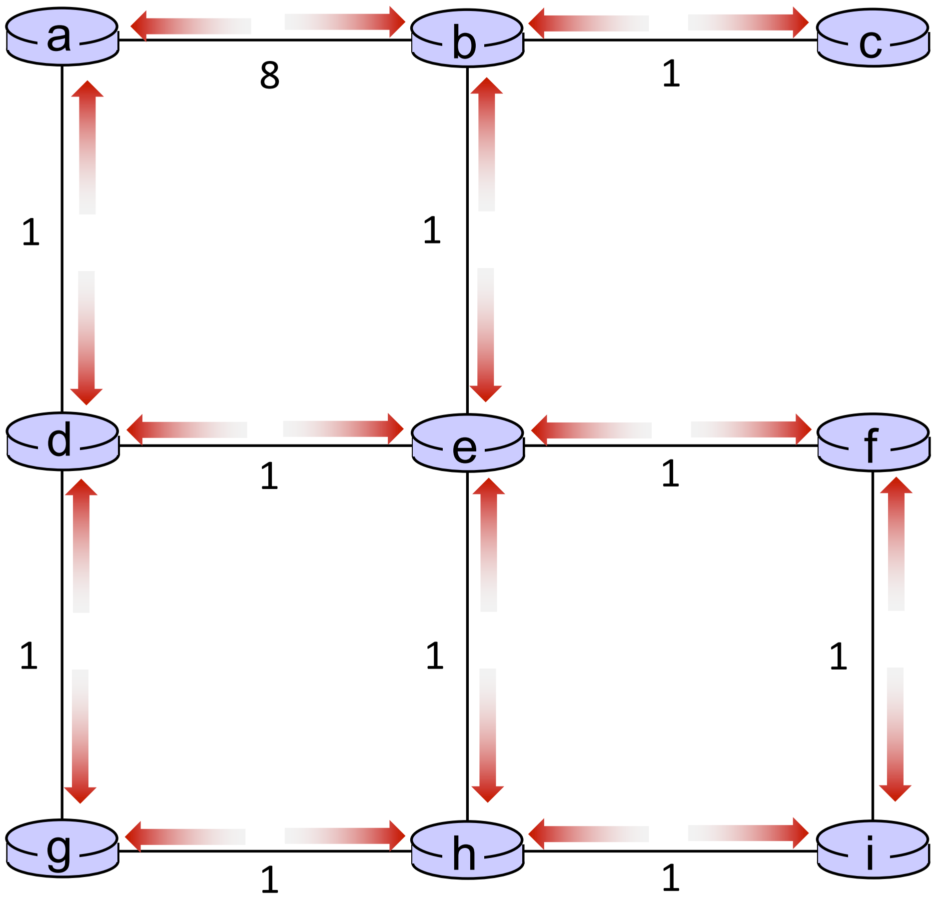
Distance vector example: iteration



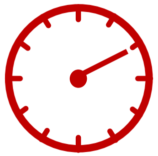
t=2

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



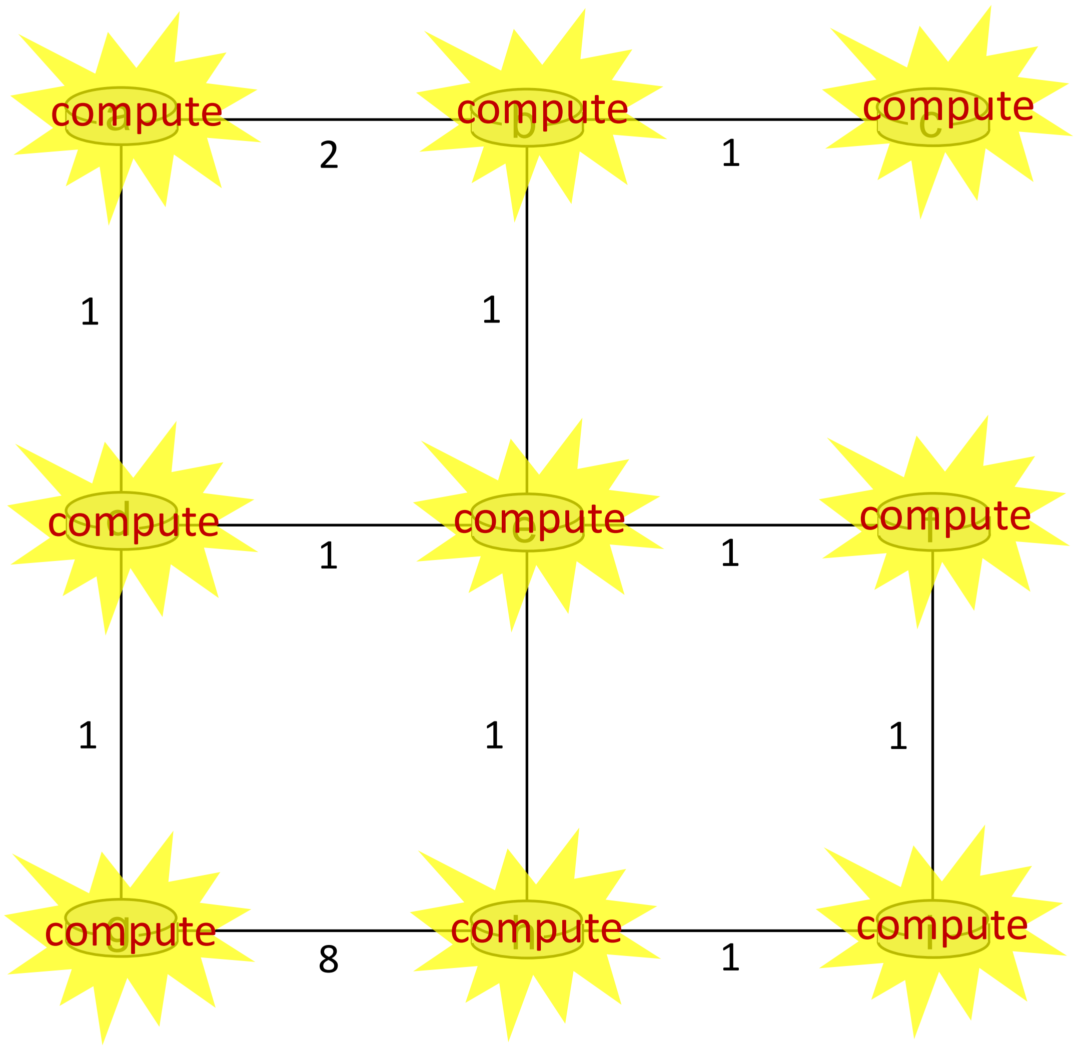
Distance vector example: iteration



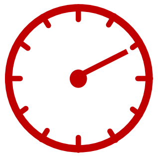
t=2

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors

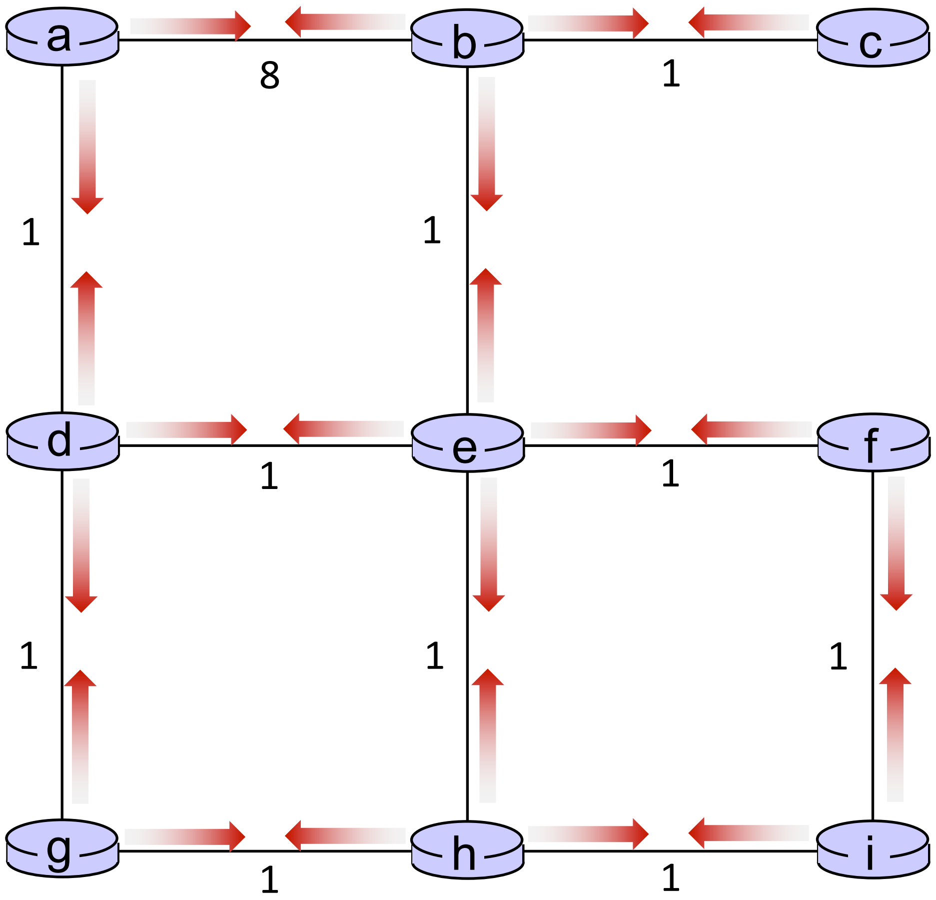


Distance vector example: iteration



t=2

All nodes:
- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



Distance vector example: iteration

…. and so on

Let’s next take a look at the iterative *computations* at nodes

Distance vector example:

t=1

- b receives DVs from a, c, e

DV in a:

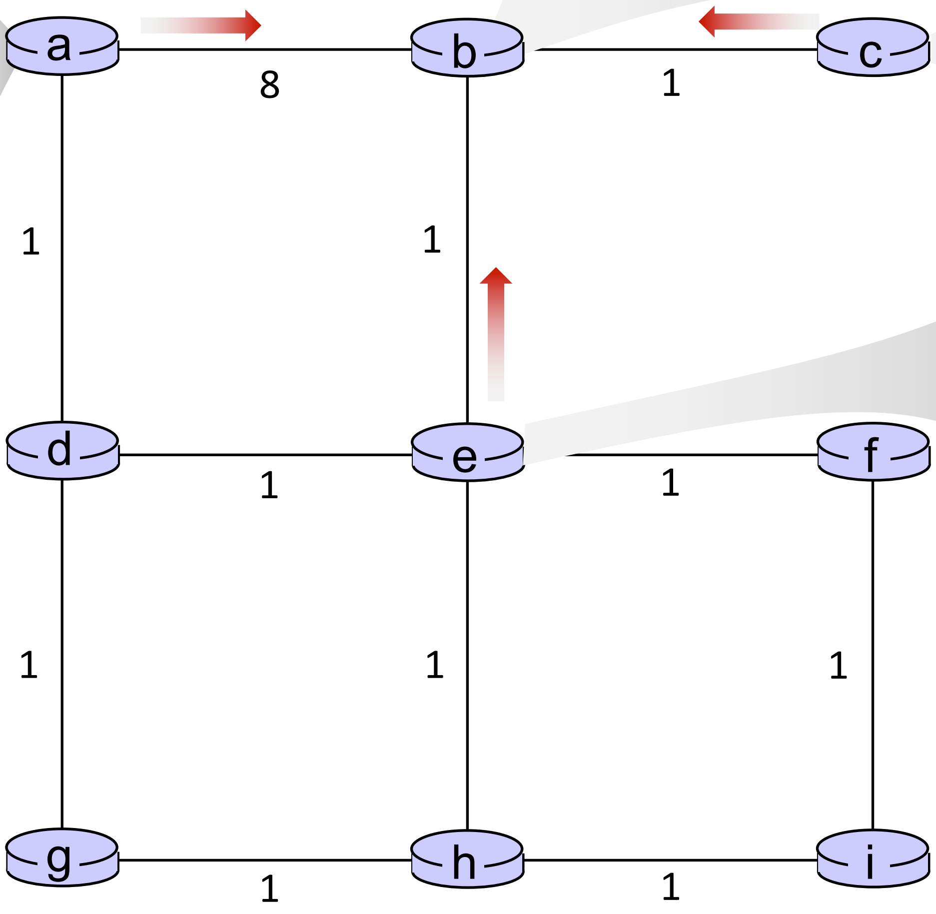
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

DV in b:

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



DV in e:

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

Distance vector example: (

t=1

- b receives DVs from a, c, e, computes:

DV in a:

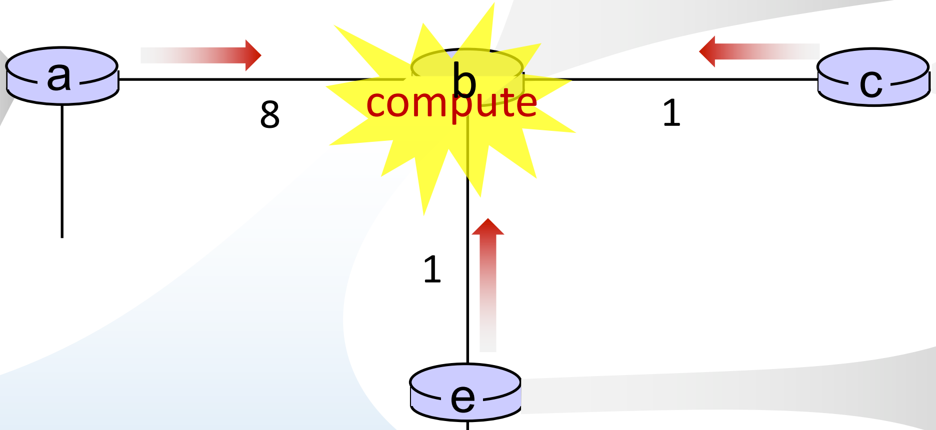
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

DV in b:

$D_b(a) = 8$ $D_b(f) = \infty$
$D_b(c) = 1$ $D_b(g) = \infty$
$D_b(d) = \infty$ $D_b(h) = \infty$
$D_b(e) = 1$ $D_b(i) = \infty$

DV in c:

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



DV in e:

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

$D_b(a) = \min\{c_{b,a}+D_a(a),\ c_{b,c}+D_c(a),\ c_{b,e}+D_e(a)\} = \min\{8,\infty,\infty\} = 8$

$D_b(c) = \min\{c_{b,a}+D_a(c),\ c_{b,c}+D_c(c),\ c_{b,e}+D_e(c)\} = \min\{\infty,1,\infty\} = 1$

$D_b(d) = \min\{c_{b,a}+D_a(d),\ c_{b,c}+D_c(d),\ c_{b,e}+D_e(d)\} = \min\{9,\infty,2\} = 2$

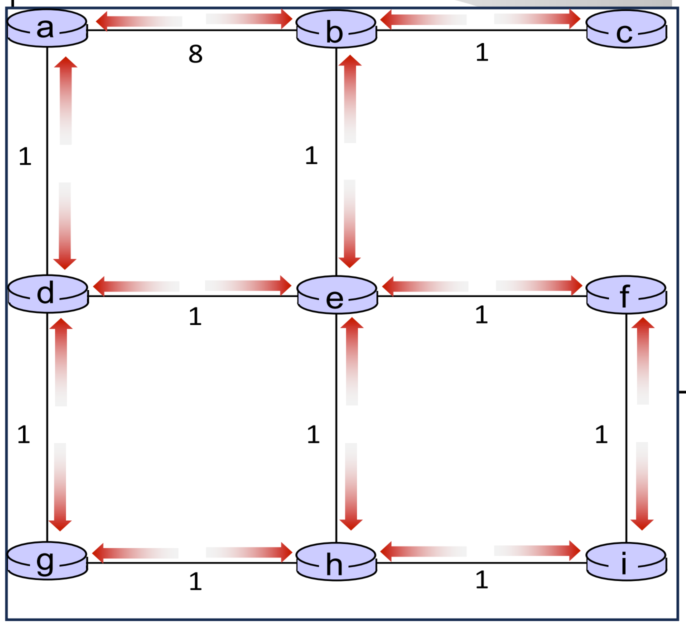
$D_b(e) = \min\{c_{b,a}+D_a(e),\ c_{b,c}+D_c(e),\ c_{b,e}+D_e(e)\} = \min\{\infty,\infty,1\} = 1$

$D_b(f) = \min\{c_{b,a}+D_a(f),\ c_{b,c}+D_c(f),\ c_{b,e}+D_e(f)\} = \min\{\infty,\infty,2\} = 2$

$D_b(g) = \min\{c_{b,a}+D_a(g),\ c_{b,c}+D_c(g),\ c_{b,e}+D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty$

$D_b(h) = \min\{c_{b,a}+D_a(h),\ c_{b,c}+D_c(h),\ c_{b,e}+D_e(h)\} = \min\{\infty, \infty, 2\} = 2$

$D_b(i) = \min\{c_{b,a}+D_a(i),\ c_{b,c}+D_c(i),\ c_{b,e}+D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty$



Distance vector example: (

t=1

- c receives DVs from b

DV in a:

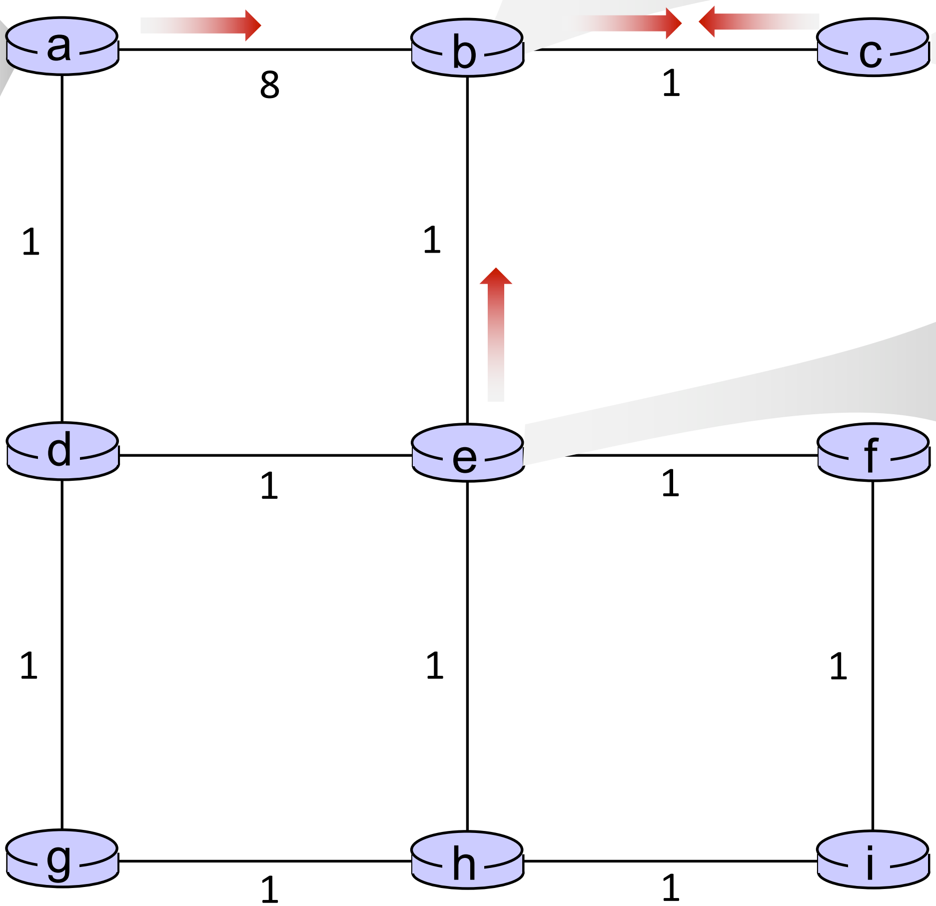
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

DV in b:

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



DV in e:

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

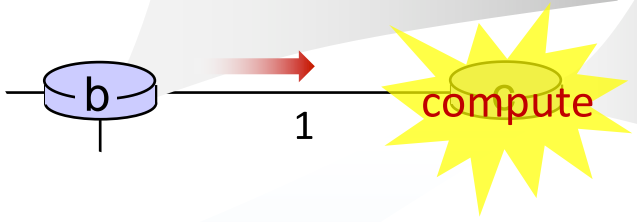
Distance vector example:

DV in b:

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:

$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$



t=1

- c receives DVs from b computes:

$D_c(a) = \min\{c_{c,b}+D_b(a)\} = 1 + 8 = 9$

$D_c(b) = \min\{c_{c,b}+D_b(b)\} = 1 + 0 = 1$

$D_c(d) = \min\{c_{c,b}+D_b(d)\} = 1+ \infty = \infty$

$D_c(e) = \min\{c_{c,b}+D_b(e)\} = 1 + 1 = 2$

$D_c(f) = \min\{c_{c,b}+D_b(f)\} = 1+ \infty = \infty$

$D_c(g) = \min\{c_{c,b}+D_b(g)\} = 1+ \infty = \infty$

$D_c(h) = \min\{c_{bc,b}+D_b(h)\} = 1+ \infty = \infty$

$D_c(i) = \min\{c_{c,b}+D_b(i)\} = 1+ \infty = \infty$

DV in c:

$D_c(a) = 9$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = 2$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Distance vector example:

t=1

- e receives DVs from b, d, f, h

DV in d:

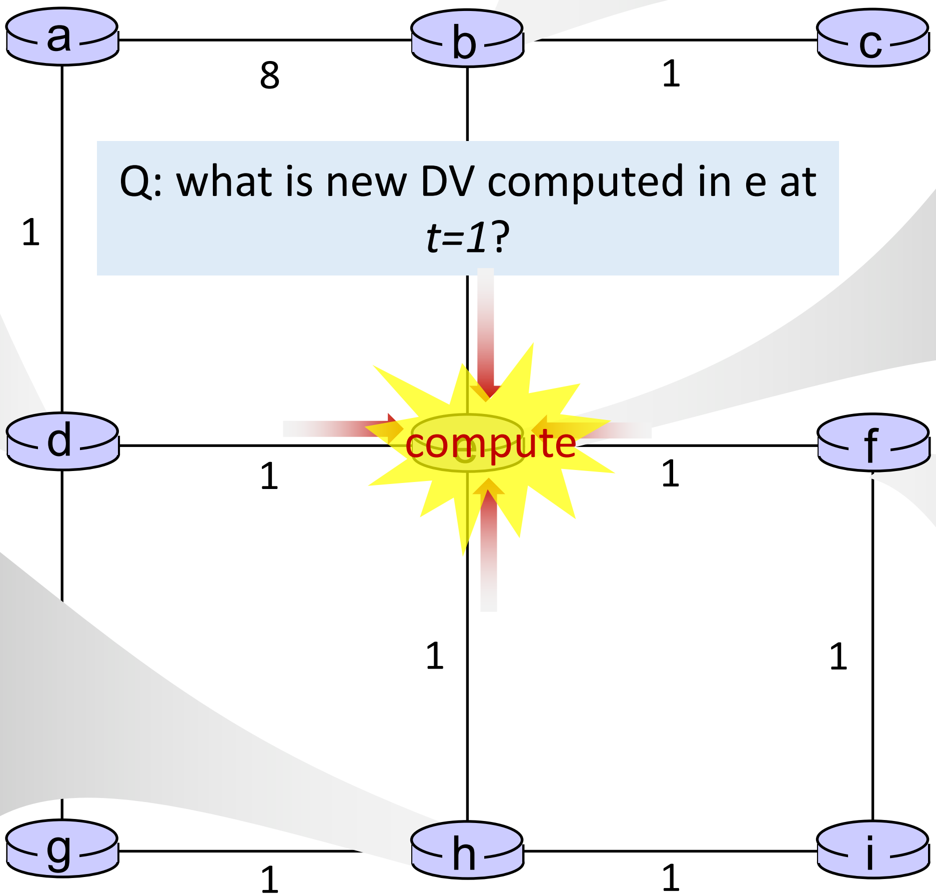
$D_c(a) = 1$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = 0$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = \infty$
$D_c(i) = \infty$

DV in b:

$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in e:

$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$



Q: what is new DV computed in e at *t=1*?

DV in h:

$D_c(a) = \infty$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = \infty$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = 0$
$D_c(i) = 1$

DV in f:

$D_c(a) = \infty$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = \infty$
$D_c(e) = 1$
$D_c(f) = 0$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = 1$

Distance vector: state information diffusion

Iterative communication, computation steps diffuses information through network:

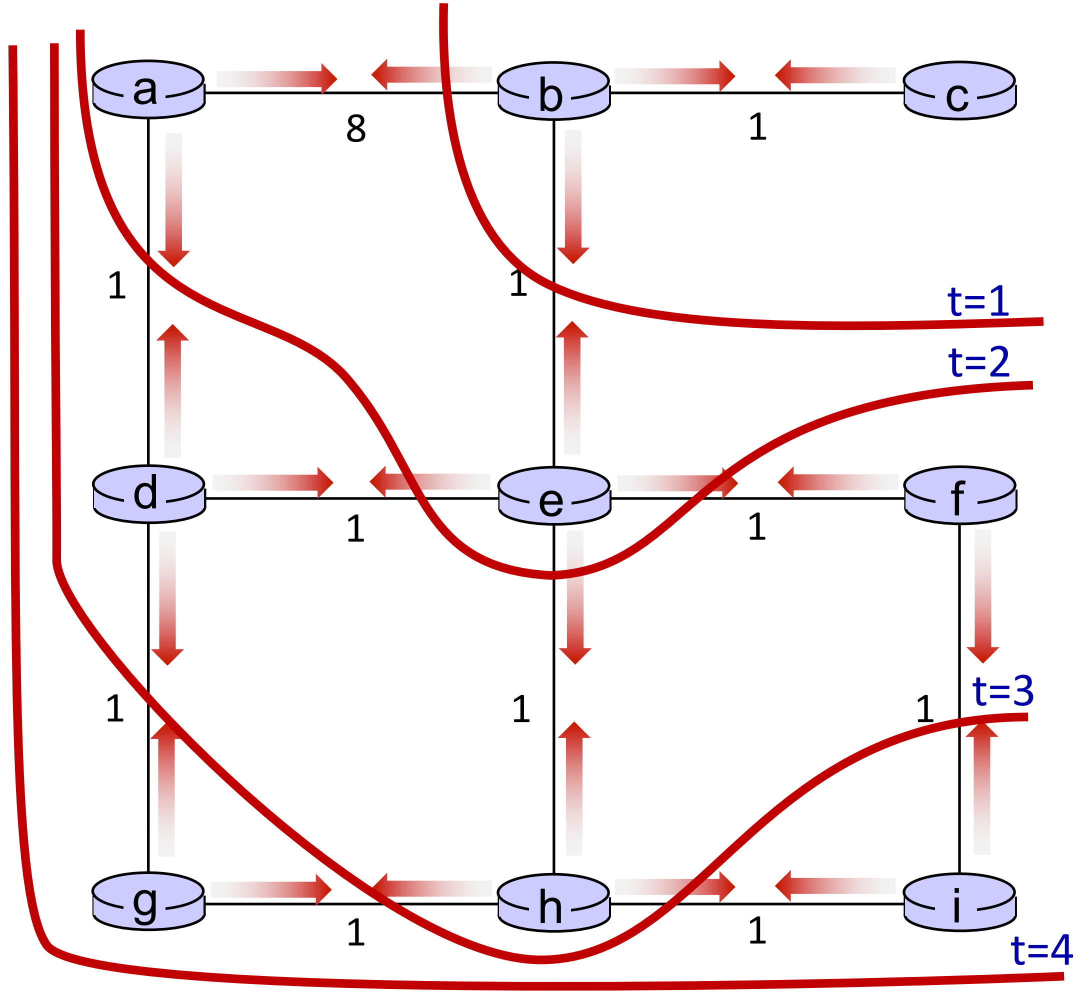
t=0 c's state at t=0 is at c only

t=1 c's state at t=0 has propagated to b, and may influence distance vector computations up to **1** hop away, i.e., at b

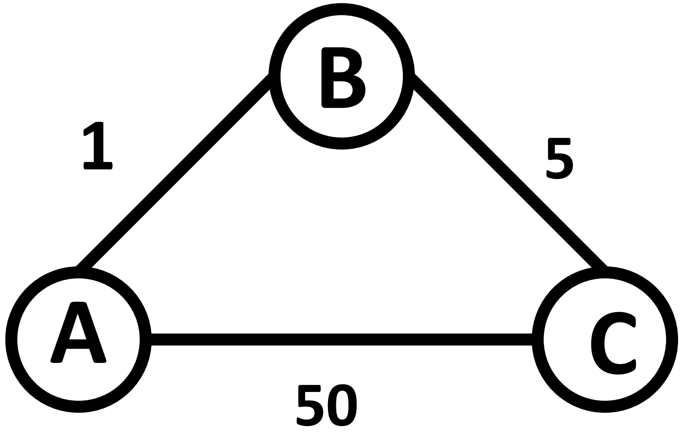
t=2 c's state at t=0 may now influence distance vector computations up to **2** hops away, i.e., at b and now at a, e as well

t=3 c's state at t=0 may influence distance vector computations up to **3** hops away, i.e., at d, f, h

t=4 c's state at t=0 may influence distance vector computations up to **4** hops away, i.e., at g, i



Distance vector: Link Cost Change



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Distance Vector of Node B

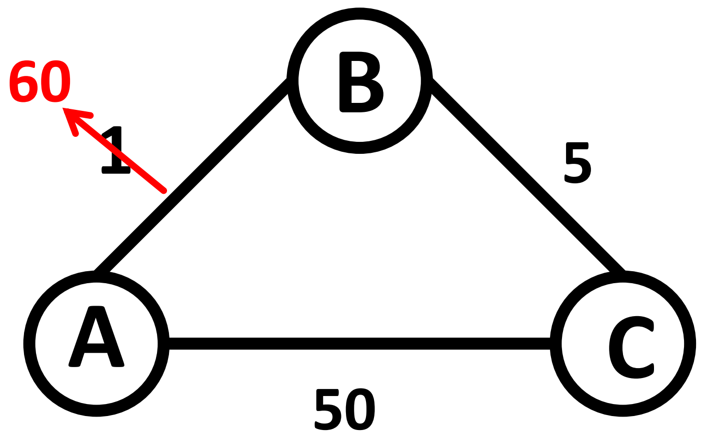
Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Distance vector: Link Cost Change

How would Node B updates its Distance Vectors?



Distance Vector of Node B

Dest	Cost	Next Hop
A	1	A
C	5	C

Node B has its own distance vector and the DV from two other nodes

Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Unknown to B

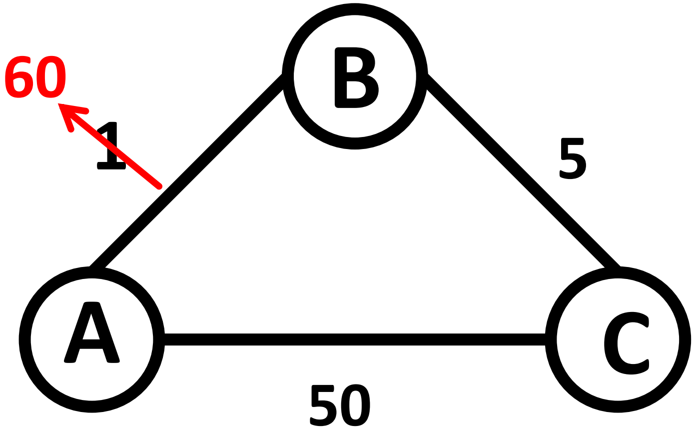
Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Unknown to B

Distance vector: Link Cost Change

How would Node B updates its Distance Vectors?



Distance Vector of Node B

Dest	Cost	Next Hop
A	~~1~~ 60	A
C	5	C

Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Unknown to B

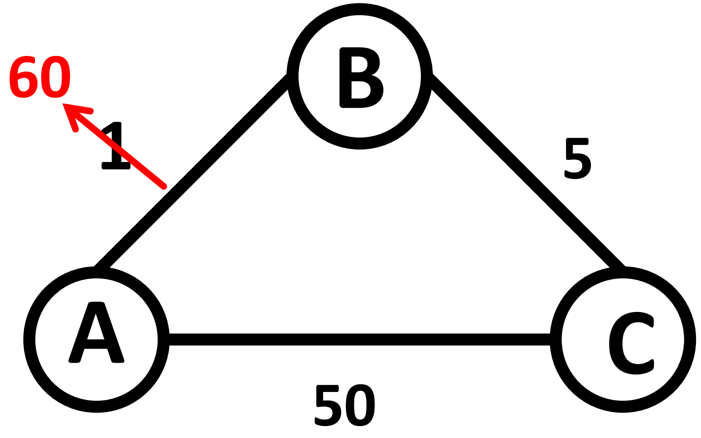
Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Unknown to B

Distance vector: Link Cost Change

How would Node B updates its Distance Vectors?



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Unknown to B

Distance Vector of Node B

Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Unknown to B

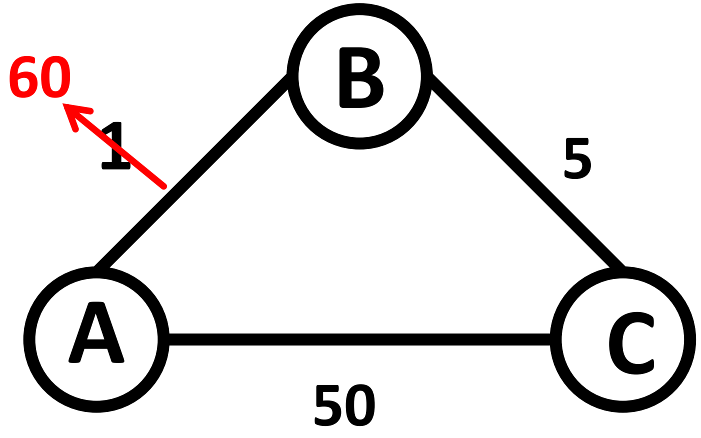
C told B that, **C can reach A** and the **cost is 5**

It looks like we have a path:

B->C-> ... ->A and the **cost is 5 + 6 = 11**

Distance vector: Link Cost Change

How would Node B updates its Distance Vectors?



Distance Vector of Node B

Dest	Cost	Next Hop
A	11	C
C	5	C

C told B that, **C can reach A** and the **cost is 5**

It looks like we have a path:
B->C-> ... ->A and the **cost is 5 + 6 = 11**

Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

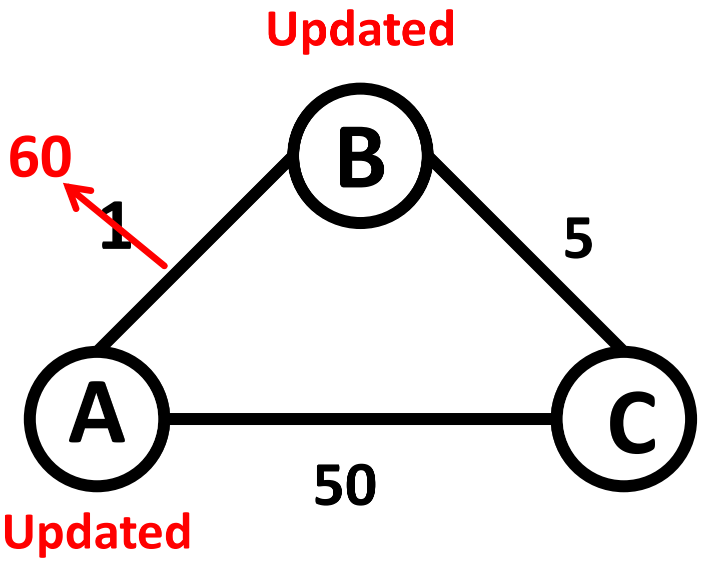
Unknown to B

Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Unknown to B

Distance vector: Link Cost Change



Distance Vector of Node A

Dest	Cost
B	60
C	50

Next Hop
B
C

Distance Vector of Node B

Dest	Cost
A	11
C	5

Next Hop
C
C

Distance Vector of Node C

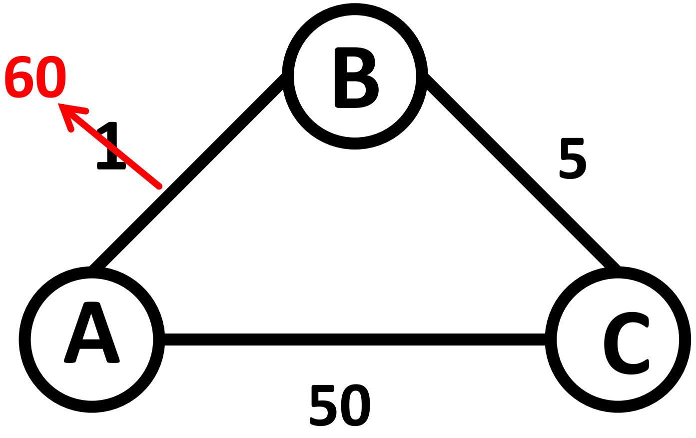
Dest	Cost
A	6
B	5

Next Hop
B
B

Node A will also updates its Distance Vector

Distance vector: Link Cost Change

Node A and B will notify
Node C about their Updates



Distance Vector of Node A

Dest	Cost	Next Hop
B	60	B
C	50	C

Distance Vector of Node C

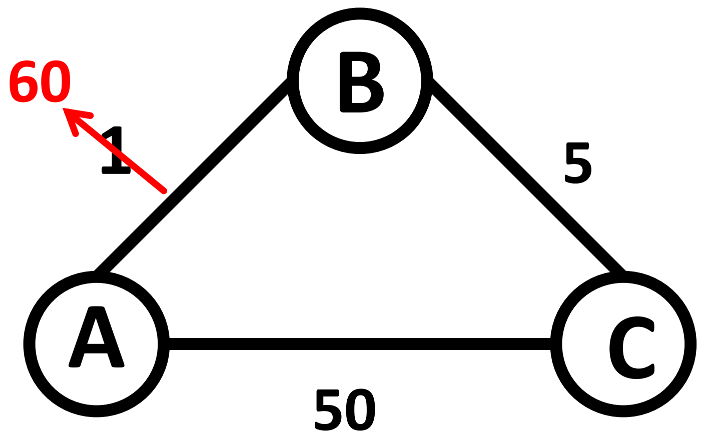
Dest	Cost	Next Hop
A	6	B
B	5	B

Distance Vector of Node B

Dest	Cost	Next Hop
A	11	C
C	5	C

The cost to A via B
increases to 11+5 = 21
Path: C->B-> ... ->A

Distance vector: Link Cost Change



Distance Vector of Node A

Dest	Cost	Next Hop
B	60	B
C	50	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	21	B
B	5	B

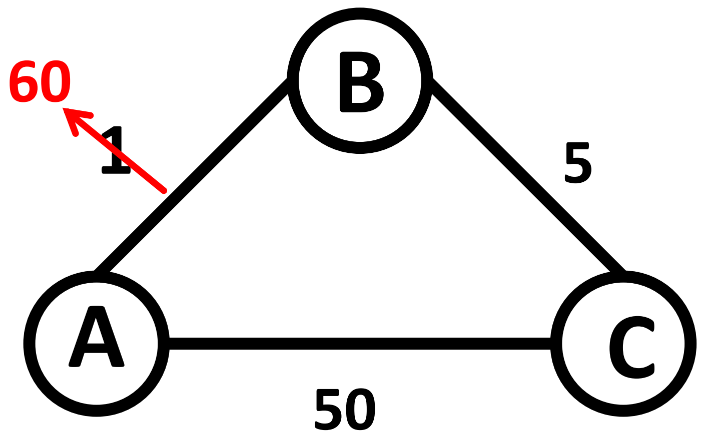
Distance Vector of Node B

Dest	Cost	Next Hop
A	11	C
C	5	C

The cost to A via B increases to 11+5 = 21

Path: C->B-> ... ->A

Distance vector: Link Cost Change



Distance Vector of Node C

Dest	Cost	Next Hop
A	21	B
B	5	B

Distance Vector of Node B

Dest	Cost	Next Hop
A	11	C
C	5	C

The cost to A via C increases to 21+5 = 26

Path: B->C-> ... ->A

Update

Distance Vector of Node B

Dest	Cost	Next Hop
A	26	C
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	21	B
B	5	B

The cost to A via B increases to 26+5 = 31

Path: C->B-> ... ->A

Update

Distance Vector of Node C

Dest	Cost	Next Hop
A	31	B
B	5	B

Distance Vector of Node B

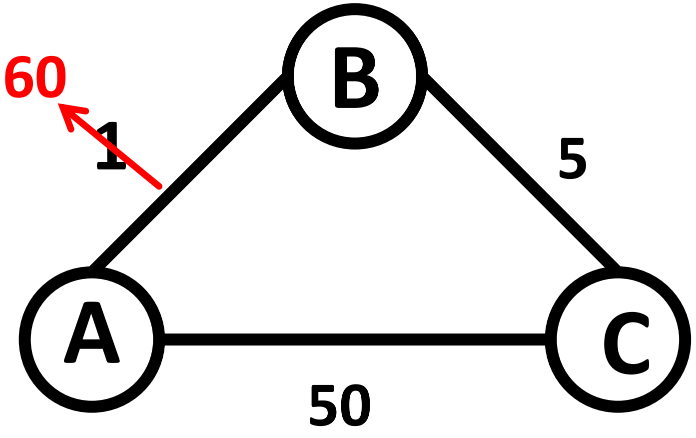
Dest	Cost	Next Hop
A	26	C
C	5	C

Path: B->C-> ... ->A

"bad news travels slow"

count-to-infinity

Distance vector: Link Cost Change --> Solution



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Unknown to B

Distance Vector of Node B

Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

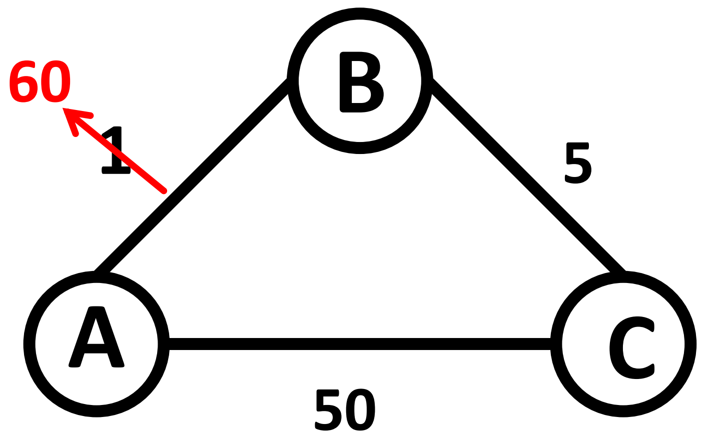
Dest	Cost	Next Hop
A	6	B
B	5	B

Unknown to B

C told B that, **C can reach A** and the **cost is 5**

It looks like we have a path:
B->C-> ... ->A and the **cost is 5 + 6 = 11**

Distance vector: Link Cost Change --> Solution



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Unknown to B

Distance Vector of Node B

Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

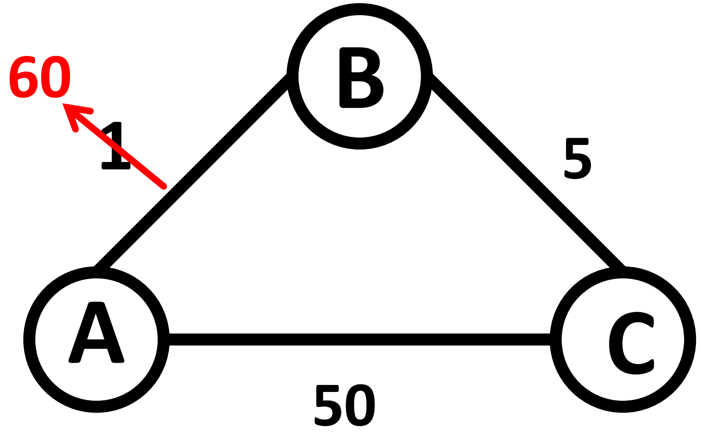
Unknown to B

C told B that, **C can reach A** and the **cost is 5**

It looks like we have a path:
B->C-> B ->A and the **cost is 5 + 6 = 11**

LOOP

Distance vector: Link Cost Change --> Solution



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Distance Vector of Node B

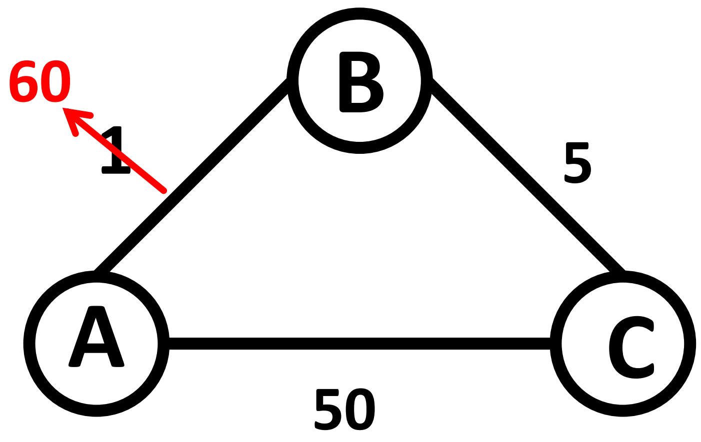
Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
A	6	B
B	5	B

Delete

Distance vector: Link Cost Change --> Solution



Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Distance Vector of Node B

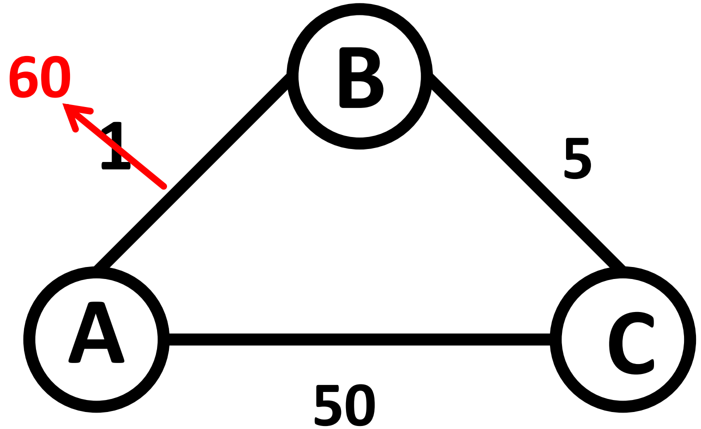
Dest	Cost	Next Hop
A	1	A
C	5	C

Distance Vector of Node C

Dest	Cost	Next Hop
B	5	B

Split Horizon

Distance vector: Link Cost Change --> Solution



Poison reverse

Distance Vector of Node A

Dest	Cost	Next Hop
B	1	B
C	6	B

Distance Vector of Node B

Dest	Cost	Next Hop
A	1	A
C	5	C

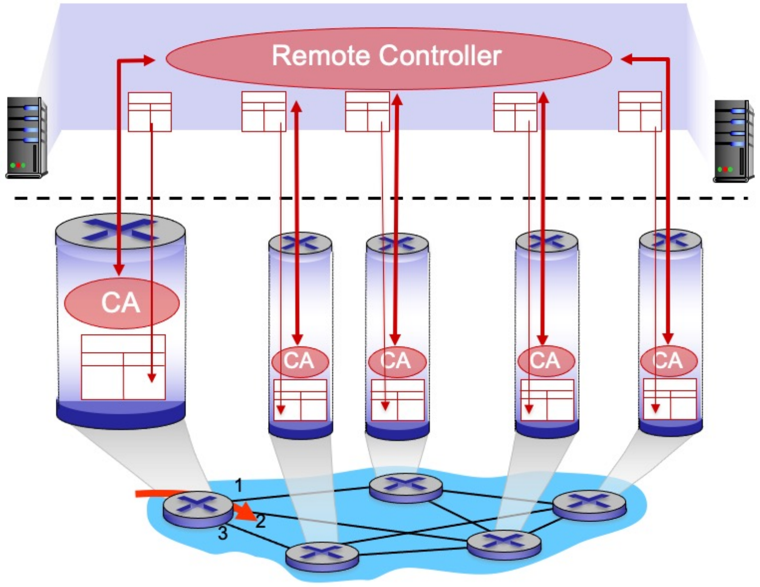
Distance Vector of Node C

Dest	Cost	Next Hop
A	∞	B
B	5	B

Key difference between LS and DV

global: all routers have *complete* topology, link cost info

- "link state" algorithms



decentralized: iterative process of computation, exchange of info with neighbors

- routers initially only know link costs to attached neighbors
- "distance vector" algorithms