

Lecture 15:

Distance Vector Routing

CSE 123: Computer Networks

Aaron Schulman



UCSDCSE



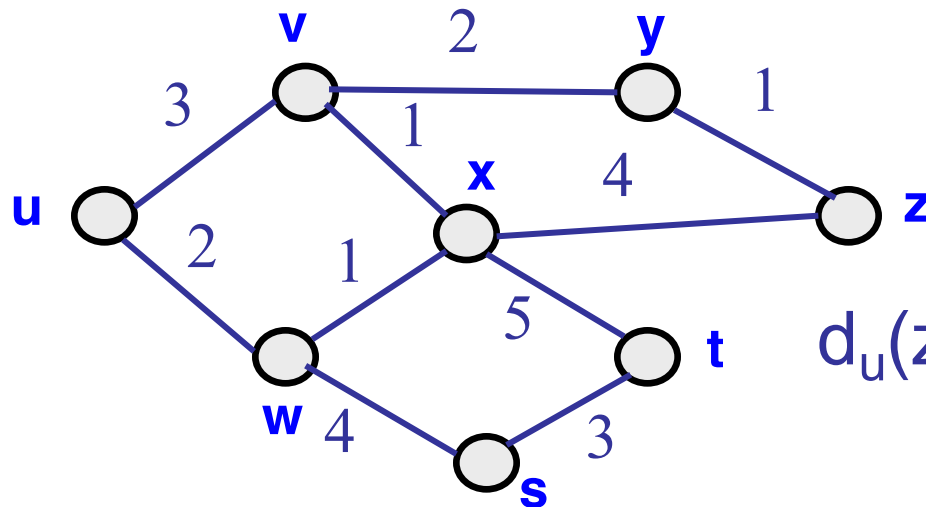
Distance vector algorithm

- Base assumption
 - ◆ Each router knows its **own address** and the cost to reach each of its **directly connected neighbors**
- Bellman-Ford algorithm
 - ◆ Distributed route computation using **only neighbor's info**
- Mitigating loops
 - ◆ Split horizon and poison reverse



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

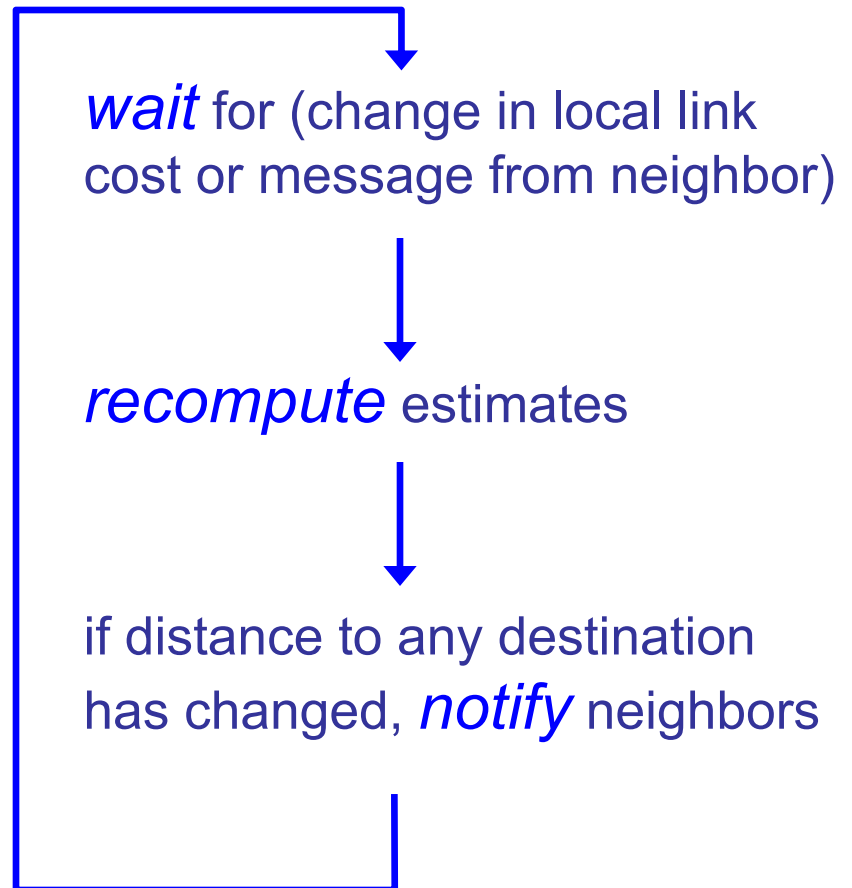
Iterative, asynchronous: each local iteration caused by:

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

Distributed:

- Each node notifies neighbors when its DV changes
- Neighbors then notify their neighbors if necessary

Each node:



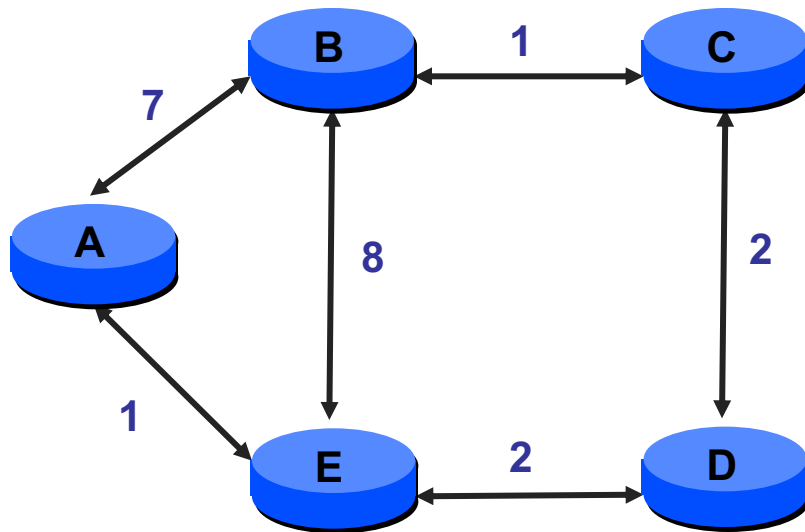


Step-by-Step

- $c(x, v)$ = 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 $D_x = [D_x(y): y \in N]$
- Node x maintains its neighbors' distance vectors
 - ◆ For each neighbor v , x maintains $D_v = [D_v(y): y \in N]$
- 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 $y \in N$



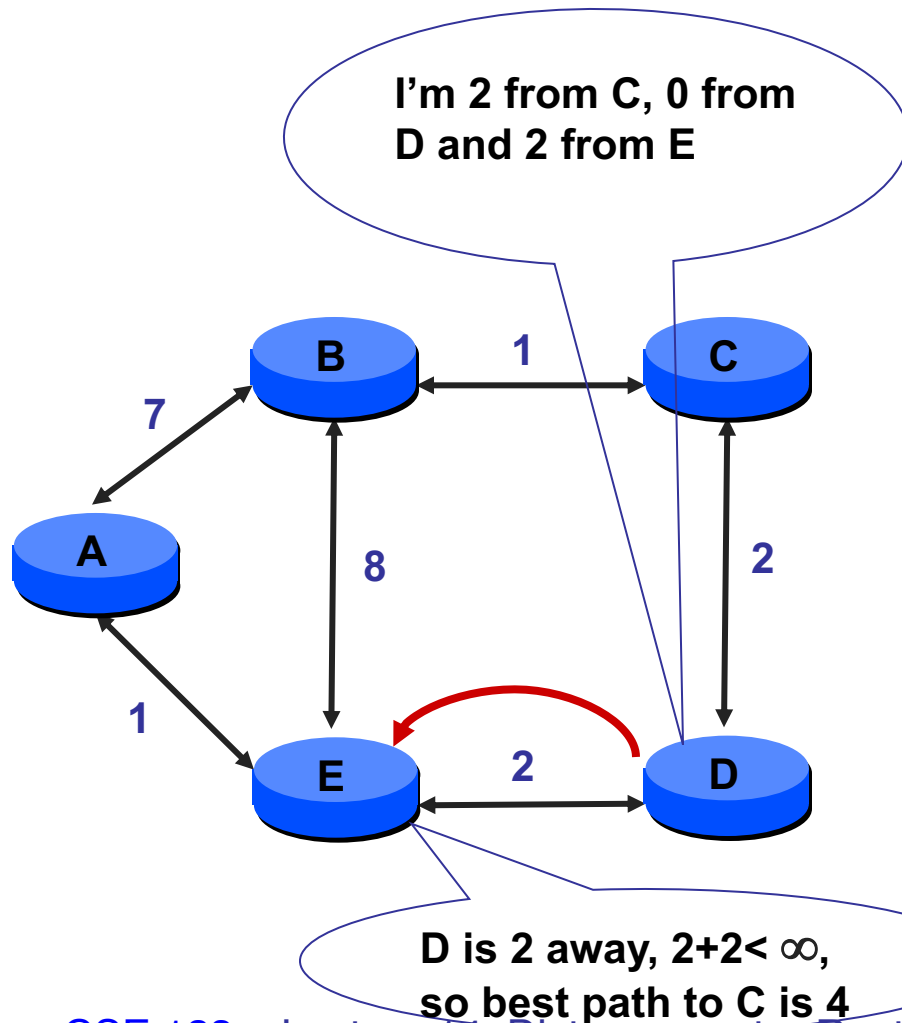
Example: Initial State



Info at node	Distance to Node				
	A	B	C	D	E
A	0	7	∞	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	∞	2	0



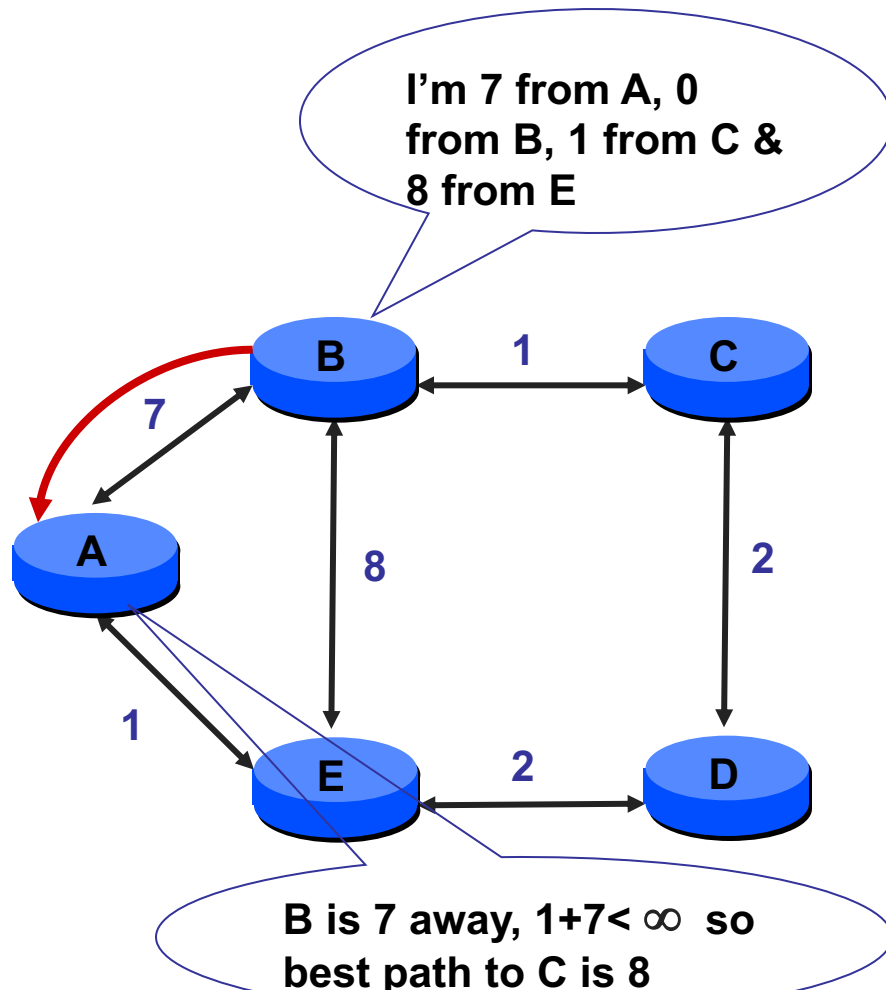
D sends vector to *E*



Info at node	Distance to Node				
	A	B	C	D	E
A	0	7	∞	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	4	2	0



B sends vector to A

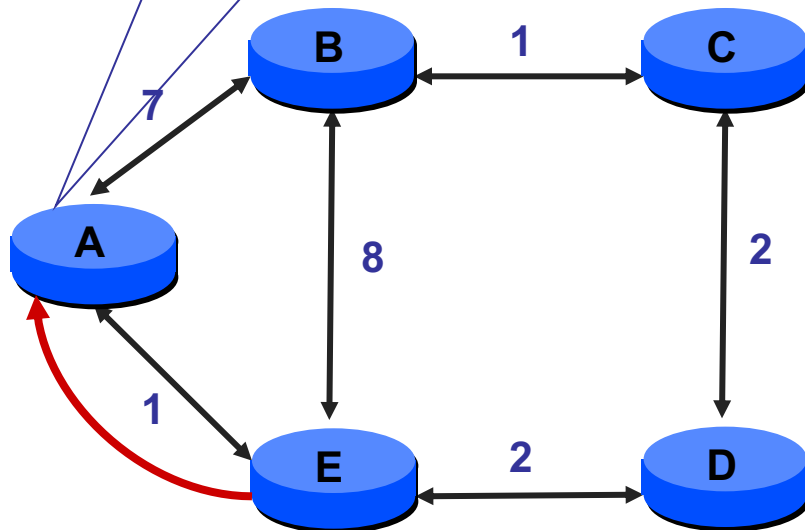


Info at node	Distance to Node				
	A	B	C	D	E
A	0	7	8	∞	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	4	2	0



E sends vector to A

E is 1 away, $4+1 < 8$
so C is 5 away, $1+2 < \infty$
so D is 3 away

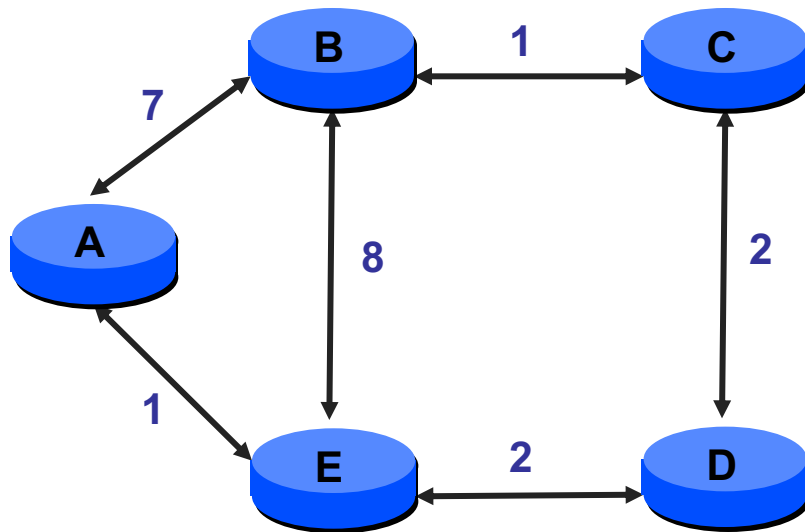


I'm 1 from A, 8 from B, 4
from C, 2 from D & 0 from E

Info at node	Distance to Node				
	A	B	C	D	E
A	0	7	5	3	1
B	7	0	1	∞	8
C	∞	1	0	2	∞
D	∞	∞	2	0	2
E	1	8	4	2	0



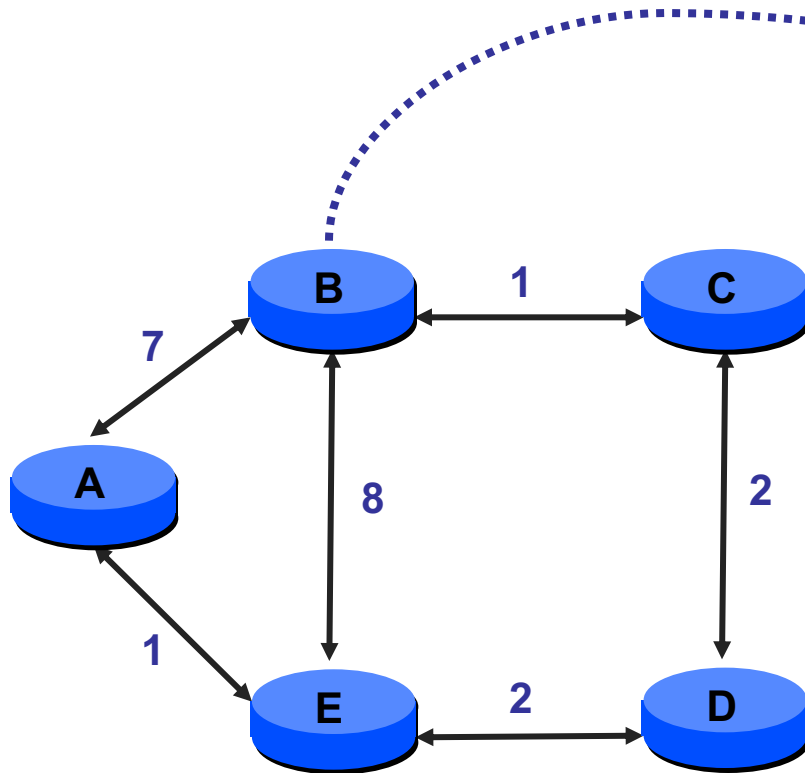
...until Convergence



Info at node	Distance to Node				
	A	B	C	D	E
A	0	6	5	3	1
B	6	0	1	3	5
C	5	1	0	2	4
D	3	3	2	0	2
E	1	5	4	2	0



Node *B*'s distance vectors



Dest	Next hop		
	A	E	C
A	7	9	6
C	12	12	1
D	10	10	3
E	8	8	5