Distance Vector Routing

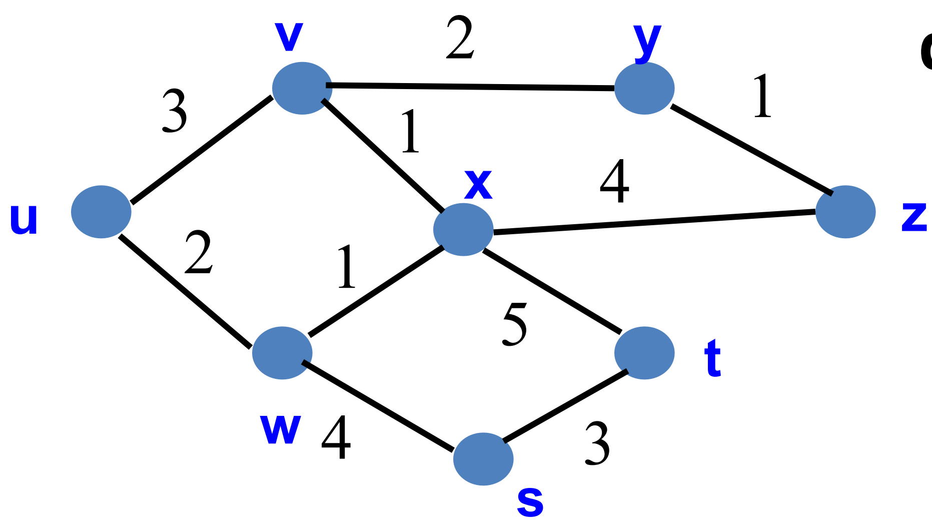
Credits: Prof. Sangtae Ha

Distance Vector:

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

Distance Vector: Bellman-Ford Algo

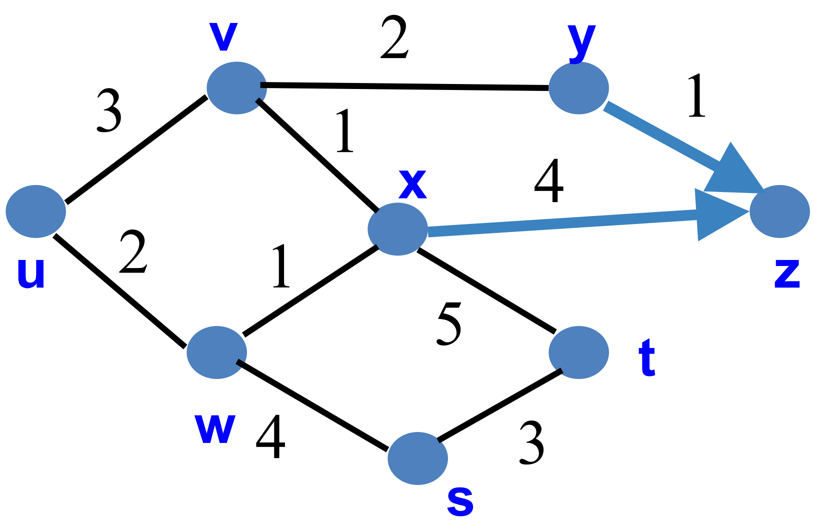
- Define distances at each node x
  - $d_x(y)$ = cost of least-cost path from x to y
- Update distances based on neighbors
  - $d_x(y) = \min \{c(x,v) + d_v(y)\}$ over all neighbors v

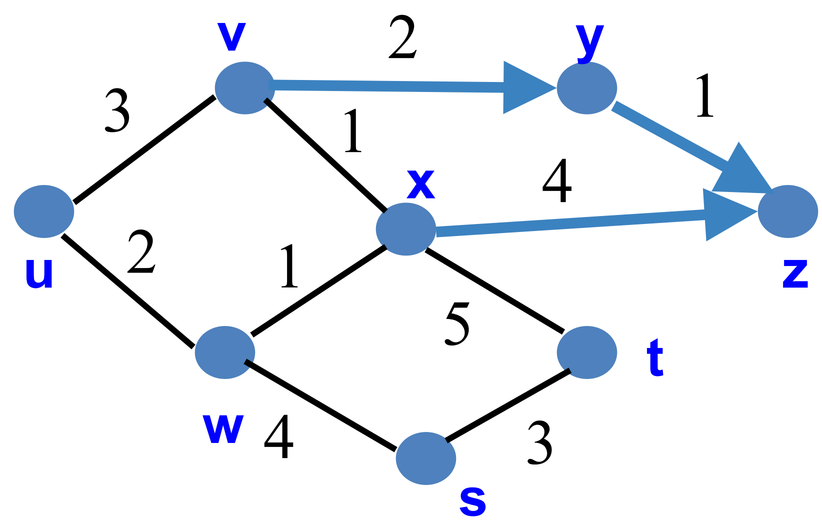


$$d_u(z) = \min\{ c(u,v) + d_v(z), \\ c(u,w) + d_w(z)\}$$

Used in RIP and EIGRP

Distance Vector Example



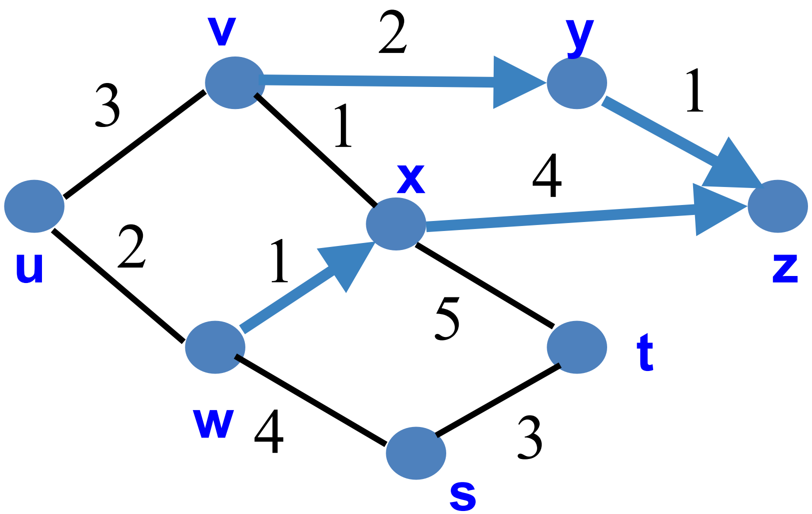


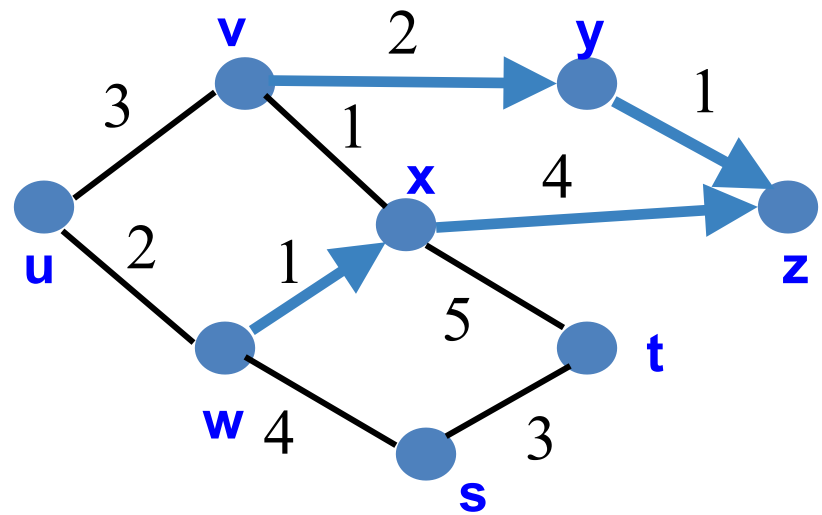
$d_y(z) = 1$

$d_x(z) = 4$

$$d_v(z) = \min\{\ 2 + d_y(z),\ 1 + d_x(z)\ \} = 3$$

Distance Vector Example (Cont.)





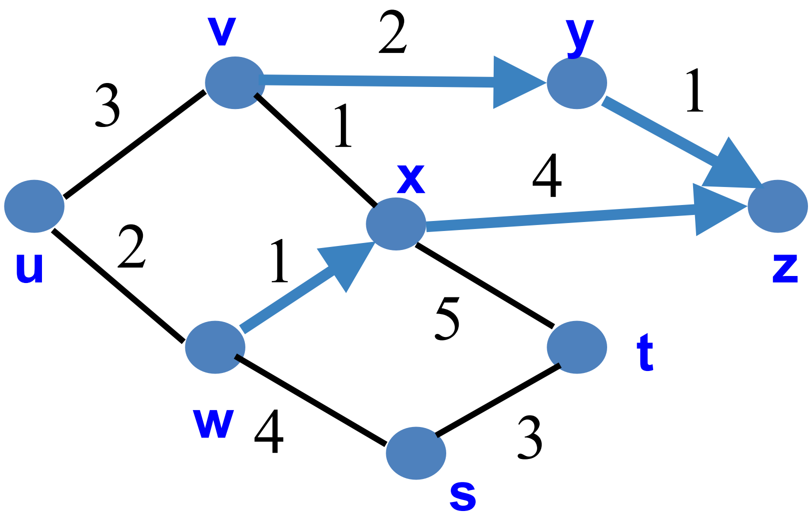
$$d_w(z) = \min\{\ 1+d_x(z),\ 4+d_s(z),\ 2+d_u(z)\ \}$$

$$= 5$$

$d_u(z)$ =

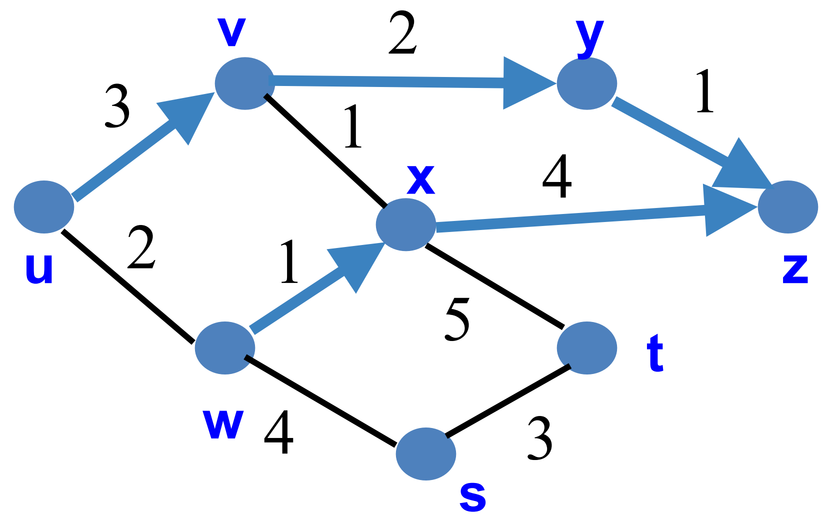
(A) 5 (B) 6 (C) 7

Distance Vector Example (Cont.)



$$d_w(z) = \min\{\ 1+d_x(z),\ 4+d_s(z),\ 2+d_u(z)\ \}$$

$$= 5$$



$$d_u(z) = \min\{\ 3+d_v(z),\ 2+d_w(z)\ \}$$

$$= 6$$

Distance Vector Example 2: 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

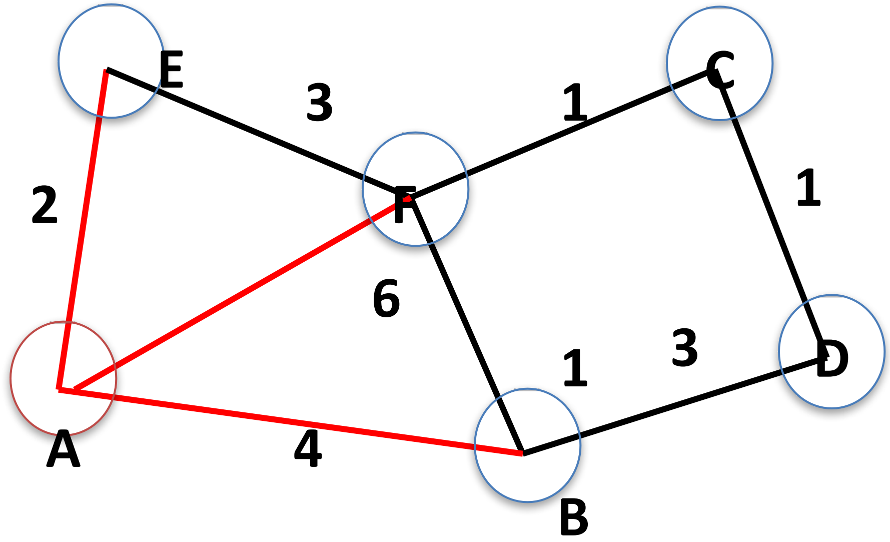


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	∞	–	A	∞	–	A	2	A	A	6	A
B	∞	–	B	3	B	B	∞	–	B	1	B
C	0	C	C	1	C	C	∞	–	C	1	C
D	1	D	D	0	D	D	∞	–	D	∞	–
E	∞	–	E	∞	–	E	0	E	E	3	E
F	1	F	F	∞	–	F	3	F	F	0	F

Explanation:

1. Initially all the nodes will only know the distance of its immediate neighbors.
2. Hence, the table for each node comprises of the Destination (Dst), Cost(Cst), Next hop for every other router in the network from an originating router.
3. Consider node A. A is connected to routers B,E and F directly. Hence, the link costs to them is stated in the table along with the next hop. All rest nodes cost is infinite from node A for now.
4. Such table is generated for every other node.

Distance Vector Example 2: 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

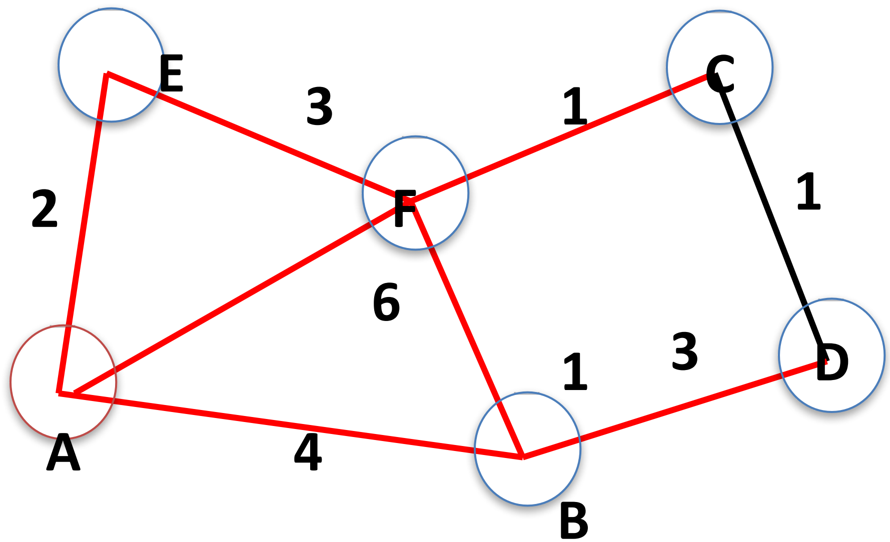


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	7	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	∞	–	D	2	C
E	4	F	E	∞	–	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

Explanation:

Each node has reported the information it had in the preceding step
to its immediate neighbors

1. Now every router will give its routing table generated in first step to their neighbor.
2. Eg: E, B and F will give their routing tables to A.
3. A will now check if it is able to reach more nodes at a lower cost.
4. After receiving routing table of E,B and F; A will be able to reach C and D which was previously infinite.
5. It also gets that to reach F, initially marked with cost 6, has got a lower cost path of 5 via E.
6. This is done for all the nodes.
7. Such process of sending routing table to its neighbors will keep happening till the entire network converges. (next slide)

Distance Vector Example 2: Step 3

Optimum 3-hop paths

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

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

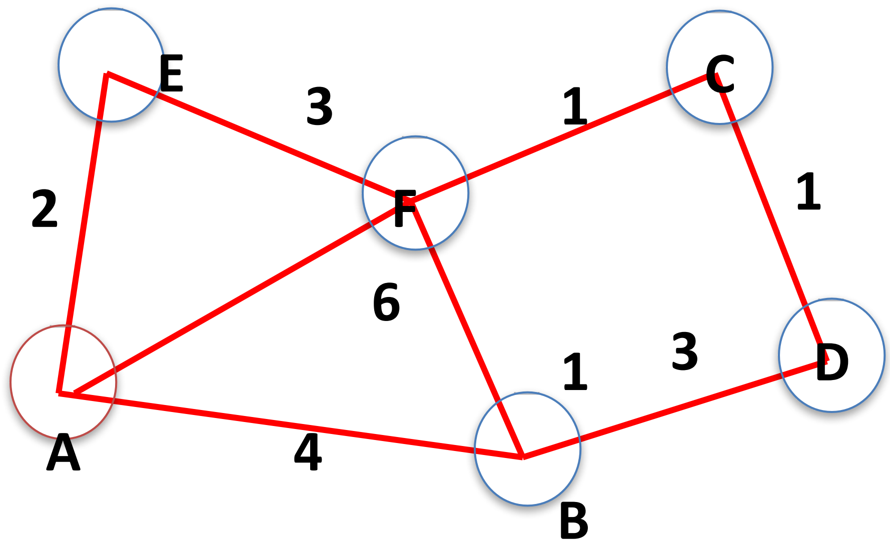


Table for C		
Dst	Cst	Hop
A	6	F
B	2	F
C	0	C
D	1	D
E	4	F
F	1	F

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

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

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