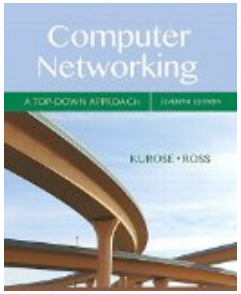
COMP 375: Lecture 32



- **News & Notes:**
  - Quiz #7 in class today
  - Project #5
    - **Protocol Spec Due**: Monday (April 23)
    - **Code Due**: Mon, April 30
- **Reading (Mon, Apr. 23)**
  - None (review today's reading)

Quiz #7

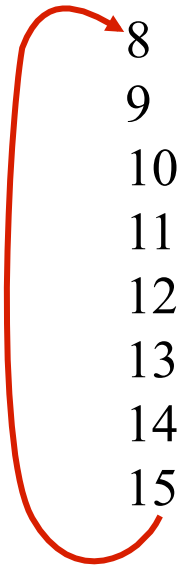
- Closed book. Closed notes.
- *Happy National Pineapple Upside-Down Cake Day!!!*

Section 5.2

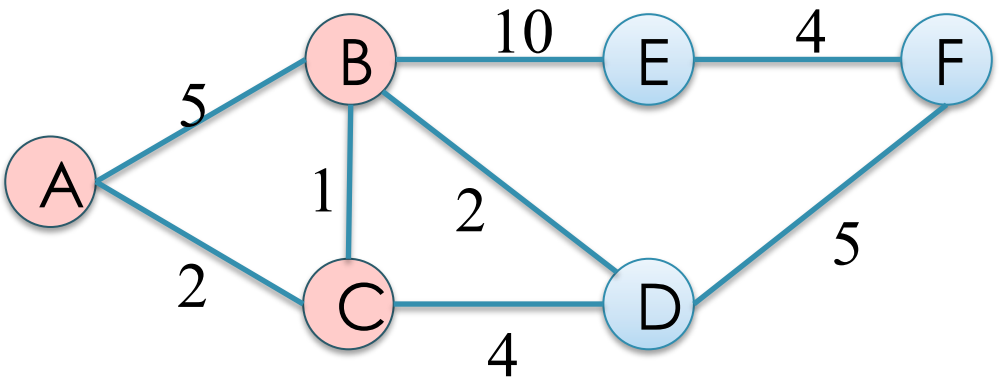
LINK-STATE ROUTING

In Dijkstra's Algorithm we calculate one new best cost each iteration.

```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v adjacent to u
5         then D(v) = c(u,v)
6     else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for all v adjacent to w and not in N' :
12      D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```



Dijkstra's Algorithm – Step 2



Choose B.

Previous Step

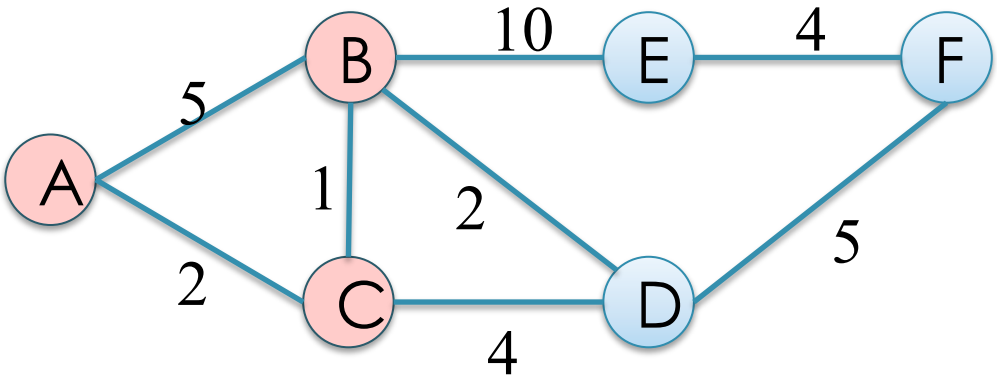
Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞

Pick Min

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D		
E		
F		

Fill out the rest of "This Step" table for this iteration.



Previous Step

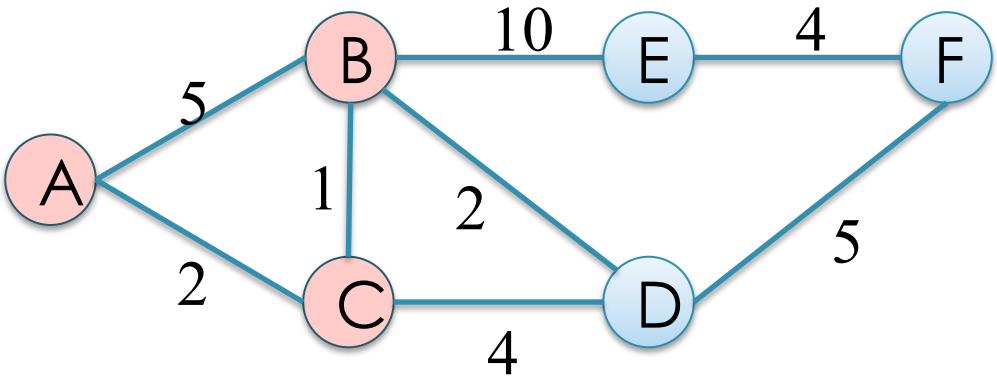
Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞

Pick Min

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D		
E		
F		

Which node will be added to N' during this iteration (i.e. which goes red next)?



A.	Node D
B.	Node E
C.	Node F
D.	Either D or E
E.	None

Previous Step

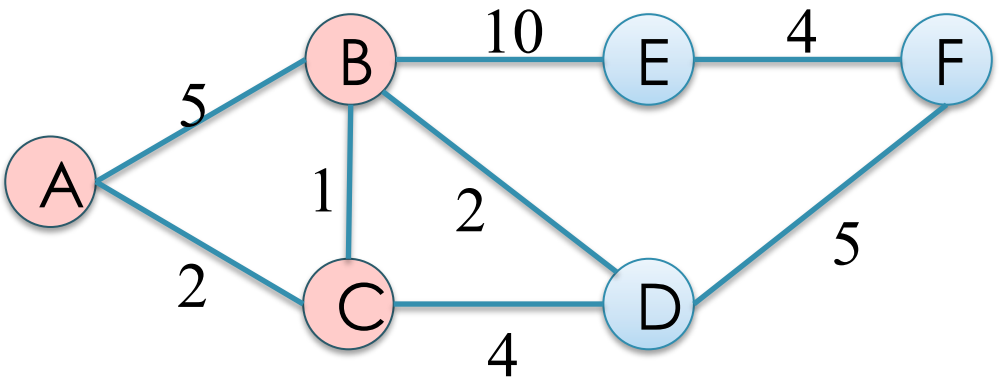
Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞

Pick Min

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D		
E		
F		

Dijkstra's Algorithm – Step 2



Consider path to D:

D(D) = 6
or
D(B) + cost(B, D)
3 + 2 = 5

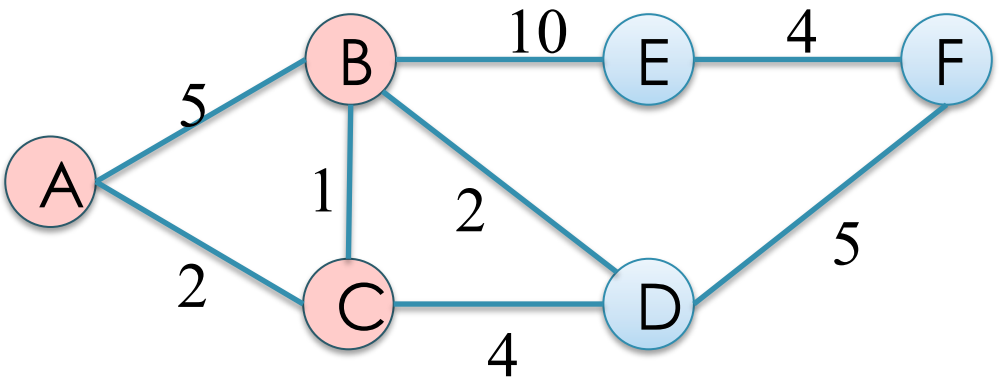
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E		
F		

Dijkstra's Algorithm – Step 2



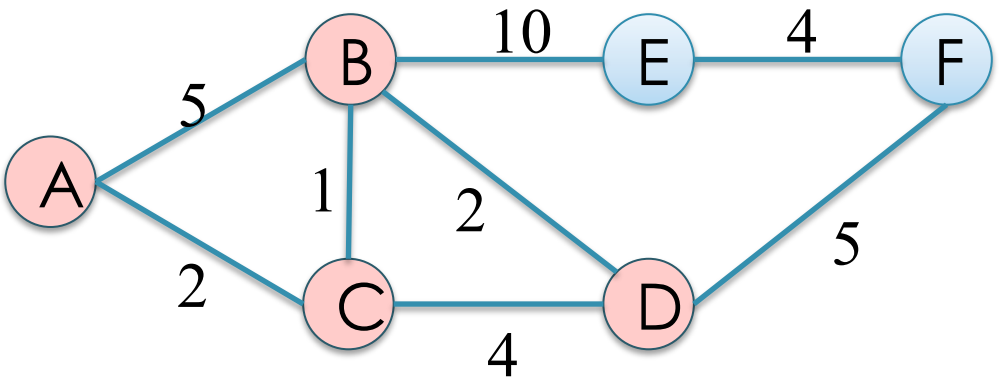
Consider path to E:

D(E) = ∞
or
D(B) + cost(B, E)
3 + 10 = 13

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, D	6
E	?	∞
F	?	∞

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	?	∞

Dijkstra's Algorithm – Step 3



Choose D.

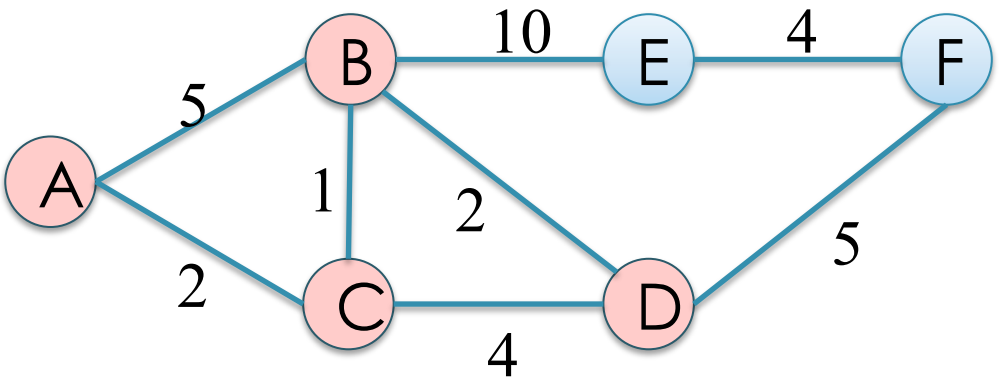
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E		
F		

Dijkstra's Algorithm – Step 3



No change for E.

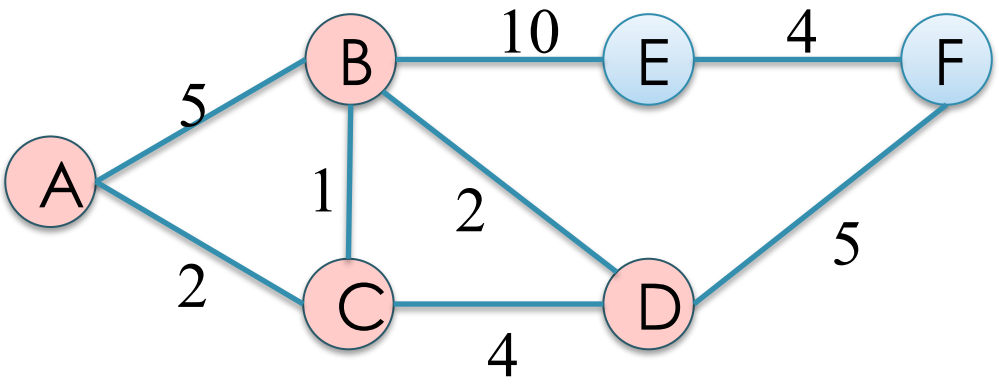
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F		

Dijkstra's Algorithm – Step 3



Consider path to F:

D(F) = ∞
or
D(D) + cost(D, F)
5 + 5 = 10

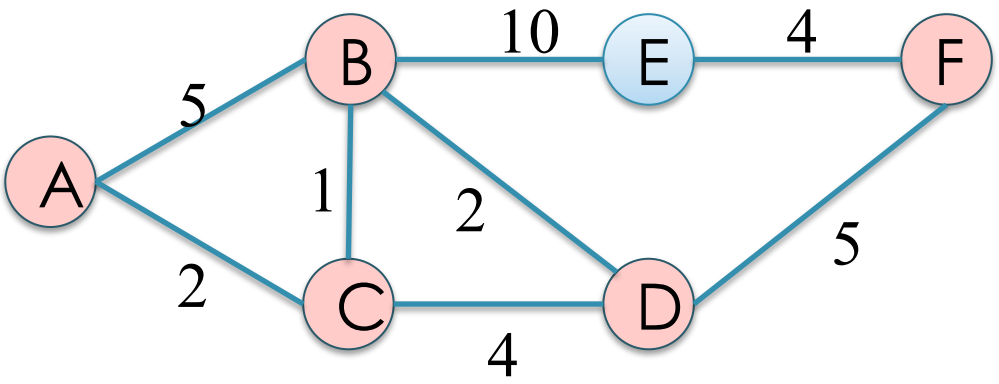
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	?	∞

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

Dijkstra's Algorithm – Step 4



Choose F.

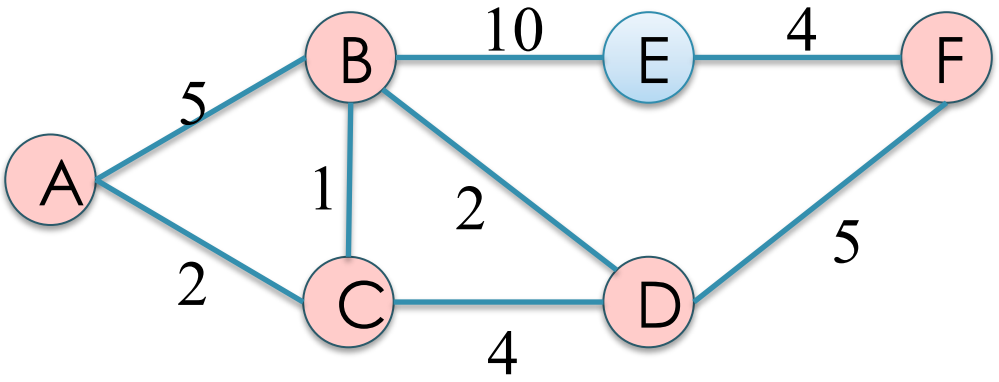
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
F	C, B, D, F	10

Dijkstra's Algorithm – Step 4



Consider path to E:

D(E) = 13
or
D(F) + cost(F, E)
10 + 4 = 14

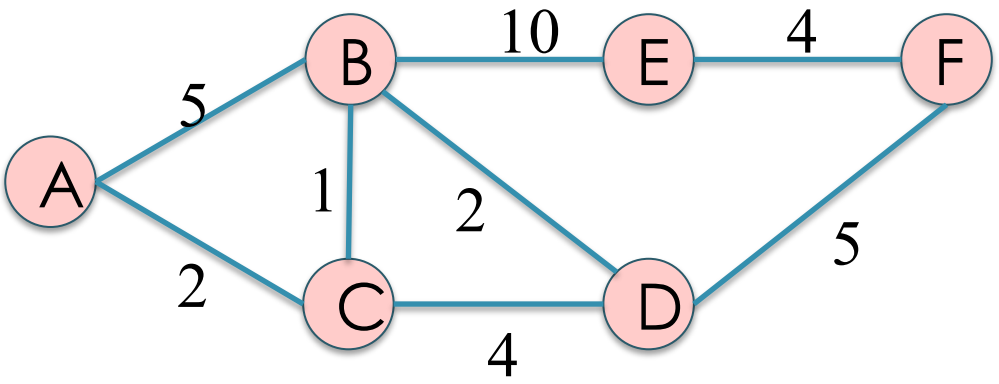
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

Dijkstra's Algorithm – Step 5



Choose E.

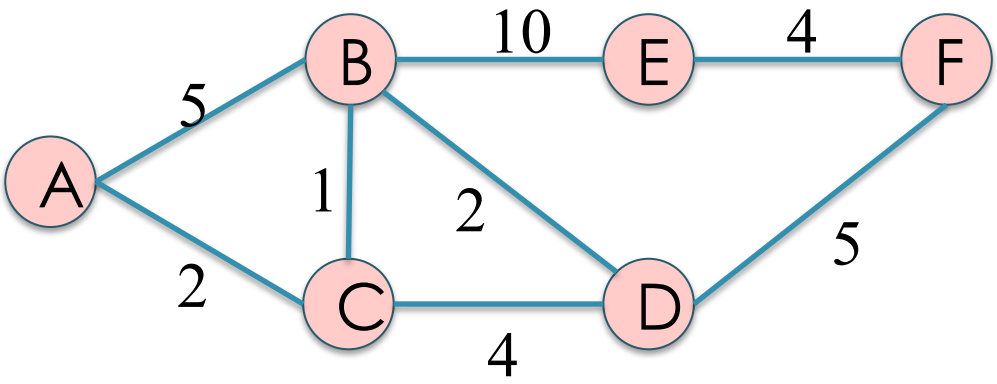
Previous Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

This Step

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

Dijkstra's Algorithm – Done!



Final Answer

Dest	Path	Cost D(v)
A	A	0
B	C, B	3
C	C	2
D	C, B, D	5
E	C, B, E	13
F	C, B, D, F	10

Populate Forwarding Table

Forwarding Table

Dest	Forward To
B	C
C	C
D	C
E	C
F	C

Dijkstra's Algorithm is naively an $O(N^2)$ algorithm, but can be made more efficient.

- As previously described: $O(N^2)$
  - At each step, there are N nodes to choose next
  - Total of N steps

- Fastest known is $O(N*log_2(N) + E)$
  - Uses a min-heap

Section 5.2

DISTANCE VECTOR ROUTING

Bellman-Ford Equation

let

$d_x(y)$:= cost of least-cost path from x to y

then

$d_x(y) = min_v \{c(x,v) + d_v(y)\}$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

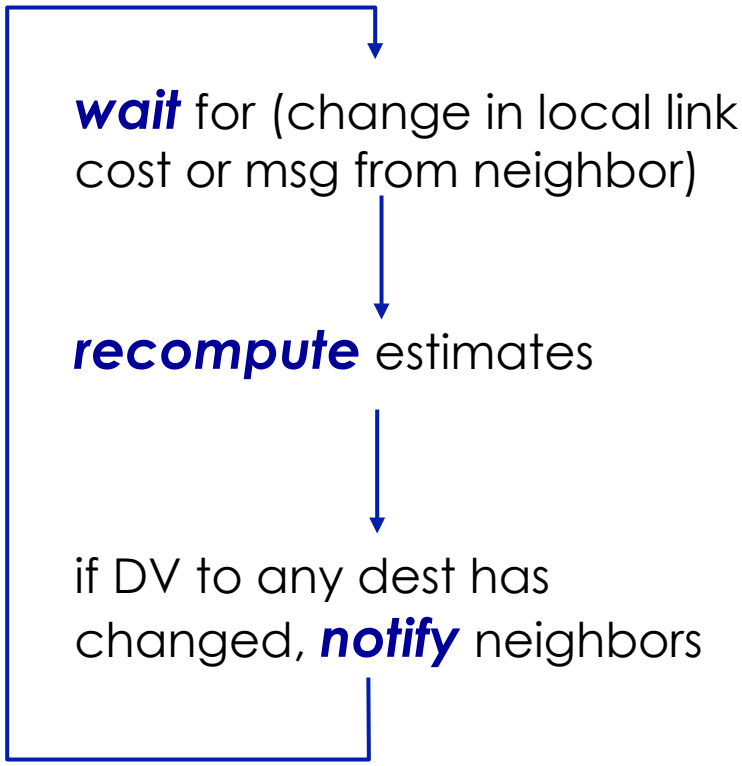
A node's **distance vector** is its estimated costs to every other node in the graph.

$$D_x = [D_x(y): y \in N]$$

- Node x knows the following about neighbor v:
  - Its cost to v: c(x,v)
  - v's Distance Vector: $\mathbf{D}_v = [D_v(y): y \in N]$

DV is iterative and asynchronous, with link cost changes triggering updates.

Each node:



$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$

x's table

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

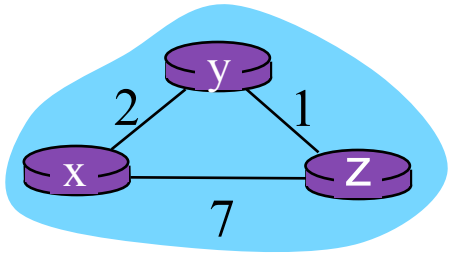
	x	y	z
x	0		
y	2	0	1
z	7	1	0

y's table

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

z's table

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



time

Key: cost to

from \	x	y	z
x			
y			
z			

x's table

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

y's table

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

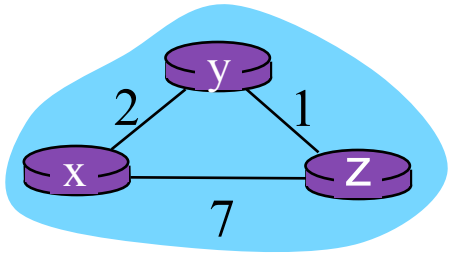
	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

z's table

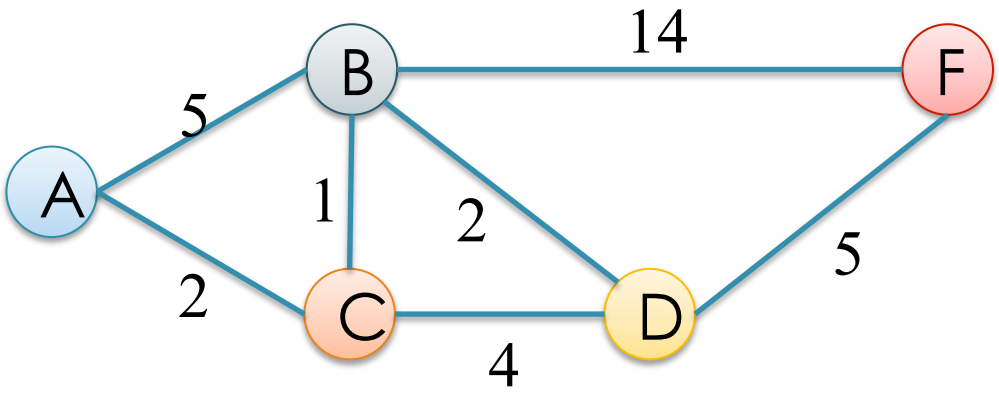
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



Distance Vector Example



Goal: Compute **routing table** for each router.

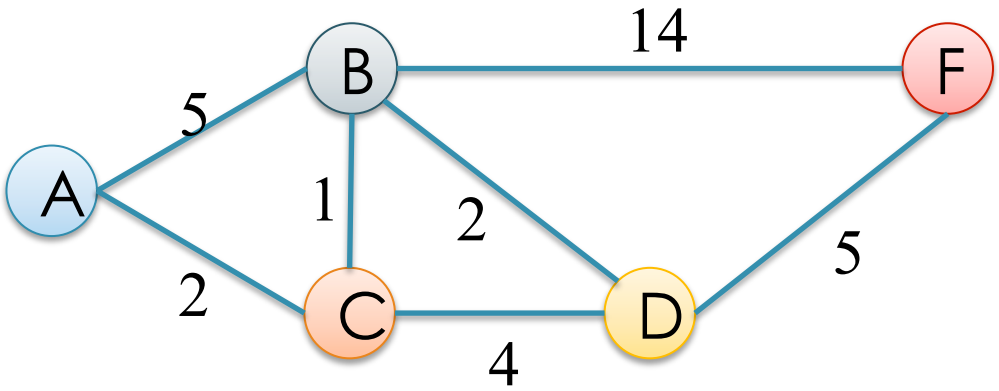
Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	10

A's Routing Table

Dest	Next Hop	Cost
B	C	3
C	C	2
D	C	5
F	C	10

A's Forwarding Table

Distance Vector – Round 0



Routers populate their forwarding table by taking the row minimum.

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C		1		
D			2	
F				14

Router C

Via→ ↓ To	A	B	D
A	2		
B		1	
D			4
F			

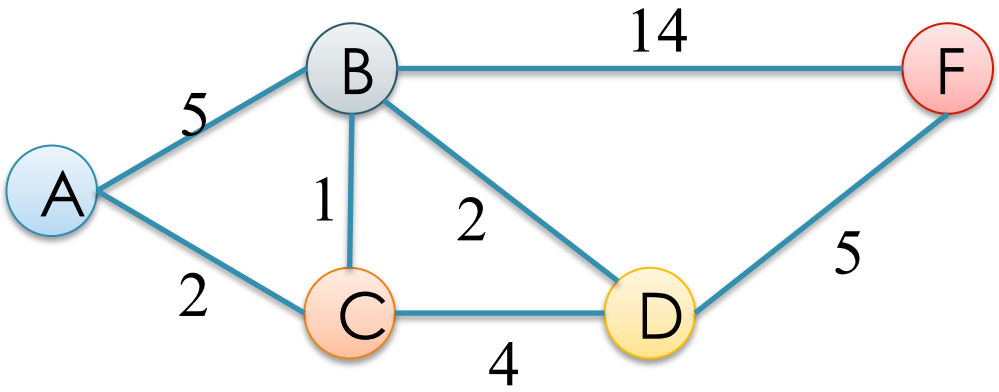
Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Distance Vector – Round 0



Routers exchange their local vectors with direct neighbors.
We'll assume they all exchange at once (synchronous).
(Not realistic)

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C		1		
D			2	
F				14

Router C

Via→ ↓ To	A	B	D
A	2		
B		1	
D			4
F			

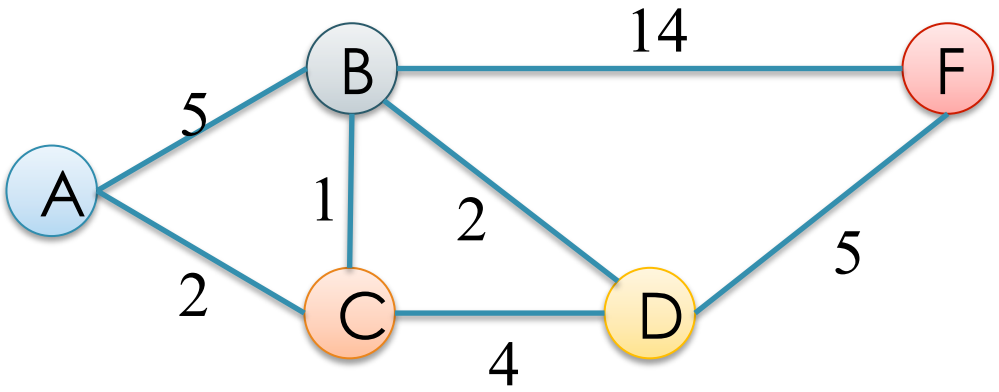
Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Distance Vector – Round 1



A will send to neighbors (B & C):
I can get to B in 5 and C in 2.

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C	7	1		
D			2	
F				14

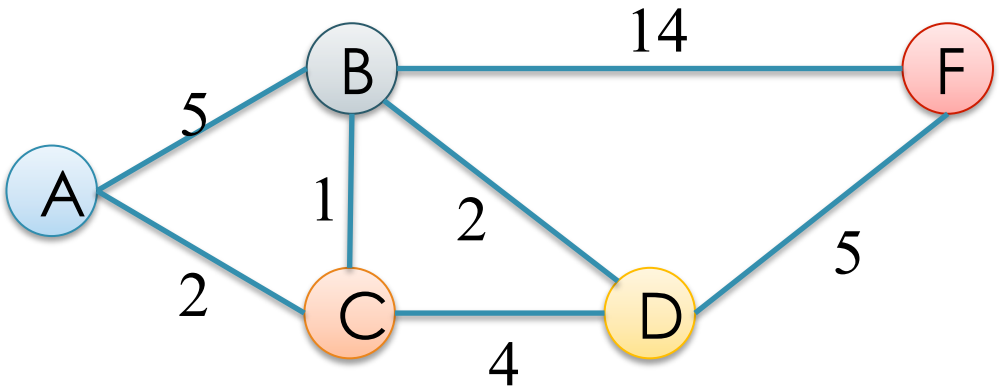
Router C

Via→ ↓ To	A	B	D
A	2		
B	7	1	
D			4
F			

Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Distance Vector – Round 1



B will send to neighbors (A, C, D, F):
I can get to A in 5, C in 1, D in 2, and F in 14.

Router F

Via→ ↓ To	B	D
A	19	
B	14	
C	15	
D	16	5

Router A

Via→ ↓ To	B	C
B	5	
C	6	2
D	7	
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5			
C	7	1		
D			2	
F				14

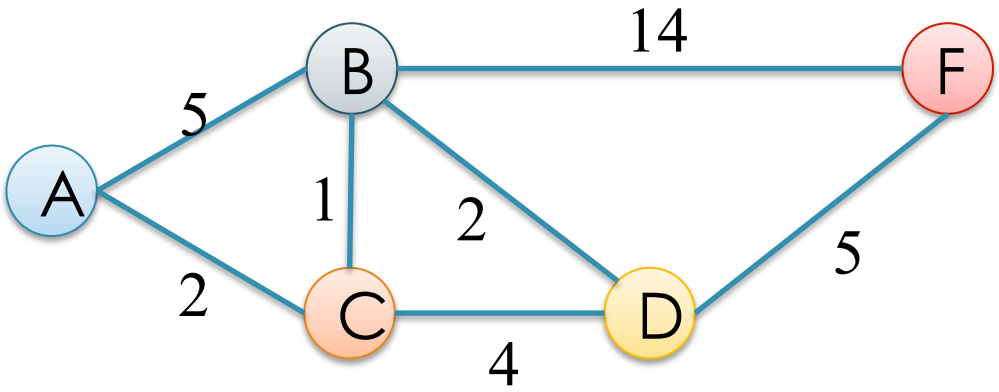
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	
D		3	4
F		15	

Router D

Via→ ↓ To	B	C	F
A	7		
B	2		
C	3	4	
F	16		5

Distance Vector – Round 1



C will send to neighbors (A, B, D):
I can get to A in 2, B in 1, and D in 4.

Router F

Via→ ↓ To	B	D
A	19	
B	14	
C	15	
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1		
D		5	2	
F				14

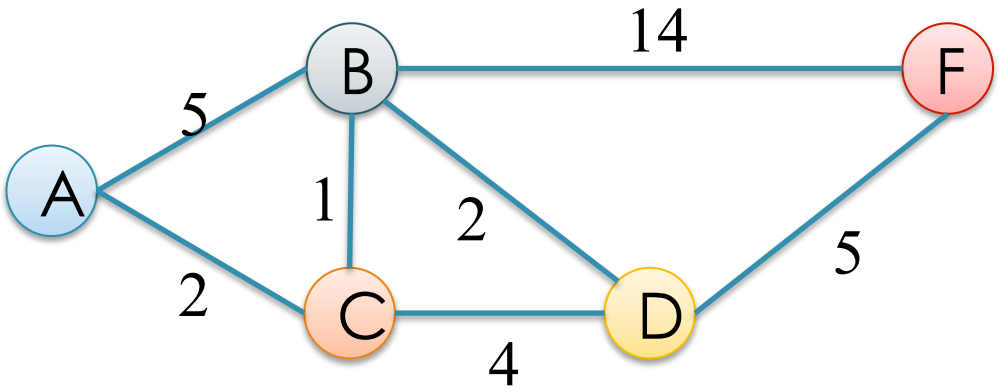
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	
D		3	4
F		15	

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	
C	3	4	
F	16		5

Distance Vector – Round 1



D will send to neighbors (B, C, F):
I can get to B in 2, C in 4, and F in 5.

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	
F			7	14

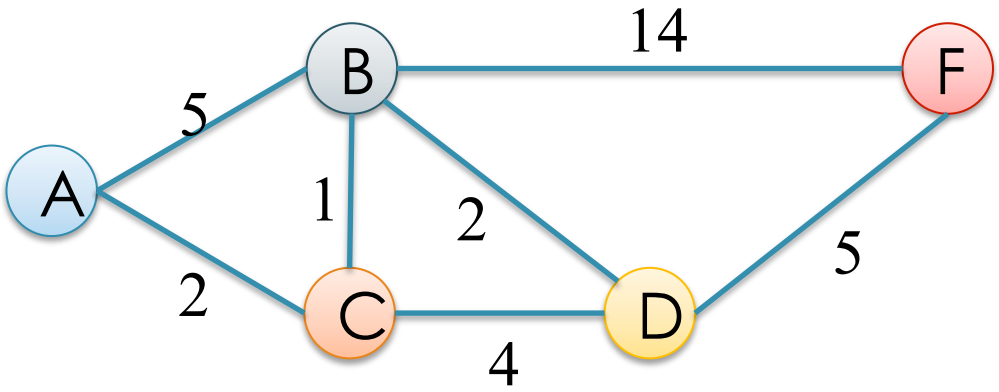
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	
C	3	4	
F	16		5

Distance Vector – Round 1



F will send to neighbors (B, D):
I can get to B in 14, D in 5.

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	19
C	3	4	
F	16		5