

CS 3800: Computer Networks

Lecture 9: Routing Algorithms

Instructor: John Korah

Acknowledgement

- The following slides include material from author resources for:
 - KR Text book
 - “Data and computer communications,” William Stallings, Tenth edition

Learning Goals

- Understand graph abstraction of networks
- Understand how commonly used shortest distance algorithms can be used to generate routing tables.
- Understand the working of two key algorithms: Link State and Distance Vector algorithms.

Topics

- **Introduction**
 - Types of routing algorithms
- **Routing algorithms**
 - Link state
 - Distance vector
- **Scalable routing**

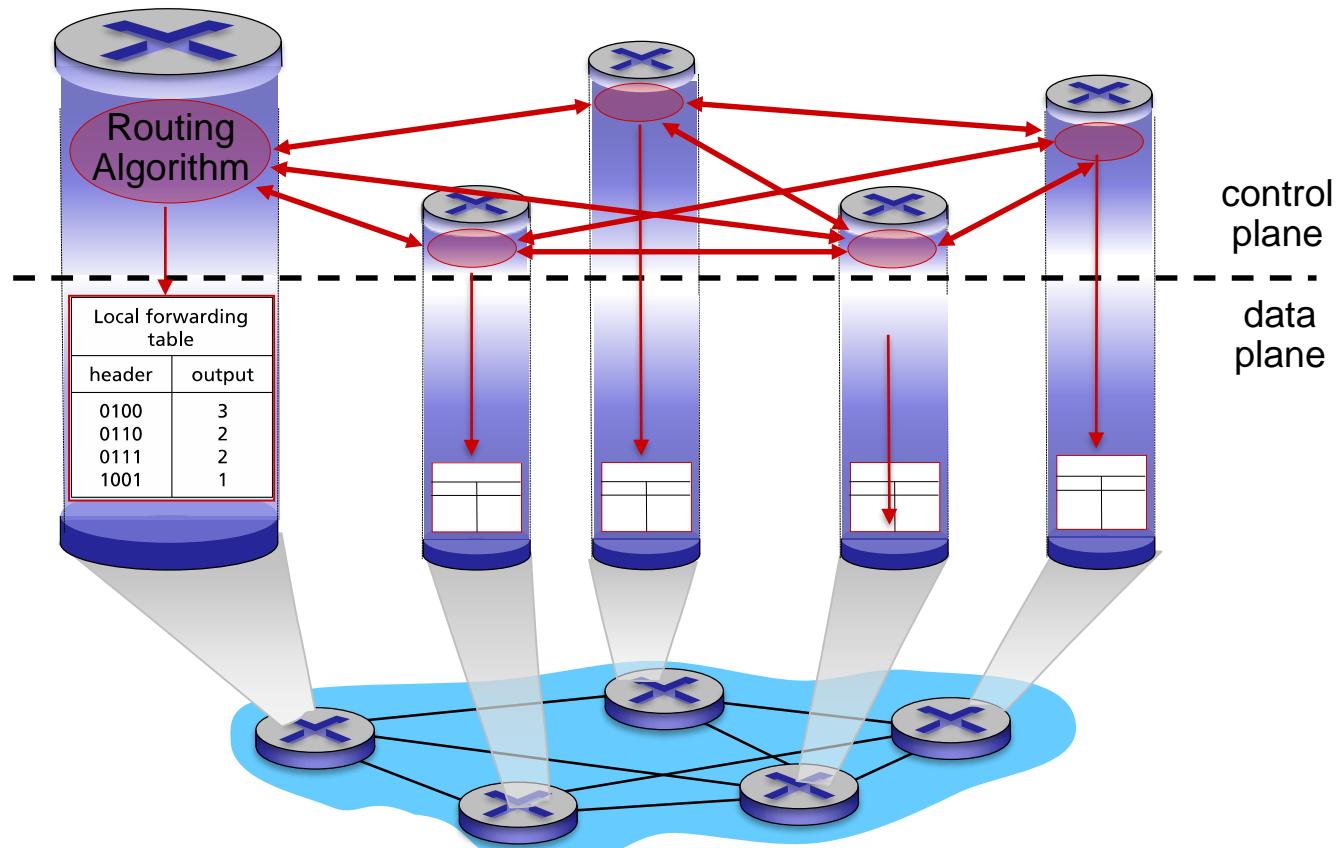
Recall: Network-layer functions

Recall: two network-layer functions:

- *forwarding*: move packets from router's input to appropriate router output *data plane*
- *routing*: determine route taken by packets from source to destination *control plane*

Recall: Per-router control plane

Individual routing algorithm components *in each and every router* interact with each other in control plane to compute forwarding tables

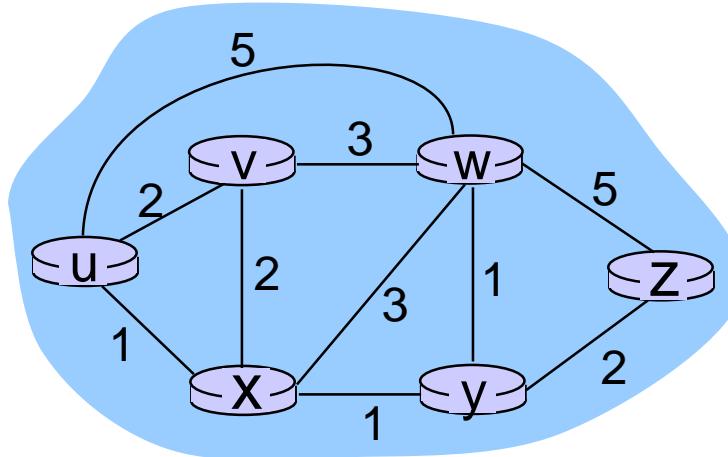


Routing protocols

Routing protocol goal: determine “good” paths (equivalently, routes), from sending hosts to receiving host, through network of routers

- path: sequence of routers packets will traverse in going from given initial source host to given final destination host
- “good”: least “cost”, “fastest”, “least congested”
- routing: a “top-10” networking challenge!

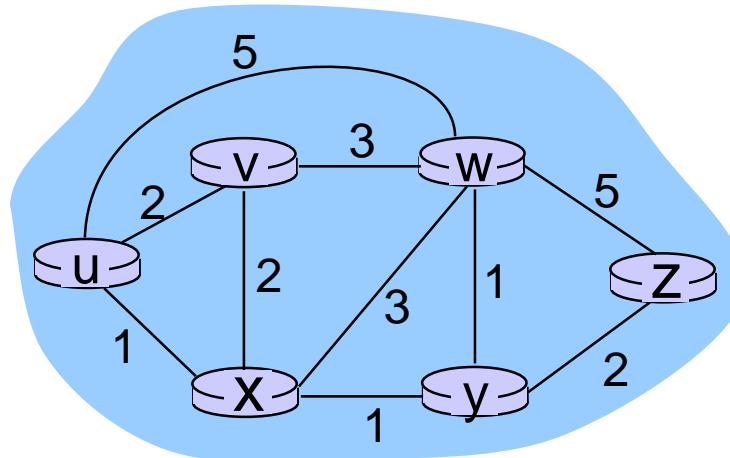
Graph abstraction of the network



Consider the network in the figure above.

- Can we represent this as a graph?
- How?

Graph abstraction of the network

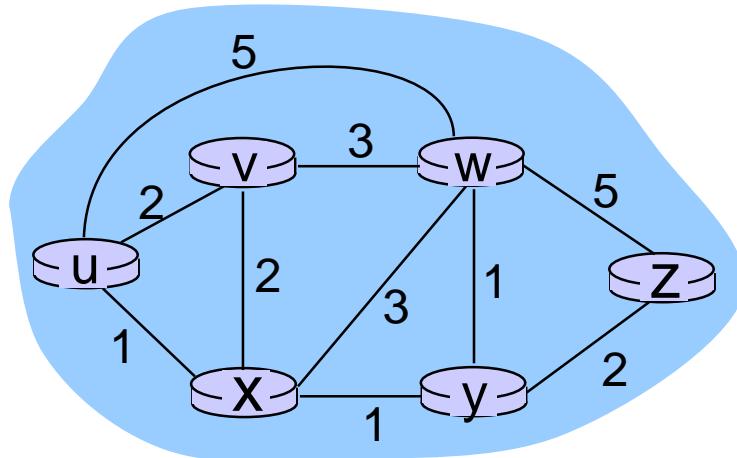


graph: $G = (V, E)$

V = set of node/routers = { u, v, w, x, y, z }

E = set of links = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Graph abstraction: costs



graph: $G = (V, E)$

V = set of node/routers = { u, v, w, x, y, z }

E = set of links = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

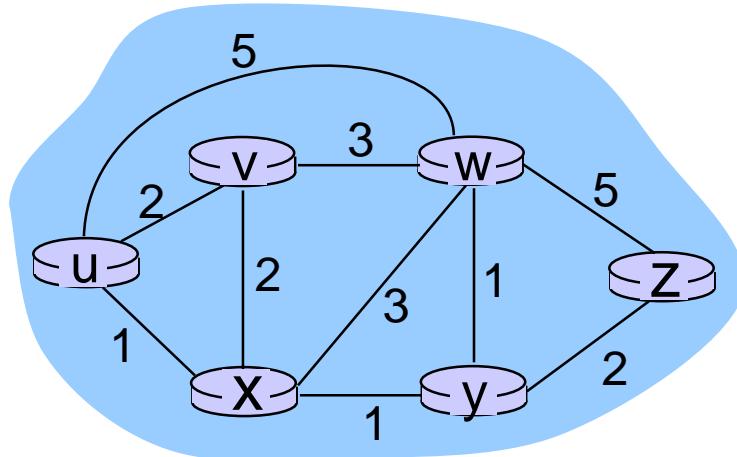
$c(x,y)$ = cost of link (x,y)

e.g., $c(w,z) = 5$

Cost is defined by the algorithm designer:

- Could be 1 to indicate connectivity, or
- Inversely related to bandwidth, or
- inversely related to congestion
- ...

Graph abstraction: costs



graph: $G = (V, E)$

V = set of node/routers = { u, v, w, x, y, z }

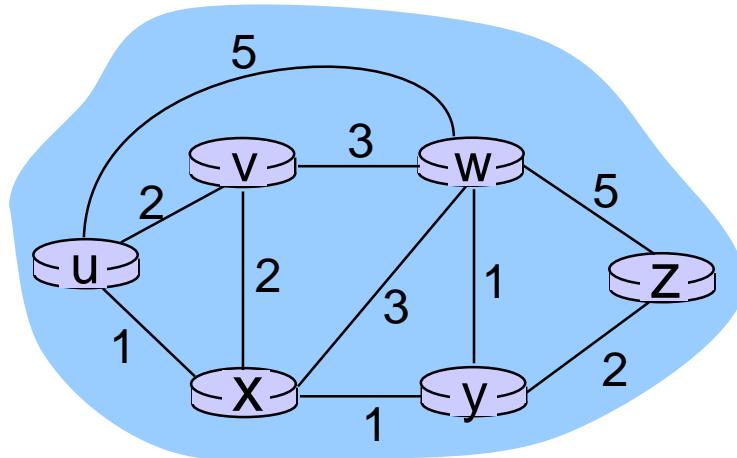
E = set of links = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

$c(x,y)$ = cost of link (x,y)
e.g., $c(w,z) = 5$

Therefore,

cost of path $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

Graph abstraction: costs



graph: $G = (V, E)$

V = set of node/routers = { u, v, w, x, y, z }

E = set of links = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

$c(x,y)$ = cost of link (x,y)
e.g., $c(w,z) = 5$

Therefore,

cost of path $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

key question: what is the least-cost path between any two nodes, e.g. u and z ?
routing algorithm: algorithm that finds that least cost path

Routing algorithm classification

Q: global or decentralized information?

global:

- all routers have complete topology, link cost info
- “link state” algorithms

decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- “distance vector” algorithms

Routing algorithm classification

Q: global or decentralized information?

global:

- all routers have complete topology, link cost info
- “link state” algorithms

decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- “distance vector” algorithms

Q: static or dynamic?

static:

- routes change slowly over time

dynamic:

- routes change more quickly
 - periodic update
 - in response to link cost changes

Topics

- Introduction
 - Types of routing algorithms
- Routing algorithms
 - Link state
 - Distance vector
- Scalable routing

A link-state routing algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
 - accomplished via “link state broadcast”
 - all nodes have same info
- computes least cost paths from one node (‘source’) to all other nodes
 - gives *forwarding table* for that node
- iterative: after k iterations, know least cost path to k destinations

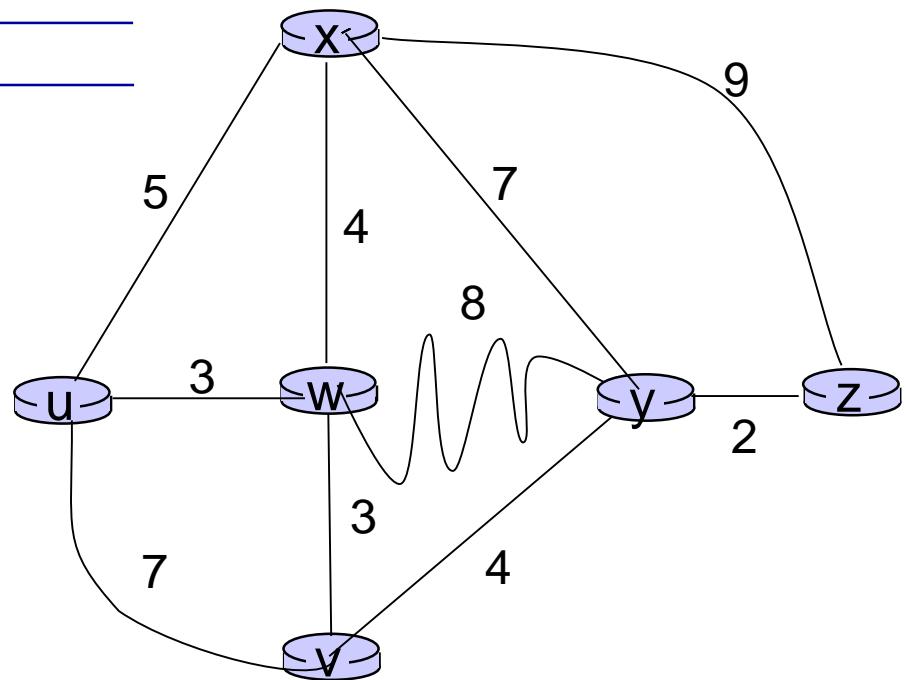
A link-state routing algorithm

notation:

- $c(x,y)$: link cost from node x to y ; $= \infty$ if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. v
- $p(v)$: predecessor node along path from source to v
- N' : set of nodes whose least cost path definitively known

Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0						
1						
2						
3						
4						
5						



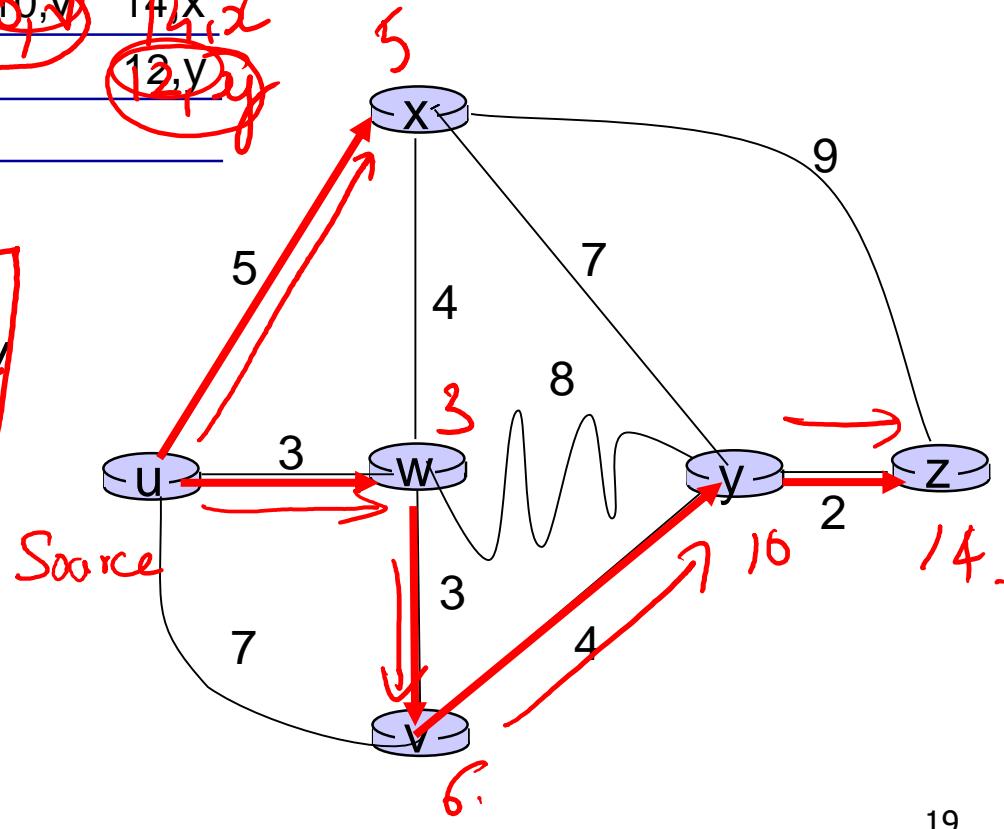
Dijkstra's algorithm: example

Step	N'	$D(v)$	$D(w)$	$D(x)$	$D(y)$	$D(z)$
		$p(v)$	$p(w)$	$p(x)$	$p(y)$	$p(z)$
0	u	7, u	3, u	5, u	∞	∞
1	uw	6, w	5, u	11, w	∞	∞
2	uwx	6, w	11, w	14, x	∞	∞
3	UWXV		10, v	14, x	∞	∞
4	UWXVY			12, y	∞	∞
5	UWXVYZ					

forwarding table for 'u'

- ❖ construct shortest path tree by tracing predecessor nodes
- ❖ ties can exist (can be broken arbitrarily)

x	(u, w)
w	(u, x)
y	(u, w)
z	(u, w)



Dijkstra's algorithm

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 $D(v) = \infty$

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' :

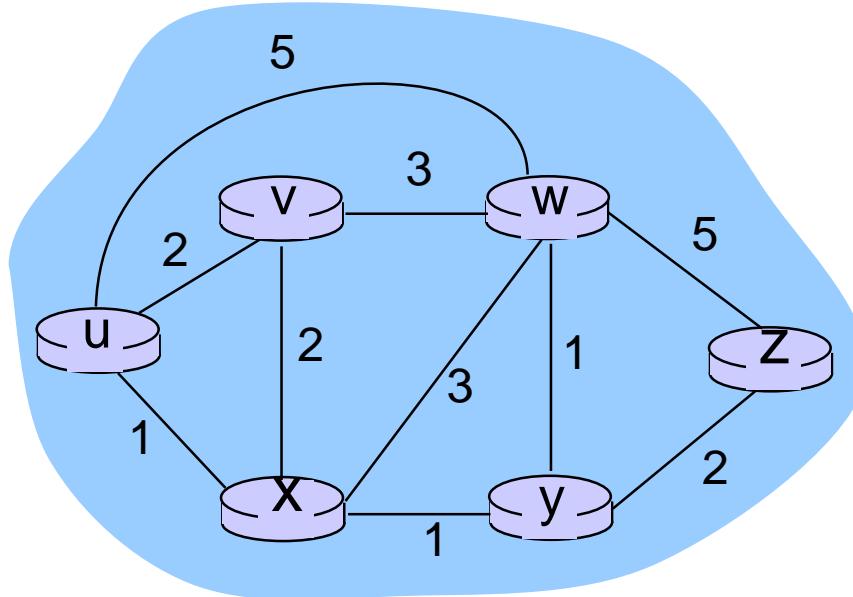
12 **$D(v) = \min(D(v), D(w) + c(w,v))$**

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'**

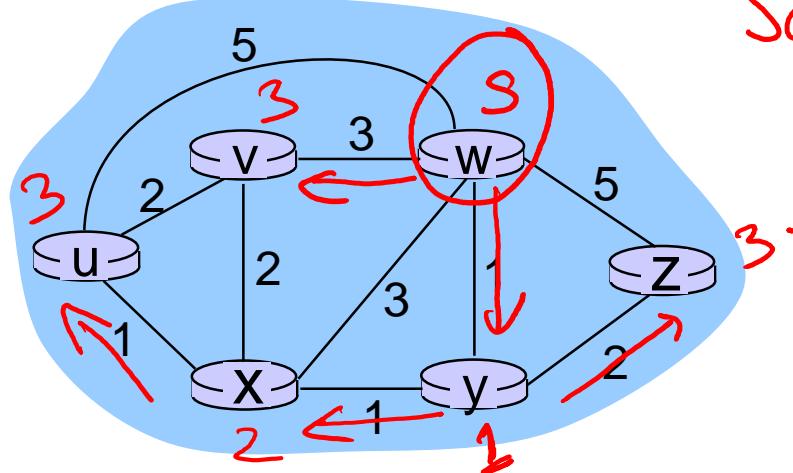
Dijkstra's algorithm: another example



Dijkstra's algorithm: another example

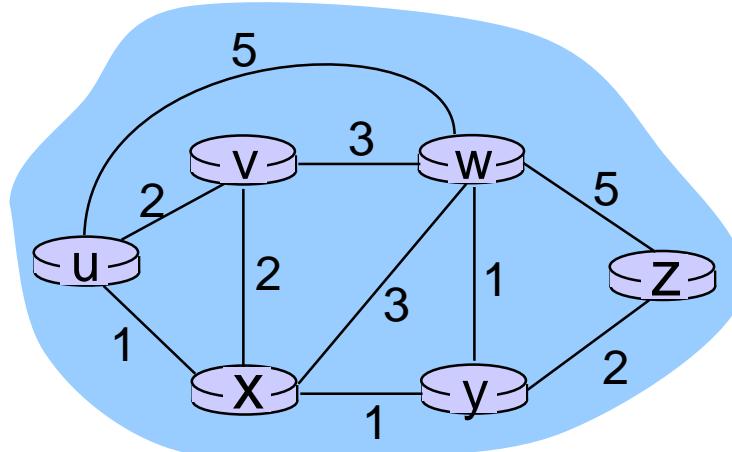
Step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	w	3, w	3, w	3, w	1, w	5, w
1	w y	3, w	5, w	2, y		3, y
2	w y x	3, w	3, x			3, y
3	w y x v	3, w				3, y
4	w y x v u					3, y
5	w y x v u z					3, y

Dest	outgoing link	
	u	v
u	(w, y)	
v	(w, v)	
x	(w, y)	
y	(w, y)	
z	(w, y)	



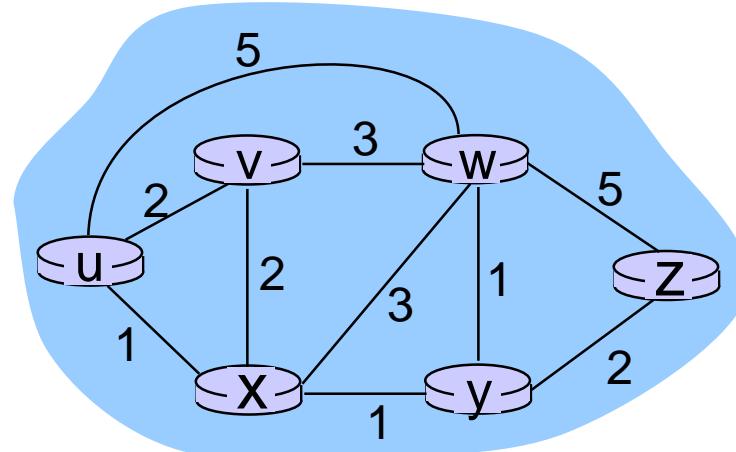
Dijkstra's algorithm: another example

Step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



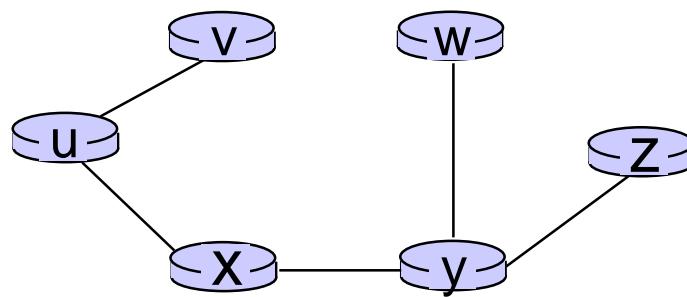
Dijkstra's algorithm: another example

Step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstra's algorithm: example (2)

resulting shortest-path tree from u:



resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Topics

- Introduction
 - Types of routing algorithms
- Routing algorithms
 - Link state
 - Distance vector
- Scalable routing

Distance vector algorithm

- A node x :
 - knows cost to each neighbor v : $c(x,v)$
 - Stores and update Distance Vector (DV)
 - $D_x(y) = \text{estimate of least cost from } x \text{ to } y$
 - Node x maintains distance vector $\mathbf{D}_x = [D_x(y) : y \in N]$

Distance vector algorithm

Key idea:

- From time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

- ❖ Under minor, natural conditions, the estimate $D_x(y)$ converge to the actual least cost $d_x(y)$

Distance vector algorithm

iterative, asynchronous:

each local iteration
caused by:

- local link cost change
- DV update message from neighbor

distributed:

- each node notifies neighbors *only* when its DV changes
 - neighbors then notify their neighbors if necessary

each node:

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

recompute estimates

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

Distance Vector algorithm

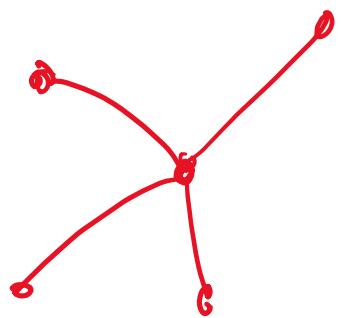
Let

$d_x(y) := \text{cost of least-cost path from } x \text{ to } y$

then

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

v
| | |
| | |
cost from neighbor v to destination y
cost to neighbor v
 \min taken over all neighbors v of x



Star

diameter = 2

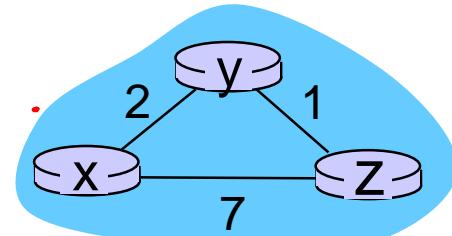


Line
diameter = 4.

		T0		
		X	Y	Z
f	X	0	2	7
y	X	2	1	1
m	Z	2	2	1

		X	Y	Z
		0 ^(x)	2 ^(y)	3 ^(y)
x*	y	20	1	
z	2	7	1	0

✓ 2 7
 ✓ 4 2 3
 (via y)
 14 8 7
 (via z)



		X	Y	Z
		1	1	1
x	X	1	1	1
y*	Y	2	0	1
z	Z	2	2	1

		X	Y	Z
		1	1	1
x	X	1	1	1
y	Y	2	1	1
z*	Z	7	1	0

		X	Y	Z
		0	2	7
f	X	*		
y	X	2	2	2
m	Z	2	2	2

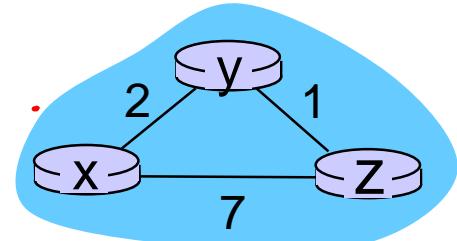
		X	Y	Z
		0 ^(x)	2 ^(y)	3 ^(y)
X*	X	20	1	
Y	X			
Z	X	7	1	0

		X	Y	Z
		0	2	7
X	X	2	0	1
Y	X	2	0	1
Z	X	2	2	2

		X	Y	Z
		0	2	7
X	X	2 ^(x)	0 ^(y)	1 ^(z)
Y	X			
Z	X	7	1	0

		X	Y	Z
		0	2	7
X	X	2	2	2
Y	X	2	2	2
Z	X	7	1	0

2 4 9 (via x)
 2 0 1
 8 2 1 (via z).
 min: 2 0 1

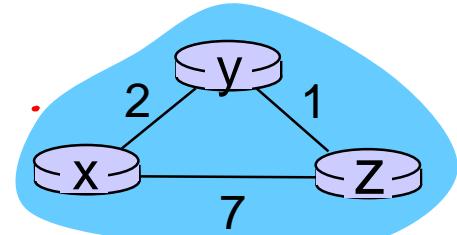


T_0

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

T_0

	x	y	z
x^*	0 ^(x)	2 ^(y)	3 ^(z)
y	20	1	
z	7	1	0



	x	y	z
x	2	2	2
y^*	2	0	1
z	2	2	2

	x	y	z
x	0	2	7
y	2 ^(x)	0 ^(y)	1 ^(z)
z	7	1	0

	x	y	z
x	2	2	2
y	2	2	2
z^*	7	1	0

	x	y	z
x	0	2	7
y	2 ^(y)	1 ^(y)	0 ^(z)
z	3 ^(y)	1 ^(y)	0 ^(z)

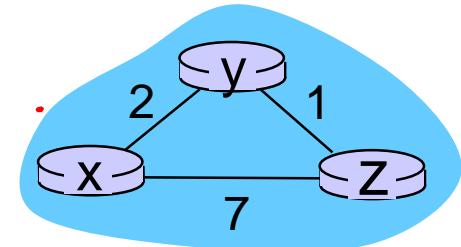
$$\begin{array}{r}
 7 \quad 9 \quad 14 \quad (\text{via } x) \\
 3 \quad 1 \quad 2 \quad (\text{via } y) \\
 7 \quad 1 \quad 0 \\
 \hline
 \min : \quad 3 \quad 1 \quad 0
 \end{array}$$

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7	1	0

	X	Y	Z
X	0	2	7
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7	1	0

	X	Y	Z
X	0	2	7
Y	2 ^(y)	0 ^(z)	1 ^(x)
Z	3 ^(y)	1 ^(z)	0 ^(x)

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	3 ^(y)	1 ^(z)	0 ^(x)



0 2 3
 4 2 3 (way)
 6 4 3 (v₁ < 3)
 min: 0 2 3

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7 ^(x)	1 ^(y)	0 ^(z)
1			

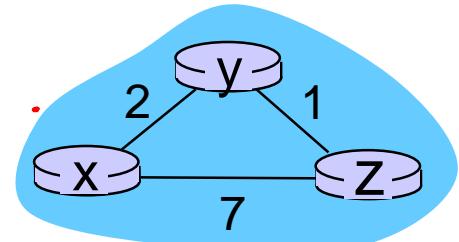
	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	3 ^(x)	1 ^(y)	0 ^(z)

	X	Y	Z
X	0	2	7
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7	1	0

	X	Y	Z
X	0	2	7
Y	2 ^(y)	0 ^(z)	1 ^(x)
Z	3 ^(y)	1 ^(z)	0 ^(x)

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	3 ^(y)	1 ^(z)	0 ^(x)

2 4 5 (via x)
 2 0 1.
 4 2 1 (via z)
 min 2 0 1



	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7	1	0
	1		

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	3 ^(y)	1 ^(y)	0 ^(z)
	1		

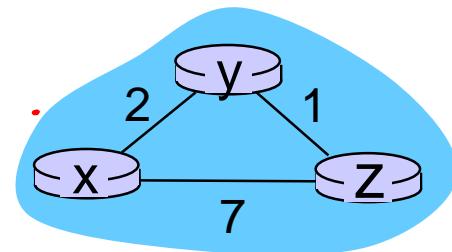
	X	Y	Z
X	0	2	7
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	7	1	0
	1		

	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2 ^(x)	0 ^(y)	1 ^(z)
Z	3 ^(y)	1 ^(y)	0 ^(z)
	1		

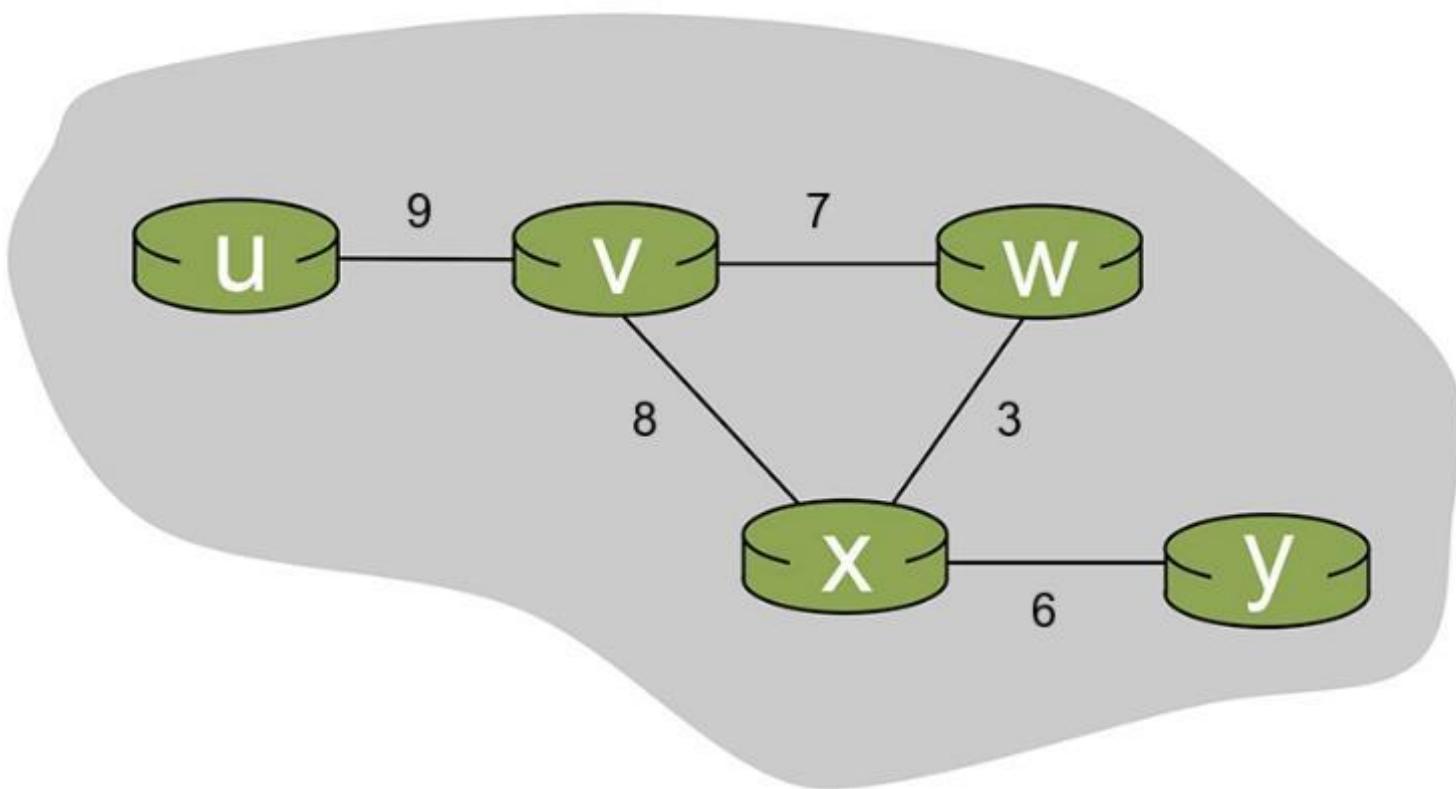
	X	Y	Z
X	0	2	7
Y	2	0 ^(y)	1 ^(z)
Z	3 ^(y)	1 ^(y)	0 ^(z)
	1		

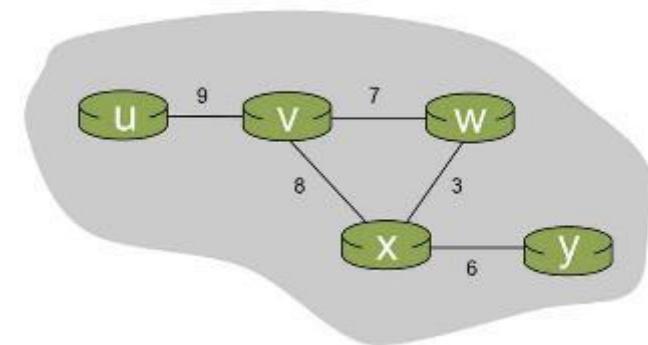
	X	Y	Z
X	0 ^(x)	2 ^(y)	3 ^(z)
Y	2	6	1
Z	3	1	0
	1		

3 5 6 (via x)
 3 1 2 (via y)
 3 1 0
 min: 3 1 0



DVR Example 2





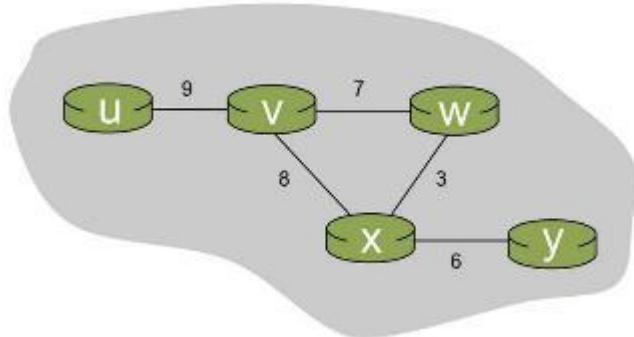
U	U	V	W	X	Y
<i>U</i>	0	9	∞	∞	∞
<i>V</i>	∞	∞	∞	∞	∞

V	U	V	W	X	Y
<i>U</i>	∞	∞	∞	∞	∞
<i>V</i>	9	0	7	8	∞
<i>W</i>	∞	∞	∞	∞	∞
<i>X</i>	∞	∞	∞	∞	∞

W	U	V	W	X	Y
<i>V</i>	∞	∞	∞	∞	∞
<i>W</i>	∞	7	0	3	∞
<i>X</i>	∞	∞	∞	∞	∞

X	U	V	W	X	Y
<i>V</i>	∞	∞	∞	∞	∞
<i>W</i>	∞	∞	∞	∞	∞
<i>X</i>	∞	8	3	0	6
<i>Y</i>	∞	∞	∞	∞	∞

Y	U	V	W	X	Y
<i>X</i>	∞	∞	∞	∞	∞
<i>Y</i>	∞	∞	∞	6	0



* DV has changed and need to be sent to neighbors

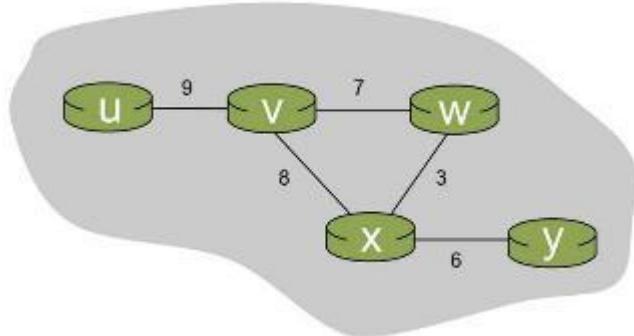
U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

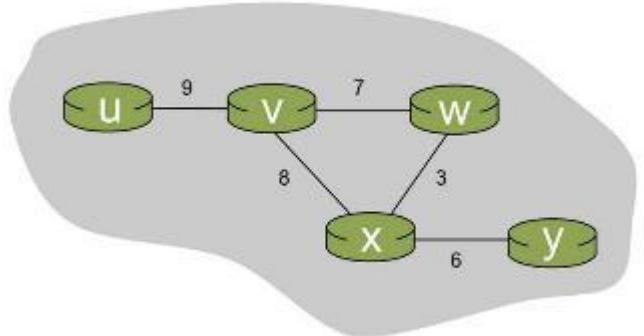
Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U	0	9	∞	∞	∞
V	9	0	7	8	∞

Update DV

0 9 ~~16~~ ~~17~~ ~~18~~
 18 9 16 17 ~~18~~
 min 0 9 16 17 ~~18~~



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

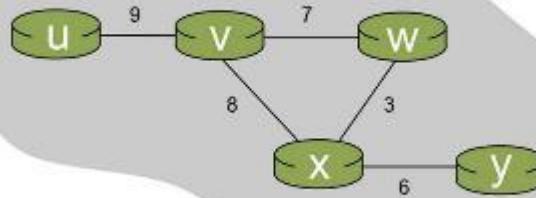
Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

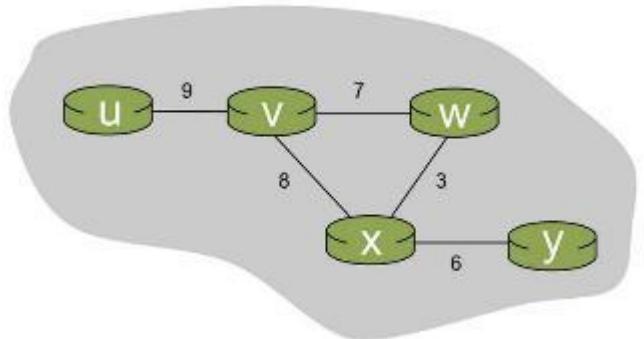
Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Update DV

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V	9	0	7	8	∞
W	∞	7	0	3	∞
X	∞	8	3	0	6

9 18 ↙ ↙ ↙ ↙ (via 2)
 9 0 ↗ ↗ ↗
 ↙ 14 ↗ 10 ↙ ↙ (via 0)
 ↙ 16 ↗ 8 14 ↙ (via X)
 min 9 0 7 8 14



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

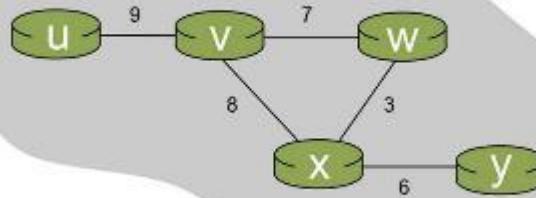
Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

DV Updated!



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

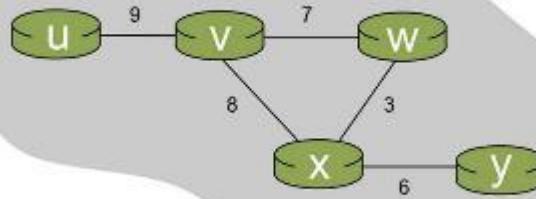
U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

W	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X	∞	8	3	0	6

Update DV

16 7 14 15 b. (6 via)
5 7 0 3 5
2 11 6 3 9 (via x)
min: 16 7 0 3 9.



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

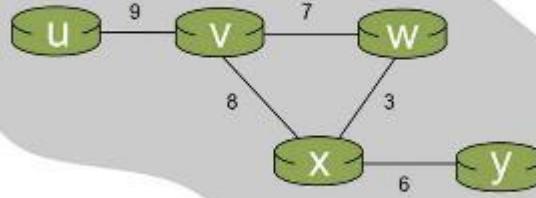
Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

DV Updated!



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

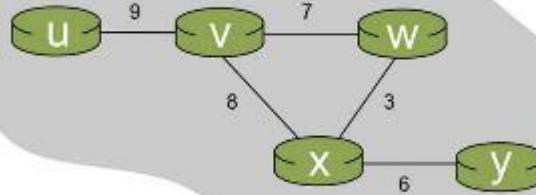
W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	6	0

Update DV

17	8	15	16	4	(via V)
5	10	3	6	2	(via W)
2	8	3	0	1	(via Y)
2	4	12	6	0	(via Y)
min	17	8	3	0	6



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

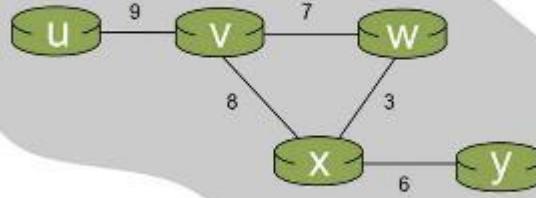
U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

DV Updated!

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

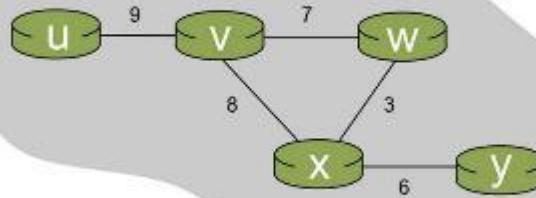
U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

Update DV

$\begin{matrix} 6 & 14 & 9 & 6 & 12 \end{matrix}$ (via X)
 $\begin{matrix} 6 & 6 & 4 & 1 & 6 \end{matrix}$
 $\min \begin{matrix} 6 & 14 & 9 & 6 & 0 \end{matrix}$



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	14	9	6	0

Send DV to Neighbor X

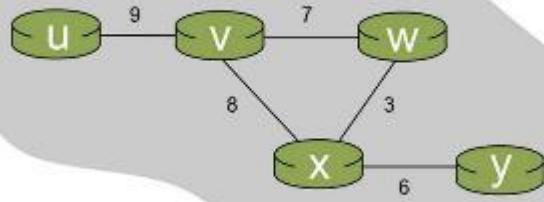
U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

DV Updated!

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

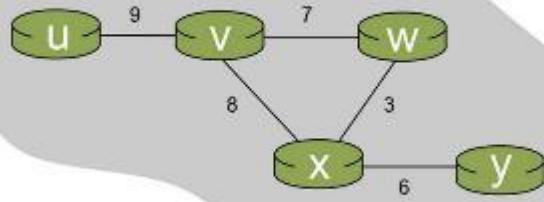
Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	14	9	6	0

Send DV to Neighbor X

u 0 9 16 17 5.
v 9 0 7 8 14.

0 9 16 17 4.
18 9 16 17 23
min 0 9 16 17 23



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

Send DV to Neighbor V, X

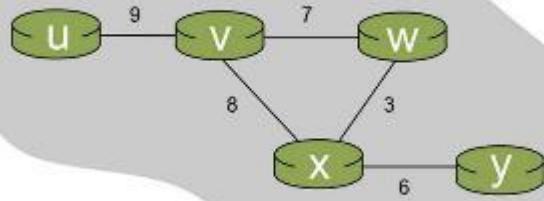
X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Y
U
W
X

9 18 25 26 2.
9 0 7 8 14 ←
23 14 7 10 15:
25 16 11 8 14
min 9 0 7 8 14.



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	7	0	3	∞
X	∞	8	3	0	6

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	8	3	0	6

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

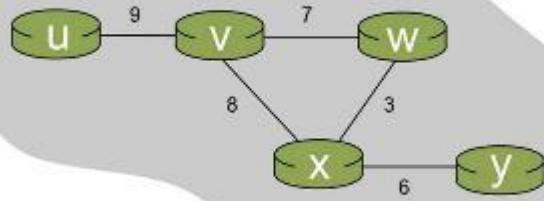
V	U	V	W	X	Y
U	0	9	16	17	∞
V*	9	0	7	8	14
W	16	7	0	3	9
X	17	8	3	0	6

~~U V W X Y~~
~~V 9 0 7 8 14~~
~~W 16 7 0 3 9~~
~~X 17 8 3 0 6~~

~~U V W X Y~~
~~16 7 0 3 9~~
~~20 11 6 3 9~~
~~15 7 0 3 9~~

(via V)
 (via X).

16 7 0 3 9



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

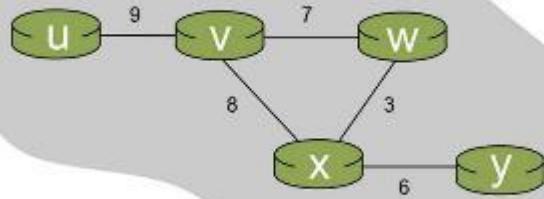
U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

V	U	V	W	X	Y
U	0	9	16	17	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	17	8	3	0	6

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	12	14	9	6	0

17	8	15	16	29
19	10	3	6	12
17	8	3	0	6
12	20	15	12	6
17	8	3	0	6



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

V	U	V	W	X	Y
U	0	9	16	17	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	17	8	3	0	6
Y	∞	14	9	6	0

Send DV to Neighbor V, X

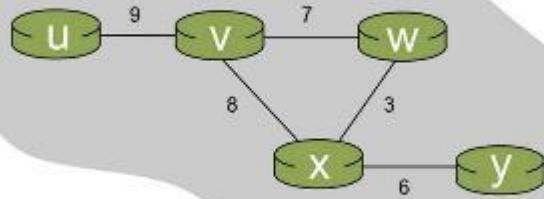
X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	14	9	6	0

Y	U	V	W	X	Y
X	17	8	3	6	6
Y *	14	9	6	0	0

23 24 9 6 12

6 14 9 6 0

min 23 14 9 6 0.



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

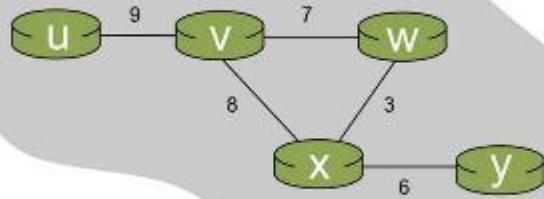
U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	17	8	3	0	6

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	14	9	6	0

Y	U	V	W	X	Y
X	17	8	3	0	6
Y*	23	14	9	6	0



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	17	8	3	0	6

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

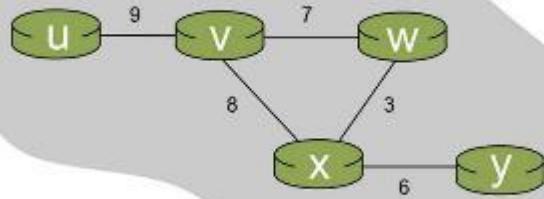
Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	17	14	9	6	0
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	17	14	9	6	0
X*	∞	∞	∞	∞	∞

Send DV to Neighbor X



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	17	8	3	0	6

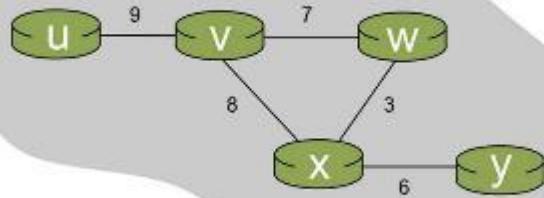
Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	14	9	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	17	8	3	0	6
Y*	23	14	9	6	0

Send DV to Neighbor X



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	8	3	0	6

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	8	3	0	6
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	∞	7	0	3	9
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor X

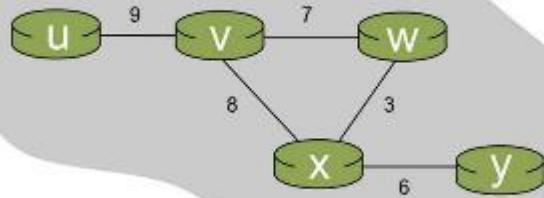
U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

V	U	V	W	X	Y
U	0	9	∞	∞	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	16	17	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	14	9	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	16	17	23
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

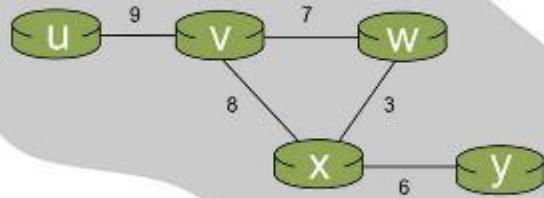
W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor X

Y	U	V	W	X	Y
X	17	8	3	0	6
Y*	23	14	9	6	0



* DV has changed and need to be sent to neighbors

U	U	V	W	X	Y
U*	0	9	∞	∞	∞
V	∞	∞	∞	∞	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	∞	∞	∞	∞	∞
V*	9	0	7	8	∞
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W*	∞	7	0	3	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	∞	∞	∞	∞	∞
W	∞	∞	∞	∞	∞
X*	∞	8	3	0	6
Y	∞	∞	∞	∞	∞

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	∞
V	9	0	7	8	∞

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	16	17	∞
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	∞
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	∞
W	∞	7	0	3	∞
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	∞	∞	∞	∞	∞
Y*	∞	∞	∞	6	0

Send DV to Neighbor X

U	U	V	W	X	Y
U*	0	9	16	17	23
V	9	0	7	8	14

Send DV to Neighbor V

V	U	V	W	X	Y
U	0	9	16	17	23
V*	9	0	7	8	14
W	∞	∞	∞	∞	∞
X	∞	∞	∞	∞	∞

Send DV to Neighbor U,W,X

W	U	V	W	X	Y
V	9	0	7	8	14
W*	16	7	0	3	9
X	∞	∞	∞	∞	∞

Send DV to Neighbor V, X

X	U	V	W	X	Y
V	9	0	7	8	14
W	16	7	0	3	9
X*	17	8	3	0	6
Y	∞	∞	∞	6	0

Send DV to Neighbor V, W, Y

Y	U	V	W	X	Y
X	17	8	3	0	6
Y*	23	14	9	6	0

Send DV to Neighbor X

Comparison of LS and DV algorithms

Communication costs

- **LS:** broadcast link info to all
- **DV:** exchange between neighbors only
 - convergence time varies

robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect *link* cost
- each node computes only its own table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
- error propagate thru network

Topics

- Introduction
 - Types of routing algorithms
- Routing algorithms
 - Link state
 - Distance vector
- Scalable routing

Making routing scalable

our routing study thus far - idealized

- all routers identical
 - network “flat”
- ... *not true in practice*

scale: with billions of destinations:

- Can't store all destinations in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

Internet approach to scalable routing

Aggregate routers into regions known as “autonomous systems” (AS) (a.k.a. “domains”)

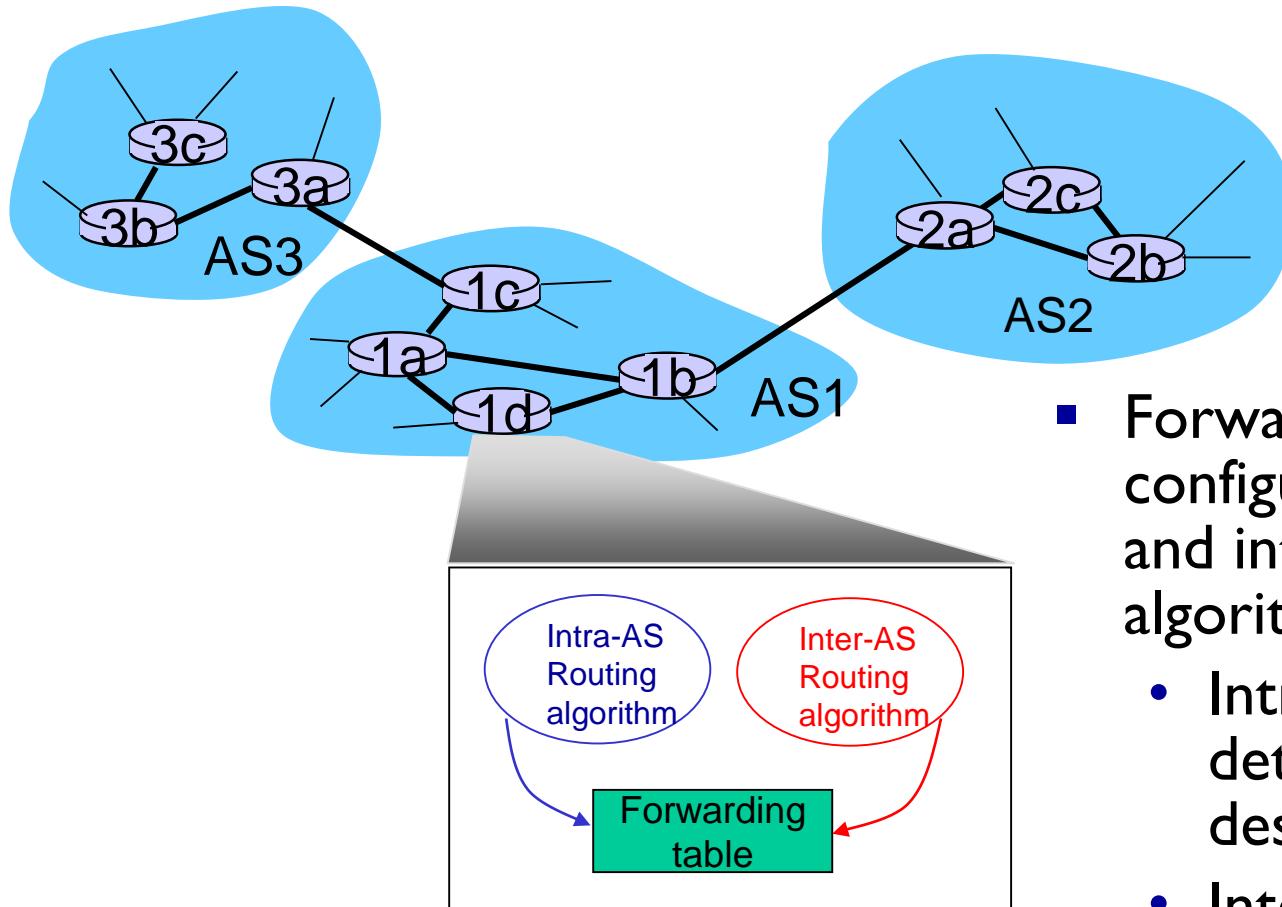
intra-AS routing

- routing among hosts, routers in same AS (“network”)
- all routers in AS must run *same* intra-domain protocol
- routers in *different* AS can run *different* intra-domain routing protocol
- gateway router: at “edge” of its own AS, has link(s) to router(s) in other AS’es

inter-AS routing

- routing among AS’es
- gateways perform inter-domain routing (as well as intra-domain routing)

Interconnected ASes



- Forwarding table configured by both intra- and inter-AS routing algorithm
 - Intra-AS routing determine entries for destinations within AS
 - Inter-AS & intra-AS determine entries for external destinations

Intra-AS Routing

- Also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First (IS-IS protocol essentially same as OSPF)
 - IGRP: Interior Gateway Routing Protocol
(Cisco proprietary for decades, until 2016)

Inter-AS tasks

- suppose router in AS1 receives datagram destined outside of AS1:
 - router should forward packet to gateway router, but which one?

AS1 must:

1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1

job of inter-AS routing!

