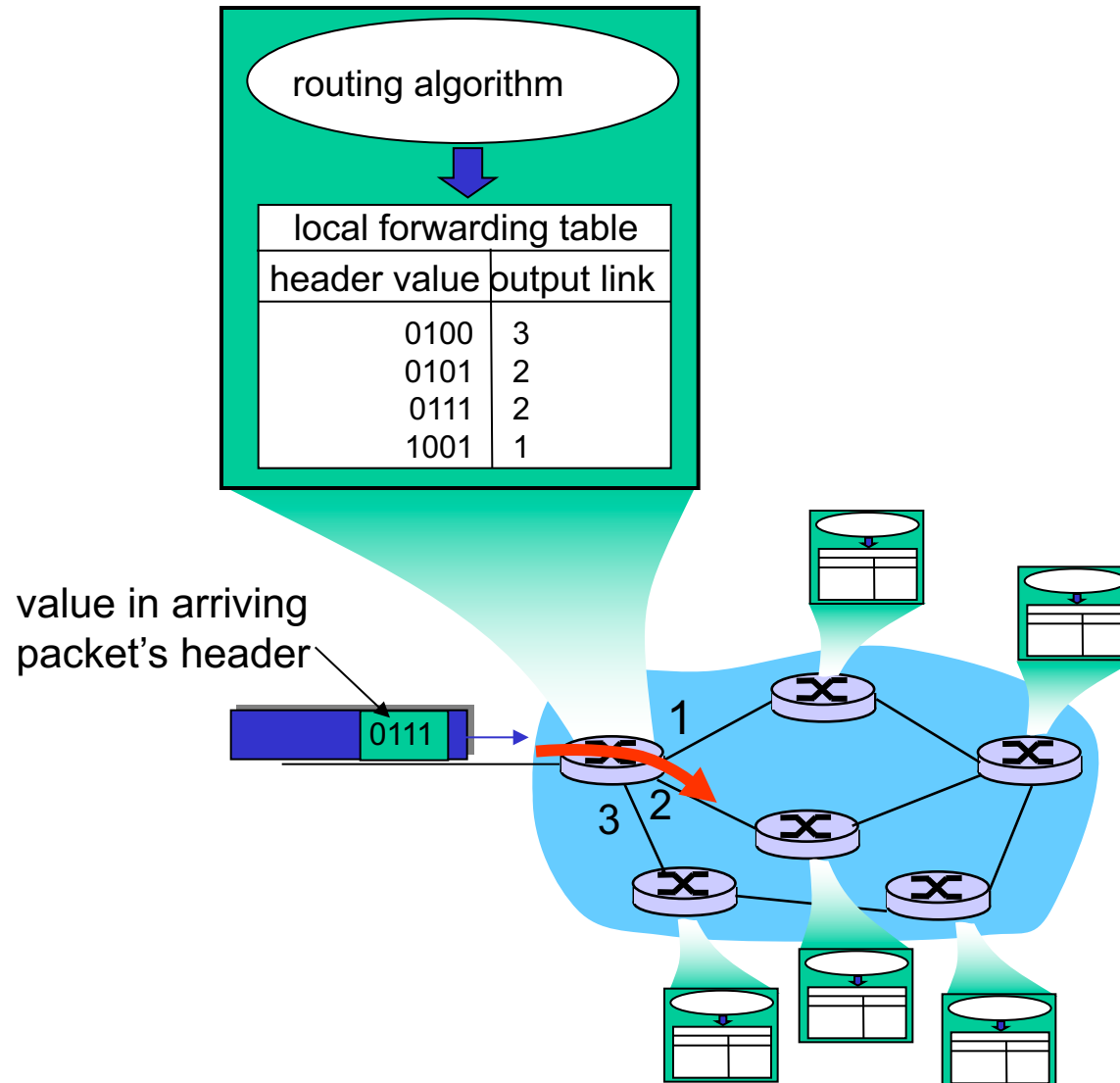


Internet Routing

Routing Basics

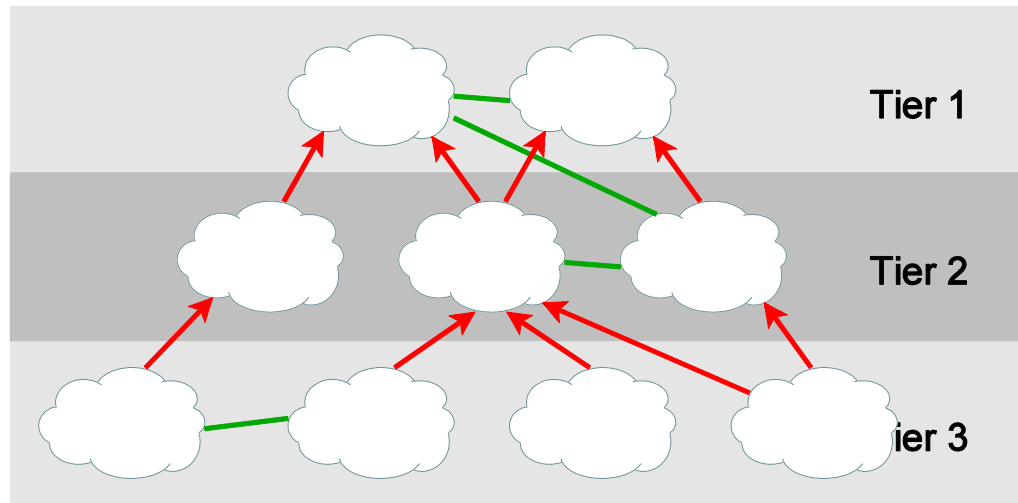
Interplay Between Routing, Forwarding



Routing Hierarchy

Why?

- ❑ Different networks
- ❑ Different administration
- ❑ Different goals



Routing Hierarchy (cont.)

- Key observation: Need less information with increasing distance to destination

- Two radically different approaches for routing
 - The landmark hierarchy
 - Usually the way you would navigate on the street network

 - The area hierarchy
 - Chosen in the Internet

Internet's Area Hierarchy

What is an **Autonomous System (AS)**?

- A set of routers under a single technical administration, using an *interior gateway protocol (IGP)* and common metrics to route packets within the AS and using an *exterior gateway protocol (EGP)* to route packets to other AS's
 - Sometimes AS's use multiple IGPs and metrics, but appear as single AS's to other AS's
-
- Each AS assigned unique ID
 - AS's peer at network exchanges

Routing Tasks

- ❑ Neighbor
 - Discovery
 - Maintenance
- ❑ Database
 - Granularity
 - Maintenance – updates
 - Synchronization
- ❑ Routing table
 - Metric
 - Calculation
 - Update

Routing Algorithm Classification

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

Static or dynamic?

Static:

- ❑ Routes change slowly over time

Dynamic:

- ❑ Routes change more quickly
 - Periodic update
 - In response to link cost changes

A Distance Vector Routing Algorithm

Decentralized algorithm:

- ❑ Router knows its neighbors and link costs to neighbors
- ❑ Iterative computation, exchange of info with neighbors

Bellman-Ford Equation (dynamic programming)

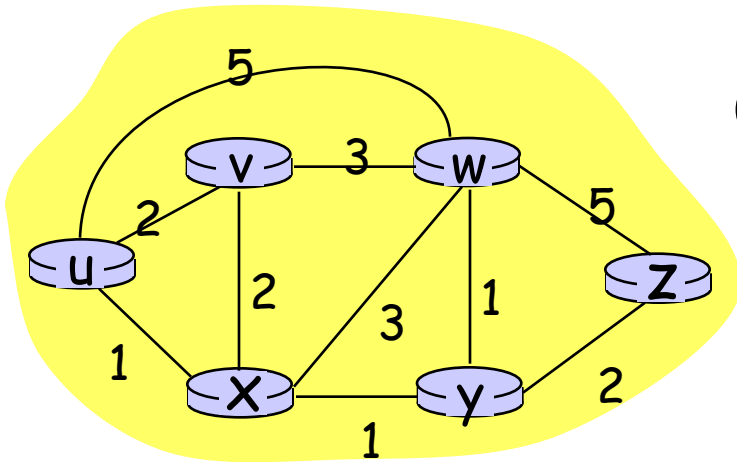
Define $d_x(y) :=$ cost of least-cost path from x to y

Then

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

where min is taken over all neighbors v of x

Bellman-Ford Example



Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Bellman-Ford equation says:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

Node that yields minimum is next
hop in shortest path → forwarding table

Distance Vector Algorithm

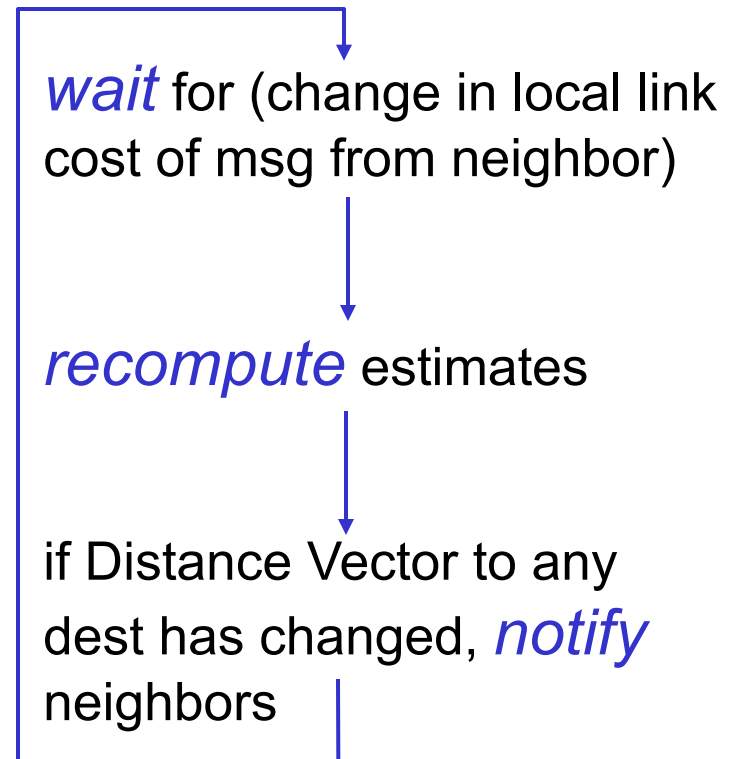
Iterative, asynchronous:

- ❑ Each local iteration caused by:
 - Local link cost change
 - DV update message from neighbor

Distributed:

- ❑ Each node notifies neighbors *only* when its Distance Vector changes
 - Neighbors then notify their neighbors if necessary

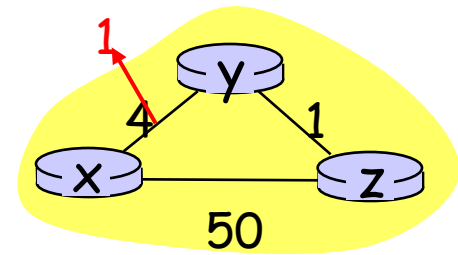
Each node:



Distance Vector (DV): Link Cost Changes

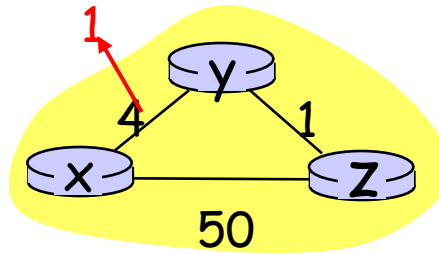
Link cost changes:

- ❑ Node detects local link cost change
- ❑ Updates routing info, recalculates distance vector
- ❑ If DV changes, notify neighbors



“good news travels fast”

- At time t_0 , y detects link-cost change, updates its DV, and informs its neighbors.
- At time t_1 , z receives update from y and updates its table, computes a new least cost to x and sends its neighbors its DV
- At time t_2 , y receives z 's update and updates its distance table. As y 's least costs do not change y does not send updates to z .



node y table

		cost to		
		x	y	z
from	x			
	y	4 1	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	2	1	0

node z table

		cost to		
		x	y	z
from	x			
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	5 2	1	0

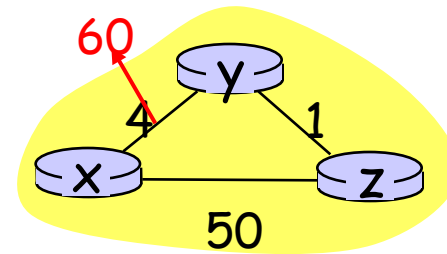
		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	2	1	0

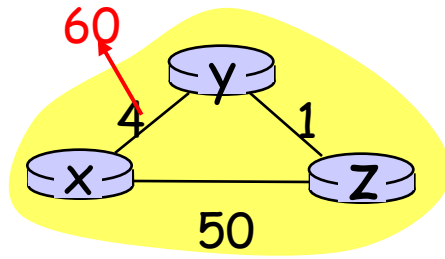
▶ time

Distance Vector: Link Cost Changes

Link cost changes:

- ❑ Good news travels fast
- ❑ Bad news travels slow





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

$$= \min\{60 + 0, 1 + 5\} = 6$$

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

$$= \min\{60 + 0, 1 + 7\} = 8$$

node y table

		cost to		
		x	y	z
from	x	4 6	0	1
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6 8	0	1
	z	7	1	0

node z table

		cost to		
		x	y	z
from	x			
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	5 7	1	0

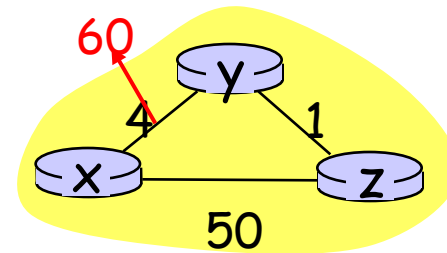
		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	7	1	0

time →

Distance Vector: Link Cost Changes

Link cost changes:

- ❑ Good news travels fast
- ❑ Bad news travels slow – “count to infinity” problem!
- ❑ E.g., 44 iterations before algorithm stabilizes



Poisoned reverse:

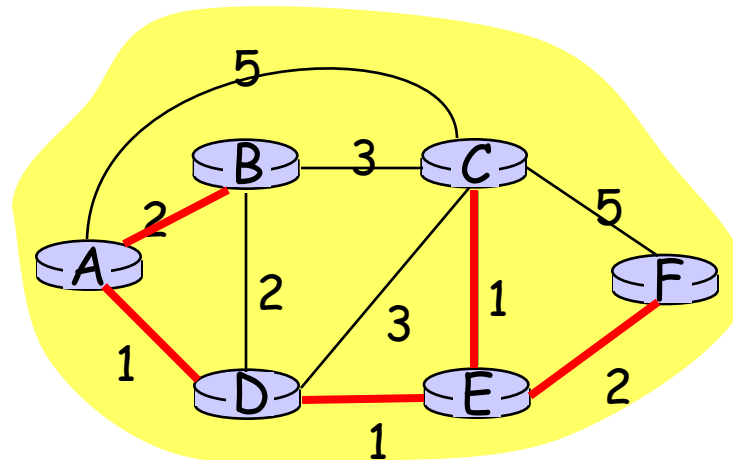
- ❑ If Z routes through Y to get to X:
 - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- ❑ Will this completely solve count to infinity problem?

A Link-State Routing 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 **routing table** for that node
- ❑ Example:
Dijkstra’s algorithm
 - Iterative: after k iterations, know least cost path to k dst.’s

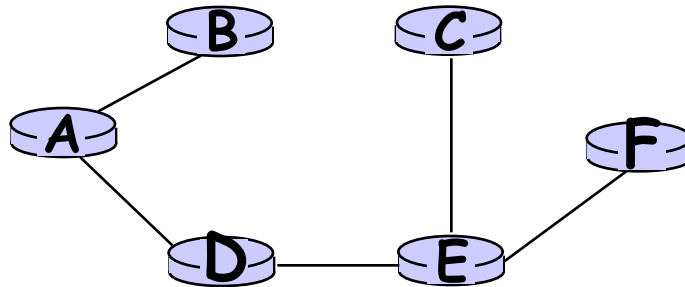
Dijkstra's Algorithm: Example

Step	start N'	$D(B),p(B)$	$D(C),p(C)$	$D(D),p(D)$	$D(E),p(E)$	$D(F),p(F)$
→ 0	A	2,A	5,A	1,A	infinity	infinity
→ 1	AD	2,A	4,D		2,D	infinity
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



Dijkstra's Algorithm: Example (2)

Resulting shortest-path tree from A:



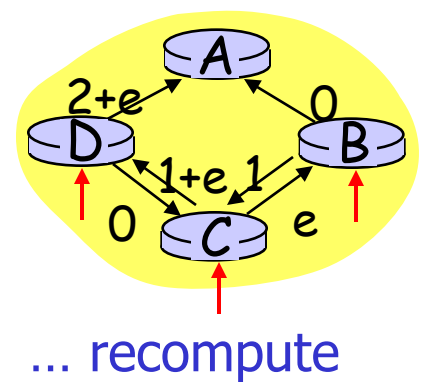
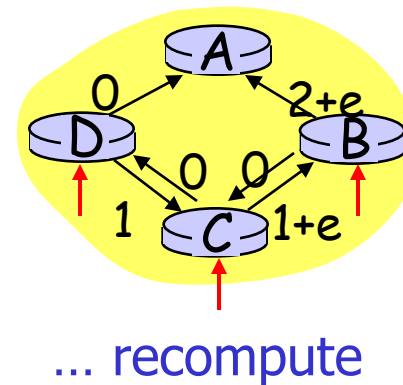
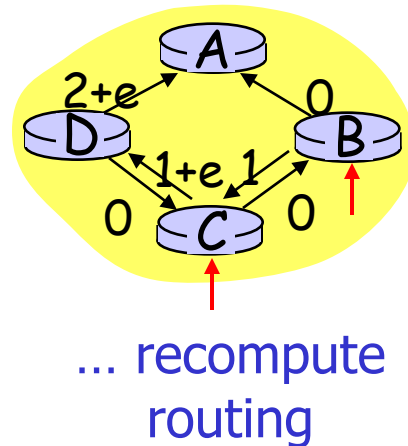
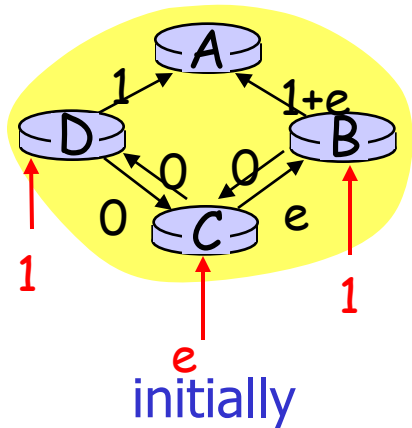
Resulting forwarding table at A:

destination	link
B	(A,B)
D	(A,D)
E	(A,D)
C	(A,D)
F	(A,D)

Dijkstra's Algorithm: Discussion

Oscillations possible:

- E.g., link cost = amount of carried traffic



Comparison of LS and DV Algorithms

Speed of Convergence

- LS: $O(n \log n)$ algorithm requires $O(nE)$ msgs
 - May have oscillations
- DV: Convergence time varies
 - May be routing loops
 - Count-to-infinity problem

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