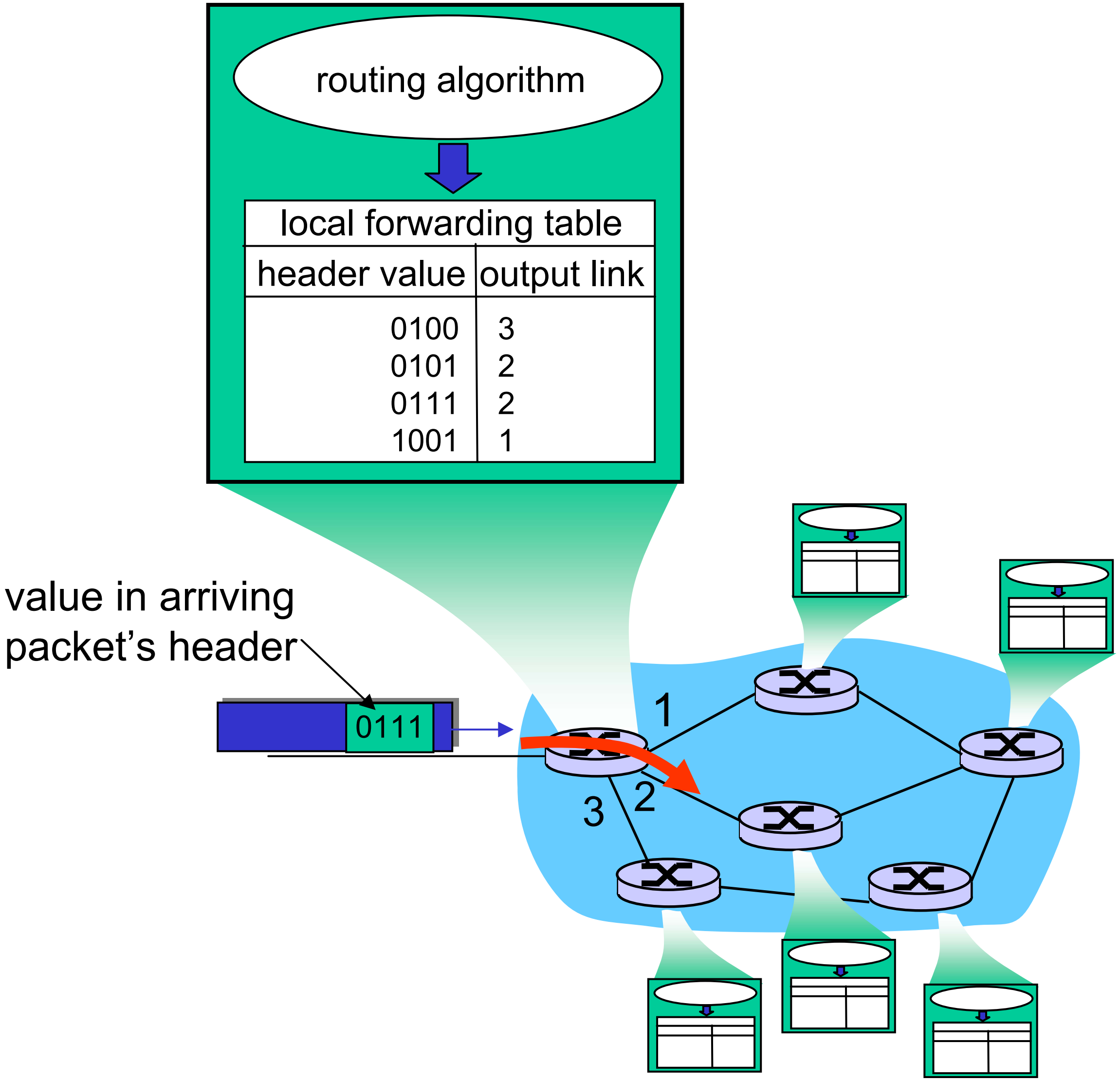
Chapter 4: Network Layer: Part II

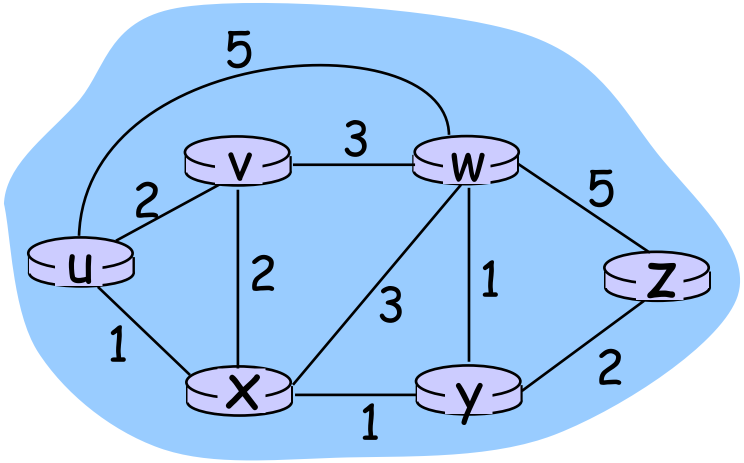
(last revision 19/04/05. v3)

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

Interplay between routing and forwarding



Graph abstraction



Graph: G = (N,E)

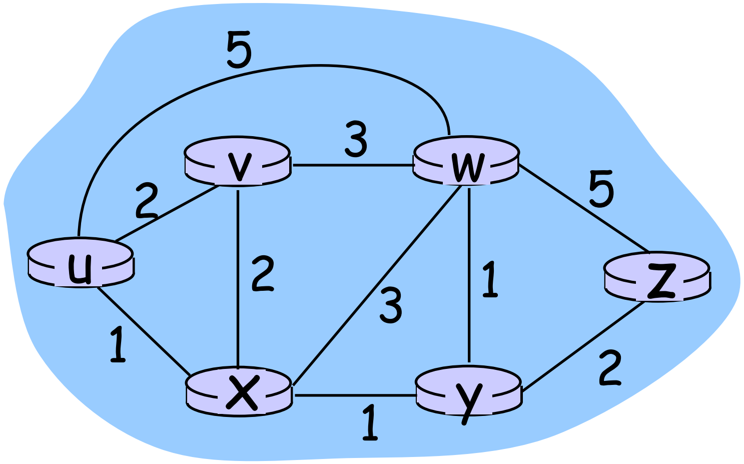
N = set of routers = { u, v, w, x, y, z }

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

Remark: Graph abstraction is useful in other network contexts

Example: P2P, where N is set of peers and E is set of TCP connections

Graph abstraction: costs



- $c(x,x')$ = cost of link (x,x')

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

- cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

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

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds "least-cost" path

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

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
    - Datagram format
    - IPv4 addressing
    - ICMP
    - IPv6
- 4.5 Routing algorithms
    - Link state
    - Distance Vector
    - Hierarchical routing
- 4.6 Routing in the Internet
    - RIP
    - OSPF
    - BGP
- 4.7 Broadcast and multicast routing (maybe)

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 dest.'s

Notation:

- c(x,y): link cost from node x to y; = ∞ 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(v): set of neighbors of v
- N': set of nodes whose least cost path definitively known

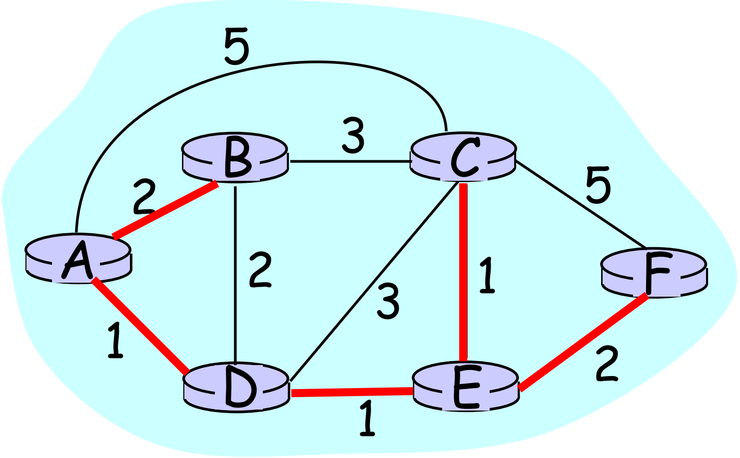
Dijsktra's Algorithm

```
Initialization:
  N' = {u}
  for all nodes v
    if $v \in N(u)$
        then $D(v) = c(u,v)$
    else $D(v) = \infty$

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all $v \in N(w)$ and not in N' :
      $D(v) = \min( D(v), D(w) + c(w,v) )$
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

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	∞	∞
1	AD	2,A	4,D		2,D	∞
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



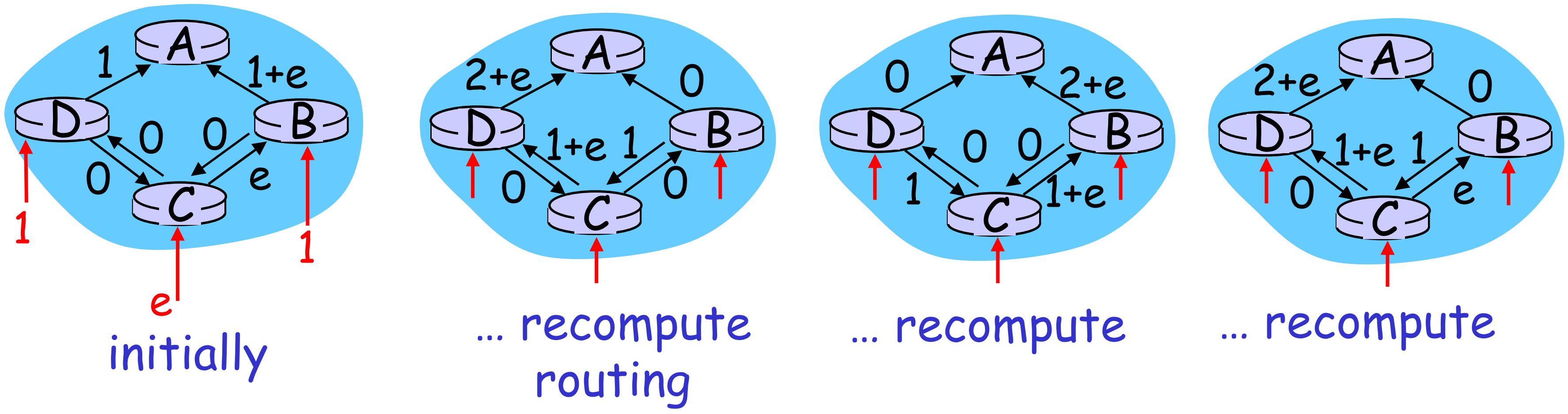
Dijkstra's algorithm, discussion

Algorithm complexity: n nodes, E links

- each iteration: need to check all nodes, w, not in N
- n(n+1)/2 comparisons: $O(n^2)$
- more efficient implementations possible: O(nlogn+E)

Oscillations possible:

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



Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Distance Vector Algorithm (1)

Bellman-Ford Equation (dynamic programming)

Define

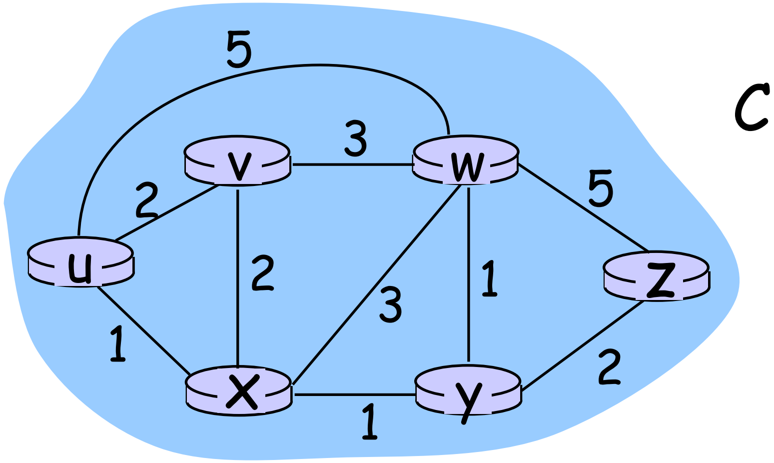
$d_x(y)$:= cost of least-cost path from x to y

Then

$$\boxed{d_x(y) = \min_{v \in N(x)}\{c(x,v) + d_v(y)\}}$$

where min is taken over all neighbors of x

Distance Vector Algorithm (2)



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

B-F equation says:

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

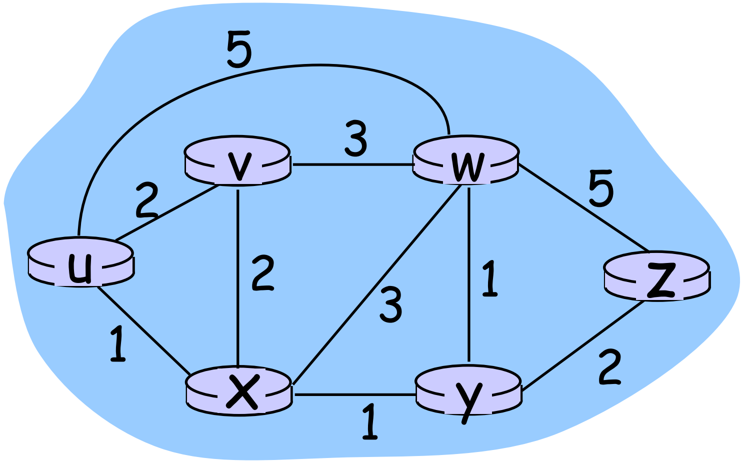
$$
\begin{aligned}
= \min \{ & 2 + 5, \\
& 1 + 3, \\
& 5 + 3\} = 4
\end{aligned}
$$

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

Distance Vector Algorithm (3)

- $D_x(y)$ = estimate of least cost from x to y
- Distance vector: $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x knows cost to each neighbor v: $c(x,v)$
- Node x maintains $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x also maintains its neighbors' distance vectors
  - For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \in N]$

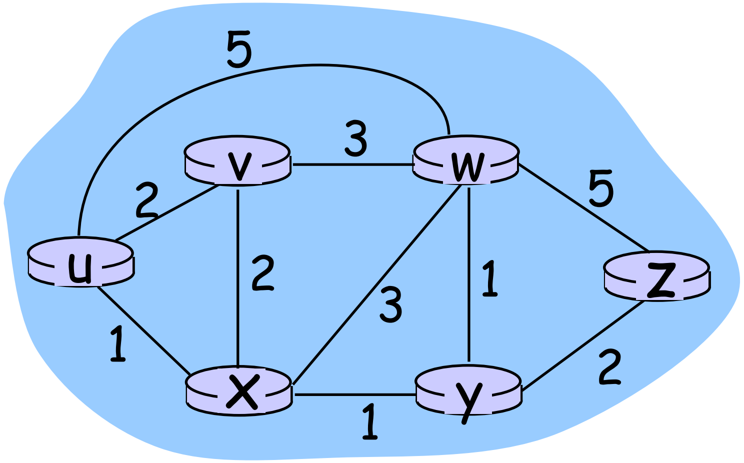
Bellman-Ford example (1)



Distance vectors stored at node x

		Cost to					
		x	y	z	u	v	w
from	**x**	**0**	**1**	**3**	**1**	**2**	**2**
	y	1	0	2	2	3	1
	u	1	2	4	0	2	5
	v	2	3	5	3	0	3
	w	2	1	3	5	3	0

Bellman-Ford example (2)



Routing table at node x

	destination				
	y	z	u	v	w
hop, cost	y,1	y,1	u,1	v,2	y,2

Cost to

from	x	y	z	u	v	w
x	**0**	**1**	**3**	**1**	**2**	**2**
y	1	0	2	2	3	1
u	1	2	4	0	2	5
v	2	3	5	3	0	3
w	2	1	3	5	3	0

Distance vector algorithm (4)

Basic idea:

- Each node periodically sends its own distance vector estimate to neighbors
- When node x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

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

- Under "natural" conditions, the estimate $D_x(y)$ converges to the actual least cost $d_x(y)$

Distance Vector Algorithm (5)

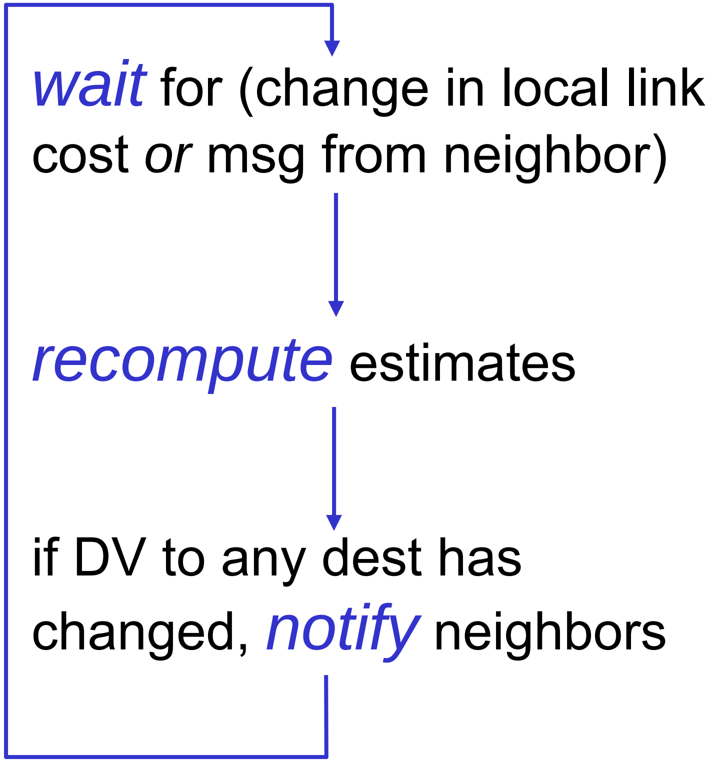
Iterative, asynchronous:
each local iteration caused by:

- 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

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2+1, 7+0\} = 3$$

node x table

cost to

from	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

cost to

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

cost to

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

node y table

cost to

from	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

cost to

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

cost to

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

node z table

cost to

from	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

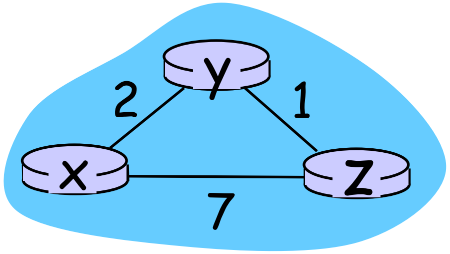
cost to

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

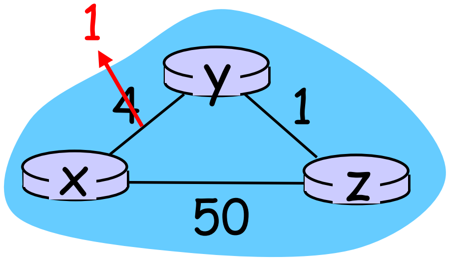
cost to

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

time



Distance Vector: 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 the link-cost change, updates its DV, and informs its neighbors.

At time t_1, *z* receives the update from *y* and updates its table. It 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. *y*'s least costs do not change and hence *y* does *not* send any message to *z*.

"good news travels fast"

node w table

cost to

from	x	y	z	w
w	1	5	1	0
x	0	4	2	1
z	2	6	0	1

node x table

cost to

from	x	y	z	w
x	0	4	2	1
w	1	5	1	0
y	4	0	6	5

node y table

cost to

from	x	y	z	w
y	4	0	6	5
x	0	4	2	1
z	2	6	0	1

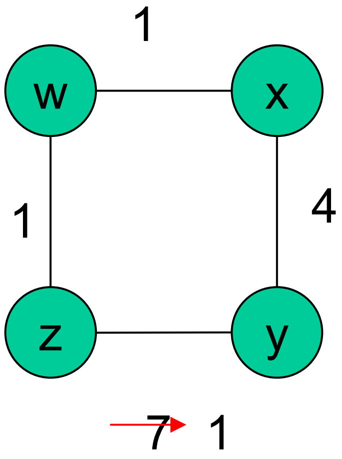
node z table

cost to

from	x	y	z	w
z	2	6	0	1
w	1	5	1	0
y	4	0	6	5

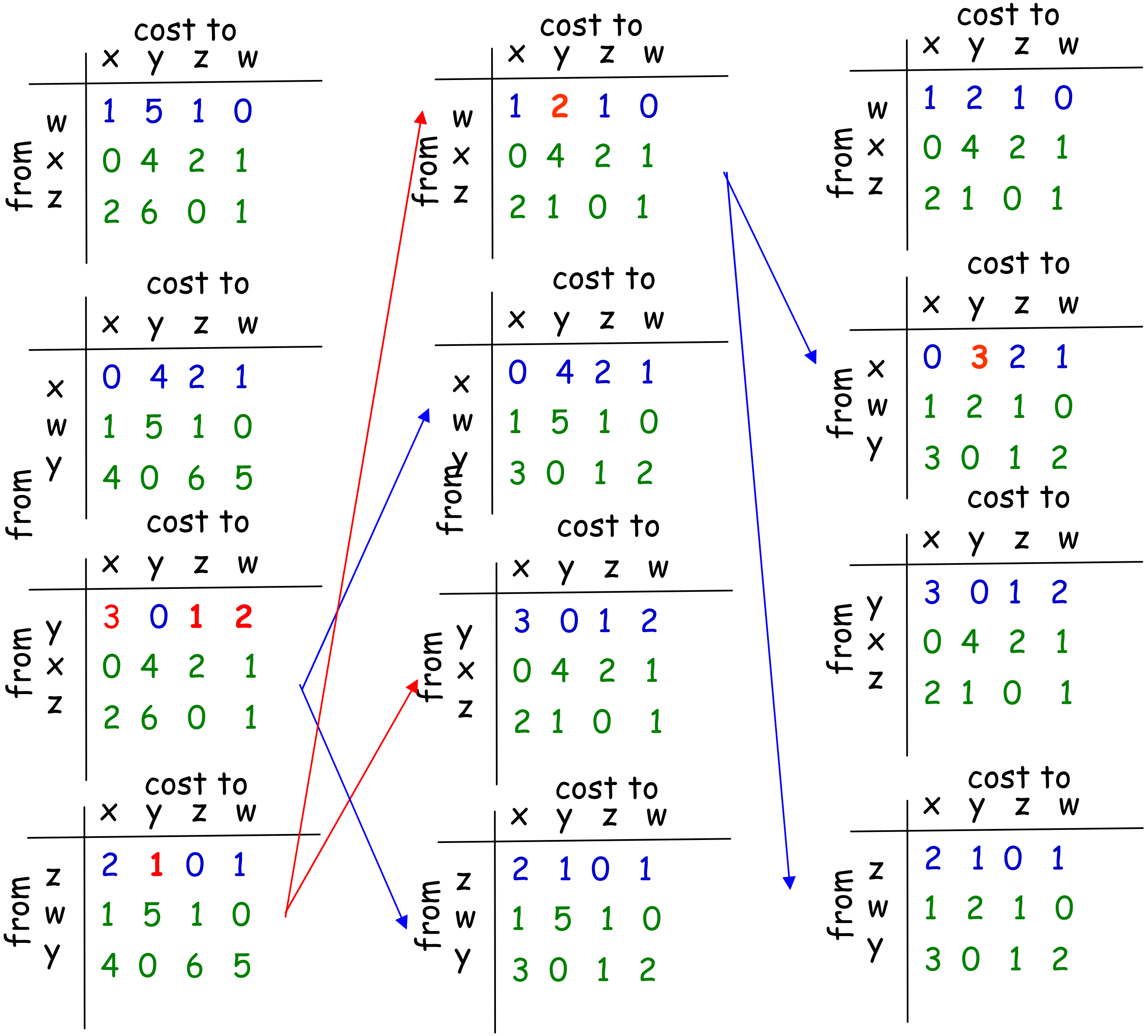
Initial routing table

(before change)



"good news travels fast"

Algorithm converges in 3 steps.



node w table

	cost to			
from	x	y	z	w
w	1	5	1	0
x	0	4	2	1
z	2	6	0	1

	cost to			
from	x	y	z	w
w	1	2	1	0
x	0	4	2	1
z	2	1	0	1

	cost to			
from	x	y	z	w
w	1	2	1	0
x	0	4	2	1
z	2	1	0	1

node x table

	cost to			
from	x	y	z	w
x	0	4	2	1
w	1	5	1	0
y	4	0	6	5

	cost to			
from	x	y	z	w
x	0	4	2	1
w	1	5	1	0
y	3	0	1	2

	cost to			
from	x	y	z	w
x	0	3	2	1
w	1	2	1	0
y	3	0	1	2

node y table

	cost to			
from	x	y	z	w
y	3	0	1	2
x	0	4	2	1
z	2	6	0	1

	cost to			
from	x	y	z	w
y	3	0	1	2
x	0	4	2	1
z	2	1	0	1

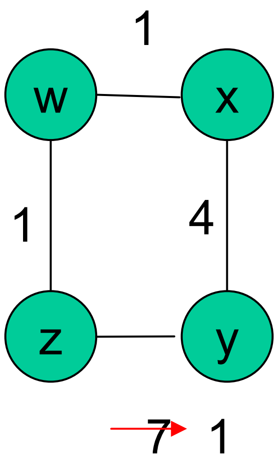
	cost to			
from	x	y	z	w
y	3	0	1	2
x	0	4	2	1
z	2	1	0	1

node z table

	cost to			
from	x	y	z	w
z	2	1	0	1
w	1	5	1	0
y	4	0	6	5

	cost to			
from	x	y	z	w
z	2	1	0	1
w	1	5	1	0
y	3	0	1	2

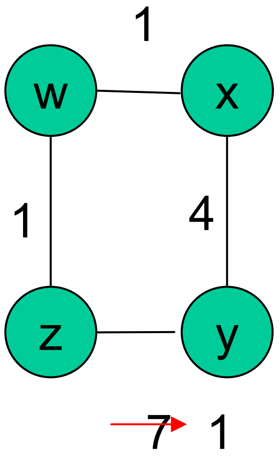
	cost to			
from	x	y	z	w
z	2	1	0	1
w	1	2	1	0
y	3	0	1	2



Cost of link zy changes to 1

"good news travels fast"

Algorithm converges in 3 steps.



Cost of link zy changes to 1

node w table

cost to

from	x	y	z	w
w	1	2	1	0
x	0	4	2	1
z	2	1	0	1

cost to

from	x	y	z	w
w	1	2	1	0
x	0	3	2	1
z	2	1	0	1

node x table

cost to

from	x	y	z	w
x	0	3	2	1
w	1	2	1	0
y	3	0	1	2

cost to

from	x	y	z	w
x	0	3	2	1
w	1	2	1	0
y	3	0	1	2

node y table

cost to

from	x	y	z	w
y	3	0	1	2
x	0	4	2	1
z	2	1	0	1

cost to

from	x	y	z	w
y	3	0	1	2
x	0	3	2	1
z	2	1	0	1

node z table

cost to

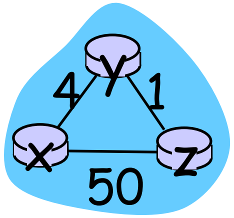
from	x	y	z	w
z	2	1	0	1
w	1	2	1	0
y	3	0	1	2

cost to

from	x	y	z	w
z	2	1	0	1
w	1	2	1	0
y	3	0	1	2

"bad news travels slow"

"count to infinity" problem



node x table

cost to

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

node y table

cost to

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

node z table

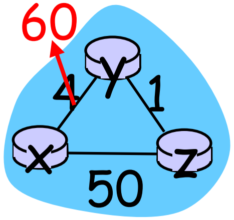
cost to

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

Initial routing table

"bad news travels slow"

Algorithm converges in 44 steps.



Cost of link xy changes to 60

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	4	0	1
	z	5	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	5	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	7	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	8	0	1
	z	7	1	0

node y table

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

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	5	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	8	0	1
	z	7	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	8	0	1
	z	7	1	0

.....

node z table

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

node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	7	1	0

node z table

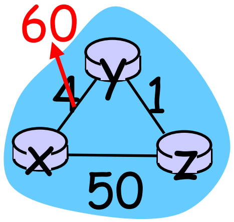
		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	7	1	0

node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	8	0	1
	z	9	1	0

"bad news travels slow"

Algorithm converges in 44 steps.



Cost of link xy changes to 60

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	48	0	1
	z	49	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	49	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	50	1	0

node x table

		cost to		
		x	y	z
from	x	0	51	50
	y	51	0	1
	z	50	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	49	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	49	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	51	0	1
	z	50	1	0

node y table

		cost to		
		x	y	z
from	x	0	51	50
	y	51	0	1
	z	50	1	0

node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	48	0	1
	z	49	1	0

node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	50	1	0

node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	50	0	1
	z	50	1	0

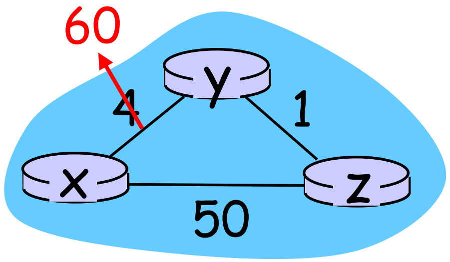
node z table

		cost to		
		x	y	z
from	x	0	51	50
	y	51	0	1
	z	50	1	0

Distance Vector: link cost changes

Link cost changes:

- ❒ good news travels fast
- ❒ bad news travels slow - "count to infinity" problem!
- ❒ 44 iterations before algorithm stabilizes: see text

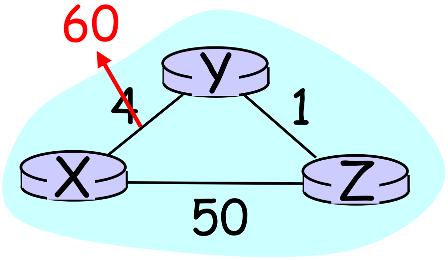


- ❑ By suitably increasing 50 by A and 60 by B, with A<B we can force algorithm to run as long as we want

- ❑ Real problem is that y thinks its shortest path to x is through z, while z thinks its shortest path to x is through y. They pingpong back and forth with this information.

- ❑Not good!

Distance Vector: 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?

poisoned reverse

node x table

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

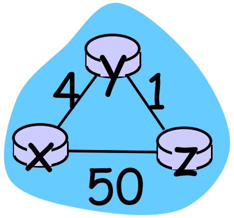
node y table

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

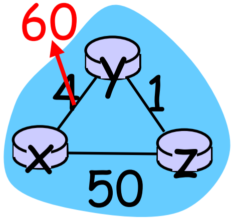
node z table

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

Initial routing table



Algorithm converges in 3 steps.



Cost of link xy changes to 60

node x table

	cost to		
	x	y	z
from x	0	51	50
y	4	0	1
z	5	1	0

node x table

	cost to		
	x	y	z
from x	0	51	50
y	60	0	1
z	5	1	0

node x table

	cost to		
	x	y	z
from x	0	51	50
y	60	0	1
z	50	1	0

node x table

	cost to		
	x	y	z
from x	0	51	50
y	51	0	1
z	50	1	0

node y table

	cost to		
	x	y	z
from x	0	4	∞
y	60	0	1
z	∞	1	0

node y table

	cost to		
	x	y	z
from x	0	51	50
y	60	0	1
z	∞	1	0

node y table

	cost to		
	x	y	z
from x	0	51	50
y	51	0	1
z	50	1	0

node y table

	cost to		
	x	y	z
from x	0	51	50
y	51	0	1
z	50	1	0

node z table

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

node z table

	cost to		
	x	y	z
from x	0	∞	50
y	60	0	1
z	50	1	0

node z table

	cost to		
	x	y	z
from x	0	∞	50
y	60	0	1
z	50	1	0

node z table

	cost to		
	x	y	z
from x	0	∞	50
y	∞	0	1
z	50	1	0

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) msgs sent
- DV: exchange between neighbors only
  - convergence time varies

Speed of Convergence

- LS: $O(n^2)$ 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 propagates thru network

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network "flat"

… *not* true in practice

scale: with 200 million destinations:

- can't store all dest's 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

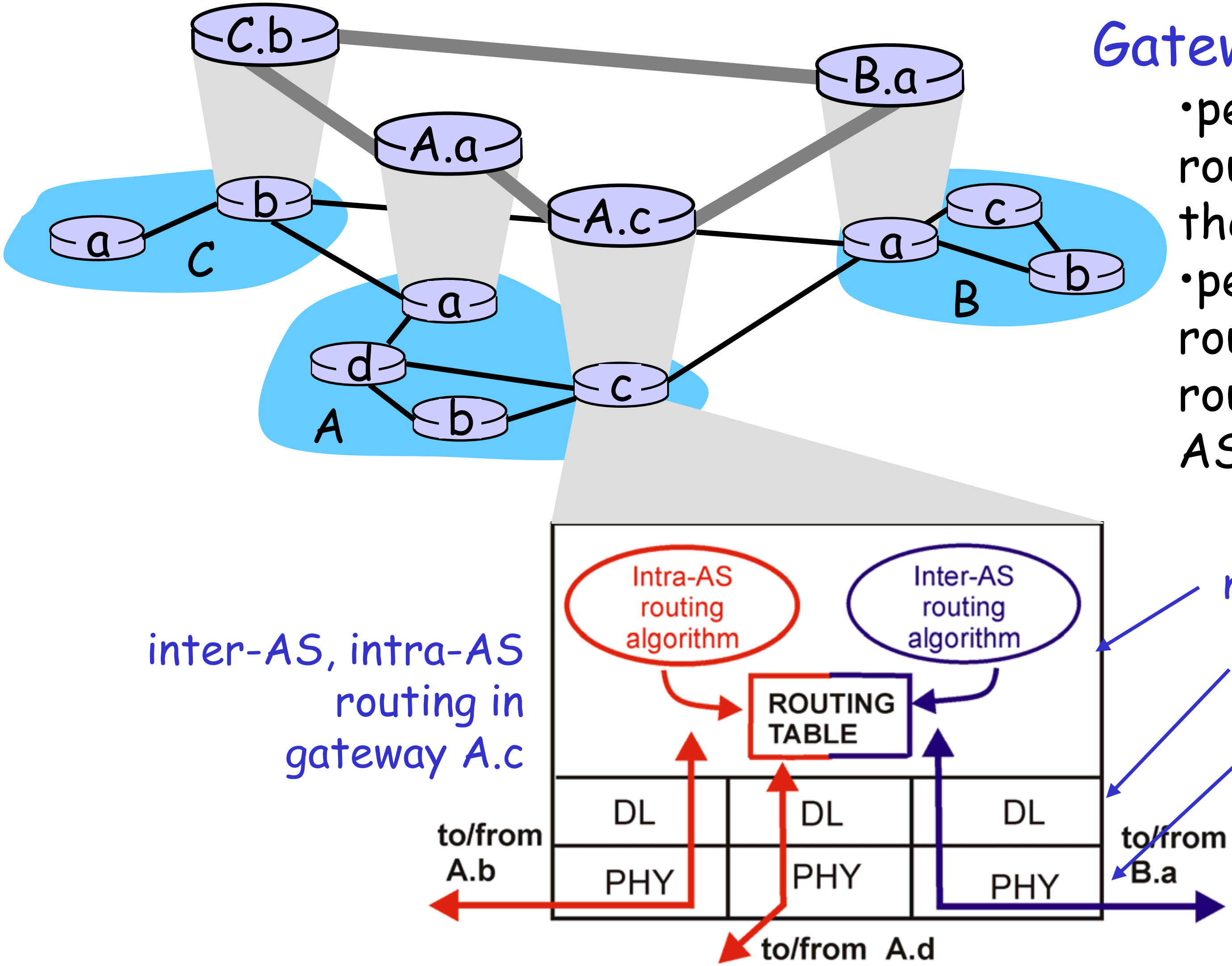
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - routers in different AS can run different intra-AS routing protocol

gateway routers

- special routers in AS
- run intra-AS routing protocol with all other routers in AS
- *also* responsible for routing to destinations outside AS
  - run *inter-AS routing* protocol with other gateway routers

Intra-AS and Inter-AS routing



Gateways:

- perform inter-AS routing amongst themselves
- perform intra-AS routing with other routers in their AS

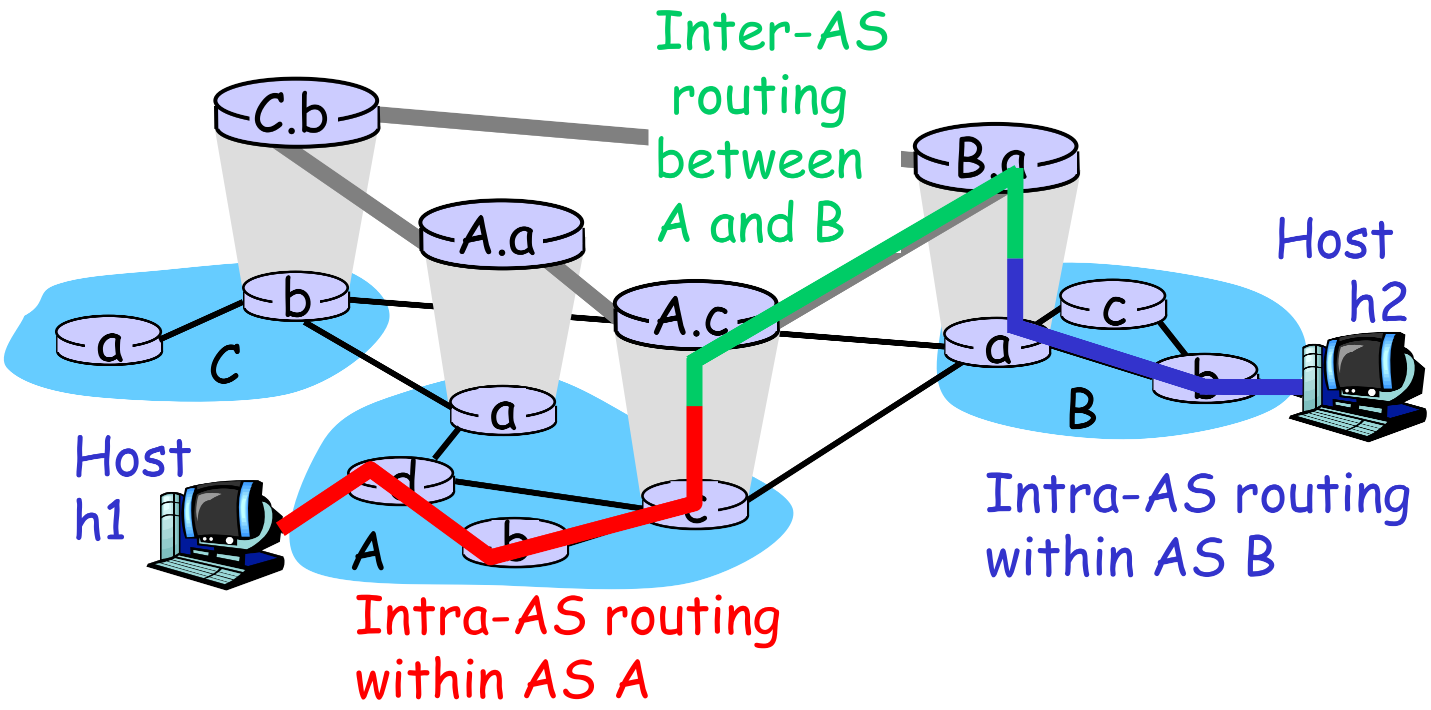
inter-AS, intra-AS routing in gateway A.c

network layer

link layer

physical layer

Intra-AS and Inter-AS routing



- We'll examine specific inter-AS and intra-AS Internet routing protocols shortly

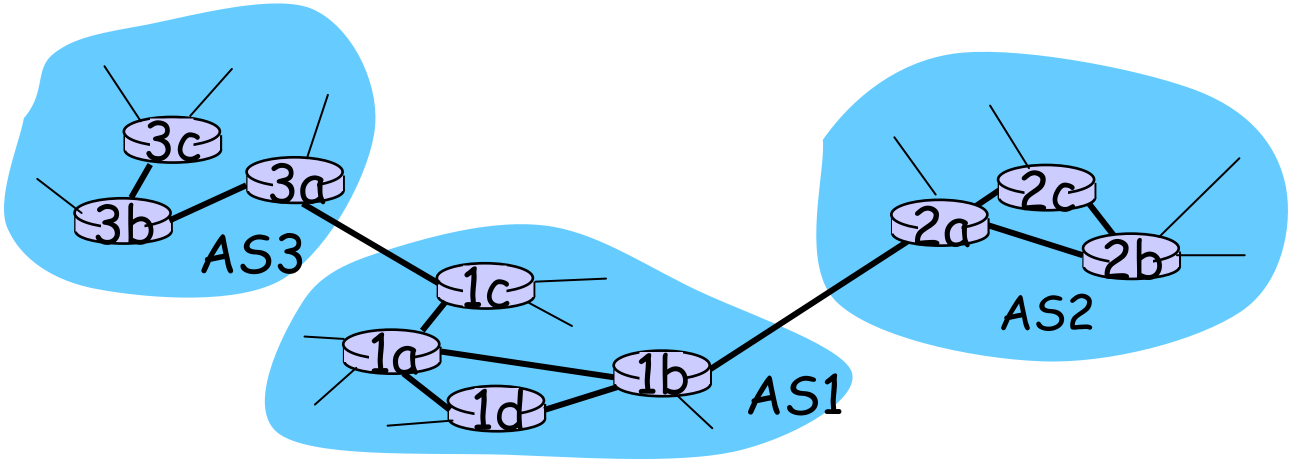
Inter-AS tasks

- Suppose router in AS1 receives datagram for which dest is outside of AS1
  - Router should forward packet towards one of the gateway routers, but which one?

AS1 needs:

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

Job of inter-AS routing!



Example: Setting forwarding table in router 1d

- Suppose AS1 learns from the inter-AS protocol that subnet *x* is reachable from AS3 (gateway 1c) but not from AS2.
- Inter-AS protocol propagates reachability info to all internal routers.
- Router 1d determines from intra-AS routing info that its interface *I* is on the least cost path to 1c.
- Adds entry *(x,I)* to forwarding table

Example: Choosing among multiple ASes

- Now suppose AS1 learns from the inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.
- To configure forwarding table, router 1d must determine towards *which* gateway it should forward packets for dest x.
- This is also the job of inter-AS routing protocol!
- Hot potato routing: send packet towards closest of two routers.

Learn from inter-AS protocol that subnet x is reachable via multiple gateways

Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways

Hot potato routing: Choose the gateway that has the smallest least cost

Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table

Chapter 4: Network Layer

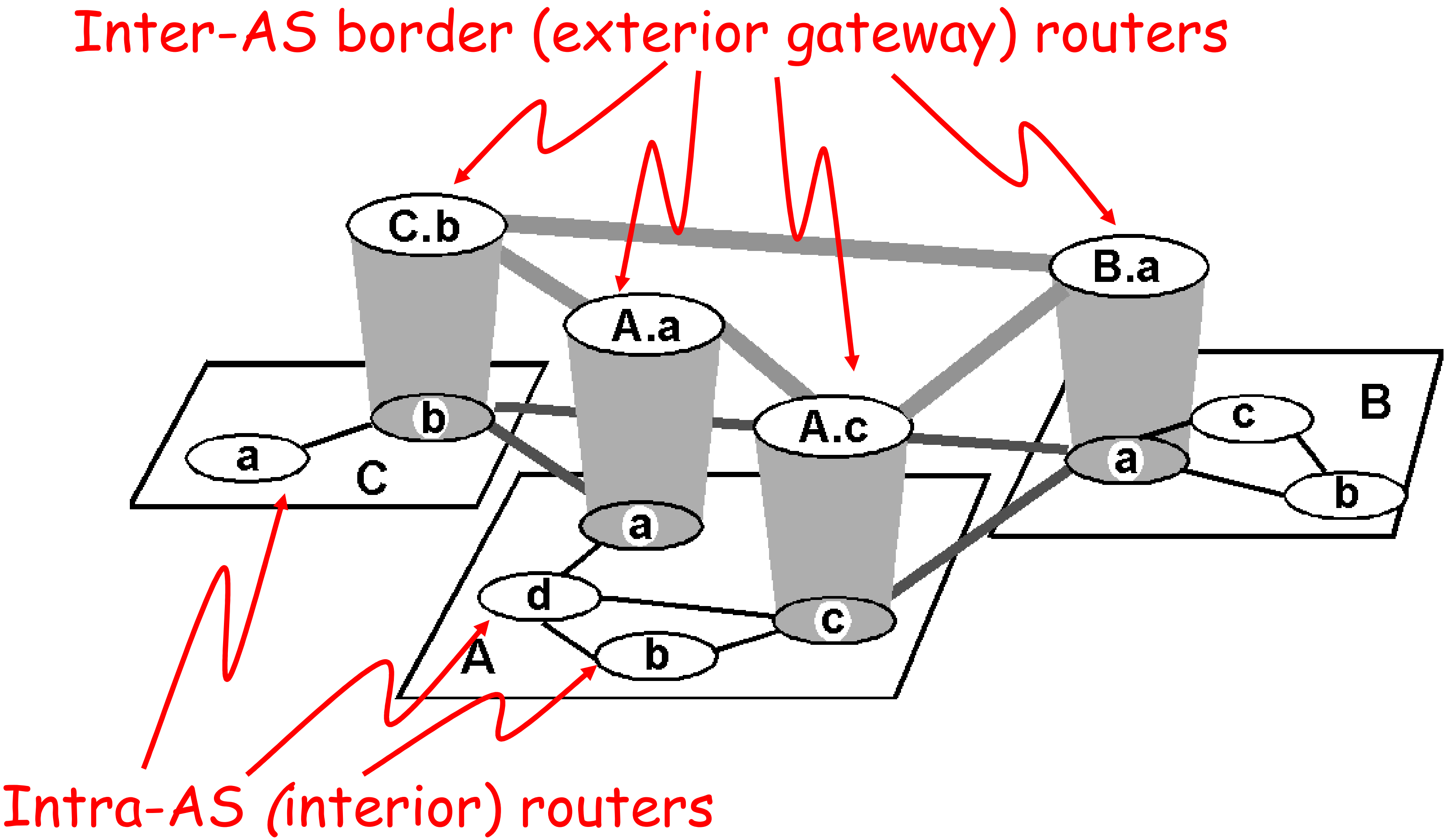
- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

Routing in the Internet

- The Global Internet consists of Autonomous Systems (AS) interconnected with each other:
  - **Stub AS**: small corporation: one connection to other AS's
  - **Multihomed AS**: large corporation (no transit): multiple connections to other AS's
  - **Transit AS**: provider, hooking many AS's together

- Two-level routing:
  - **Intra-AS:** (within AS) administrator responsible for choice of routing algorithm within network
  - **Inter-AS:** (between Ass) unique standard for inter-AS routing: BGP

Internet AS Hierarchy



Inter-AS border (exterior gateway) routers

Intra-AS (interior) routers

Intra-AS Routing

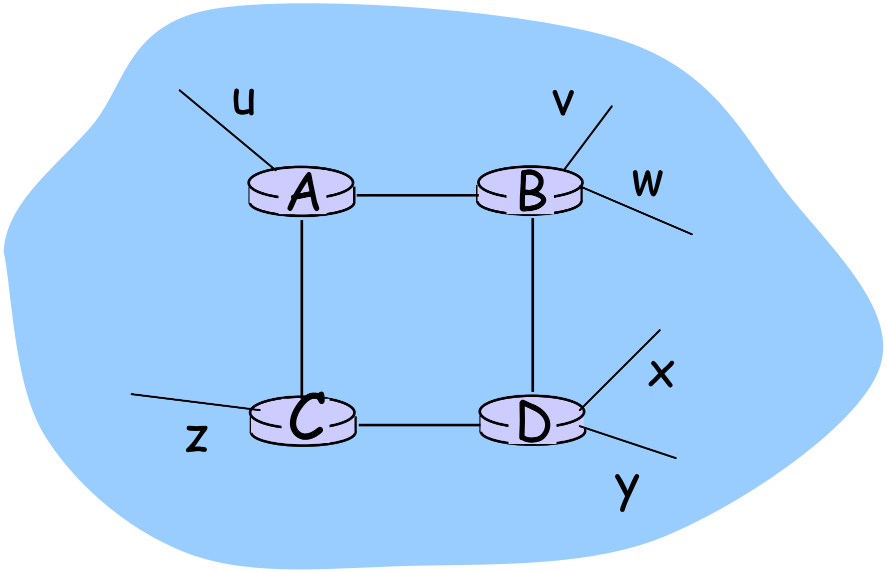
- Also known as Interior Gateway Protocols (IGP)
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)

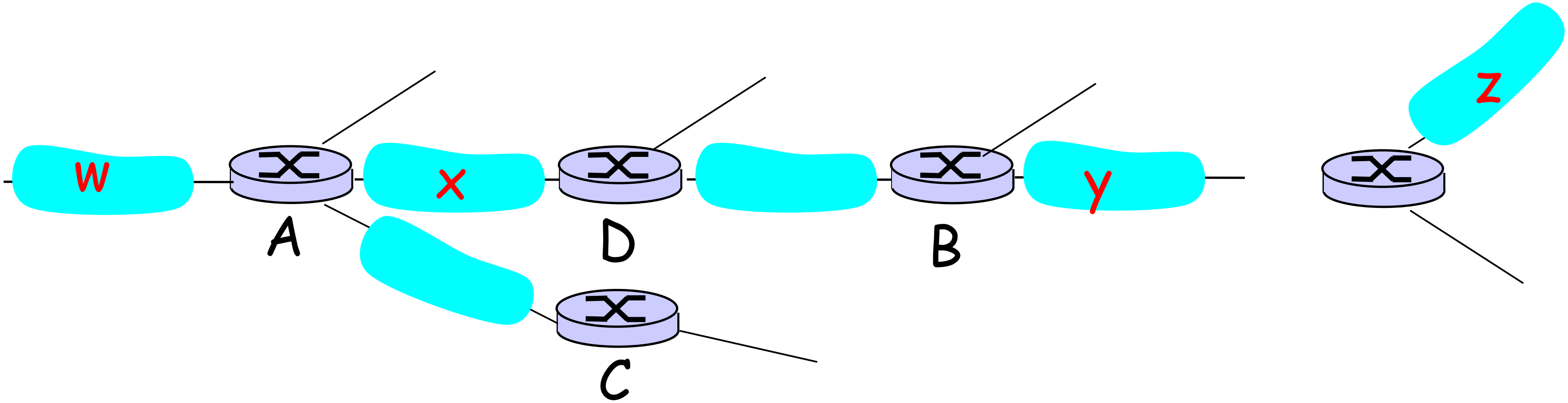


destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- Each advertisement: list of up to 25 destination nets within AS

RIP: Example



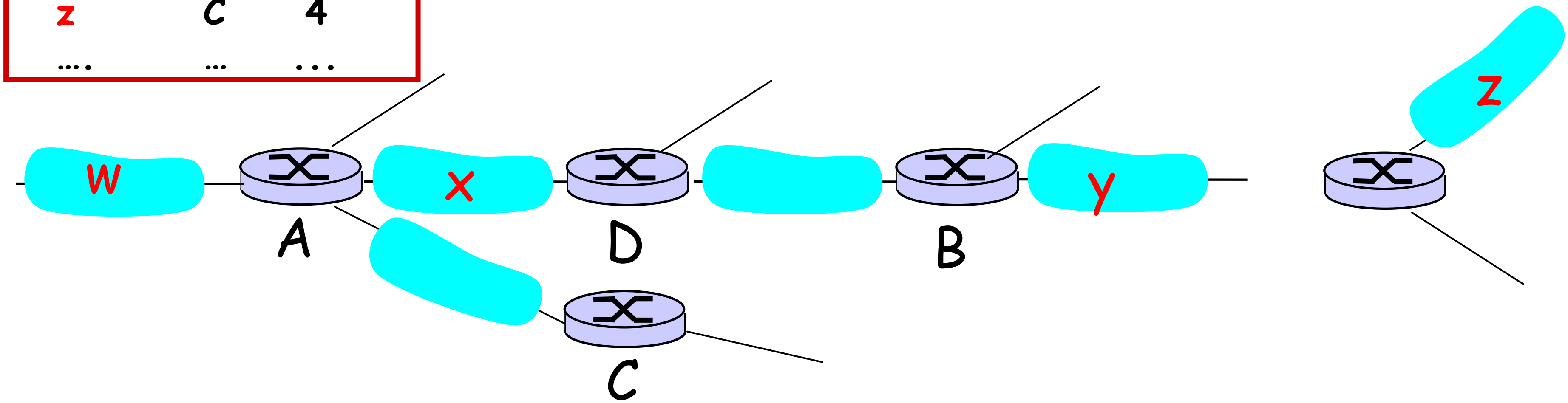
Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

Routing table in D

RIP: Example

Dest	Next	hops
w	-	-
x	-	-
z	C	4
….	…	. . .

Advertisement from A to D



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	. . . .

Routing table in D

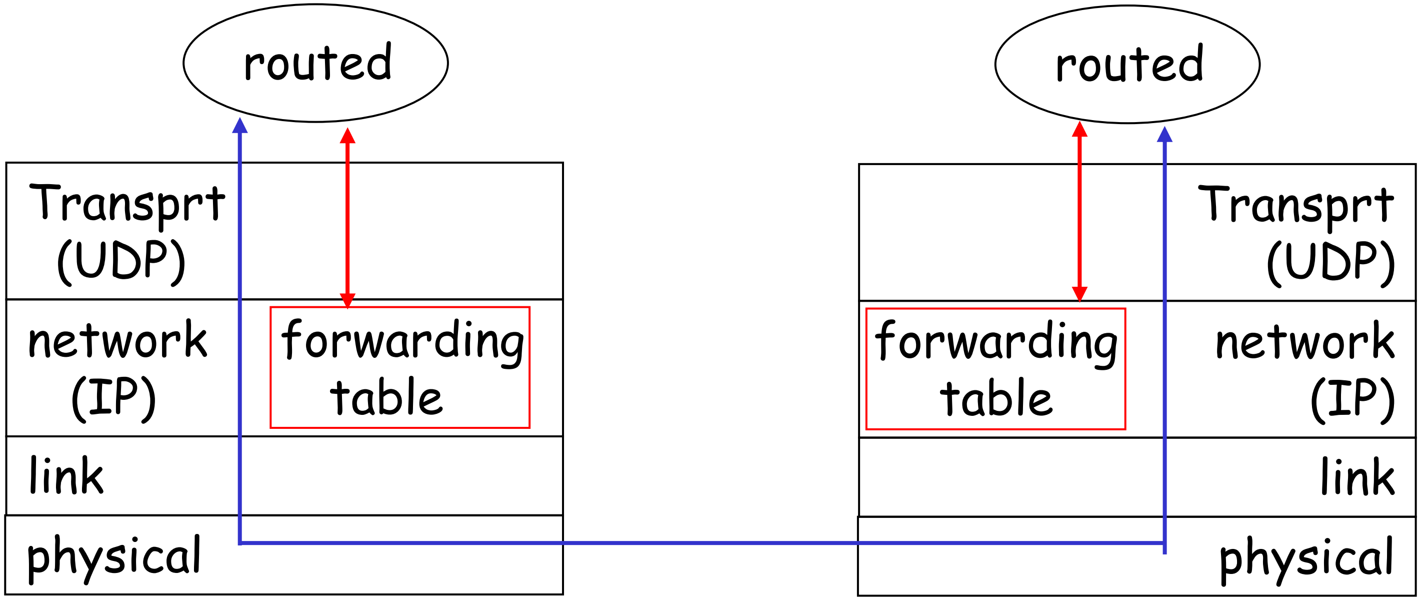
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec --> neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated



RIP Table example (continued)

Router: *giroflee.eurocom.fr*

```
  Destination            Gateway              Flags  Ref   Use    Interface
-------------------- -------------------- ----- ----- ------ ---------
127.0.0.1            127.0.0.1             UH       0  26492  lo0
192.168.2.           192.168.2.5           U        2     13  fa0
193.55.114.          193.55.114.6          U        3  58503  le0
192.168.3.           192.168.3.5           U        2     25  qaa0
224.0.0.0            193.55.114.6          U        3      0  le0
default              193.55.114.129        UG       0 143454
```

- Three attached class C networks (LANs)
- Router only knows routes to attached LANs
- Default router used to "go up"
- Route multicast address: 224.0.0.0
- Loopback interface (for debugging)

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

OSPF (Open Shortest Path First)

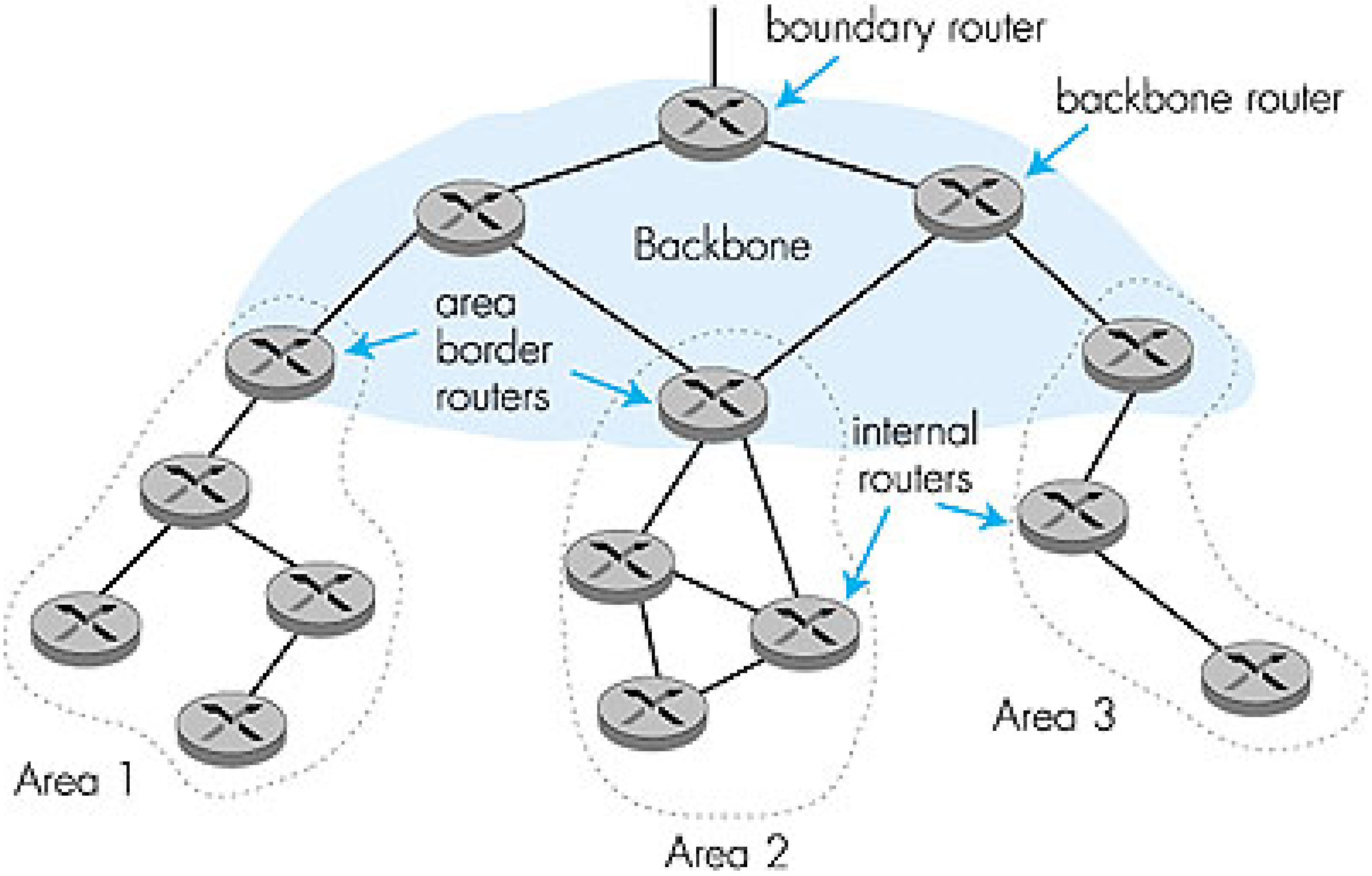
- “open”: publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm

- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP

OSPF “advanced” features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion)
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set “low” for best effort; high for real time)
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical OSPF in large domains.

Hierarchical OSPF



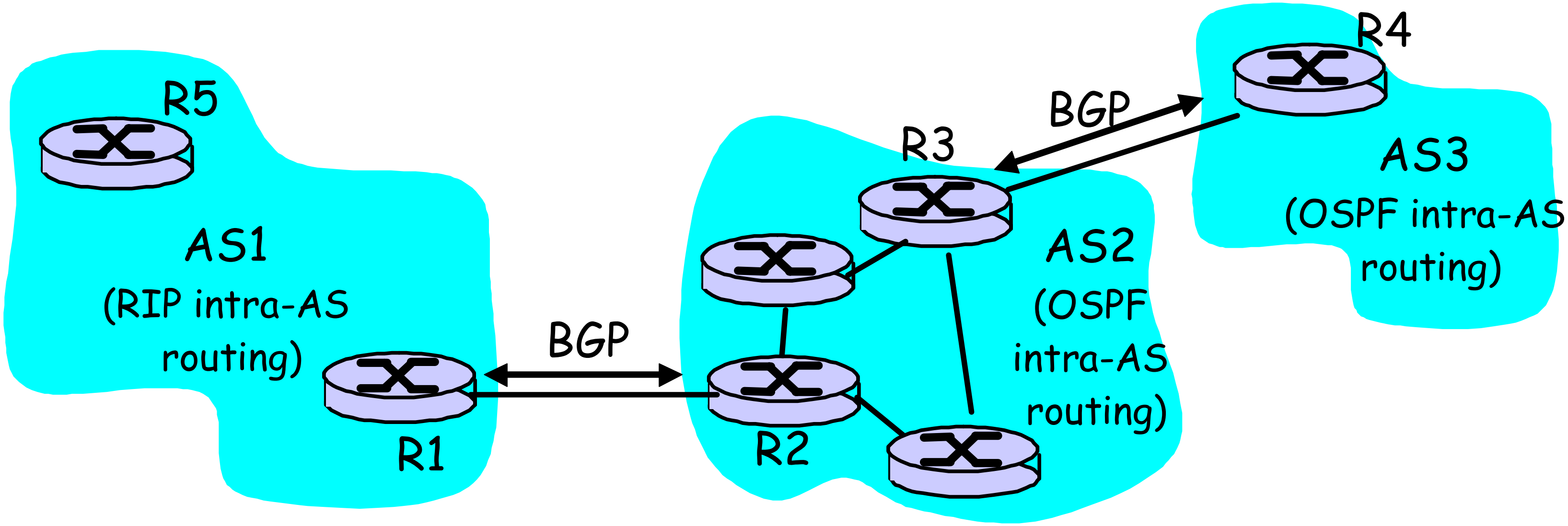
Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers:** "summarize" distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS's.

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing (maybe)

Inter-AS routing in the Internet: BGP



Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto standard
- BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine "good" routes to subnets based on reachability information and policy.
- Allows a subnet to advertise its existence to rest of the Internet: *"I am here"*

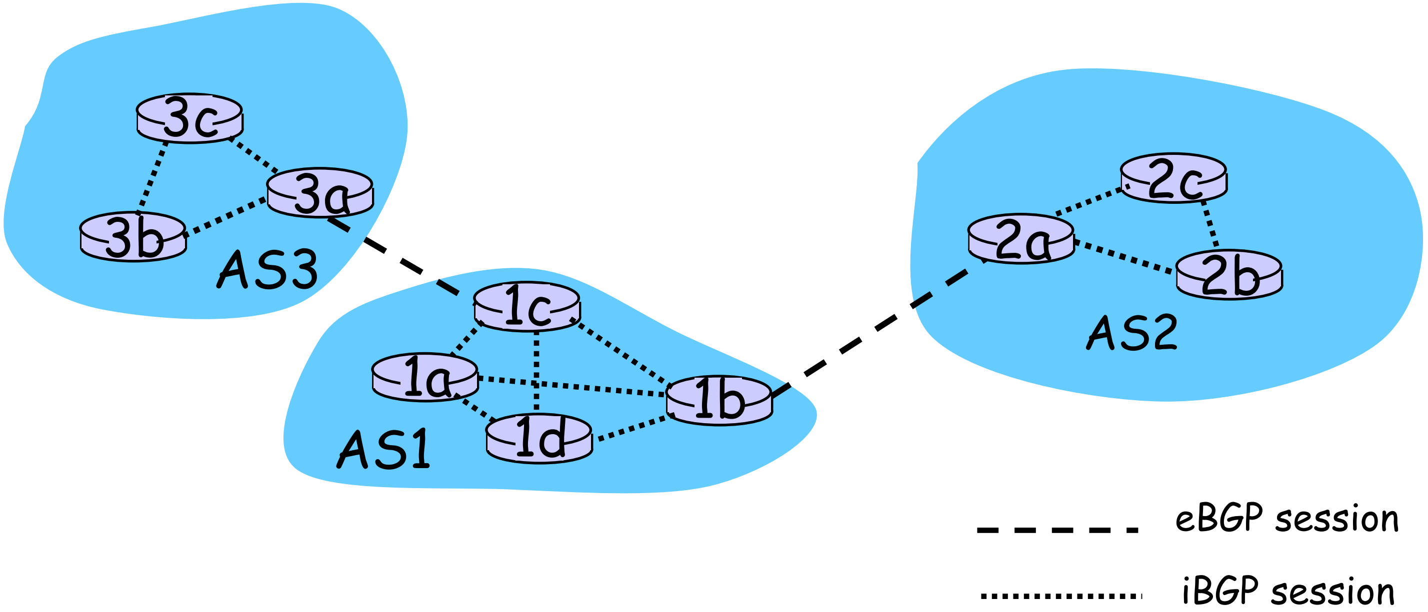
- In BGP, destination are not individual hosts, they are networks!

- A network is represented by a CIDR prefix, e.g., 138.16.64/24
- If a gateway router broadcasts a BGP message stating that it is 138.16.64/24, it is ***advertising*** that it can deliver messages to any host in subnet 138.16.64/24.

- BGP messages between routers in same AS are called (interior) iBGP messages
- BGP messages between routers in diff AS are called (exterior) eBGP messages

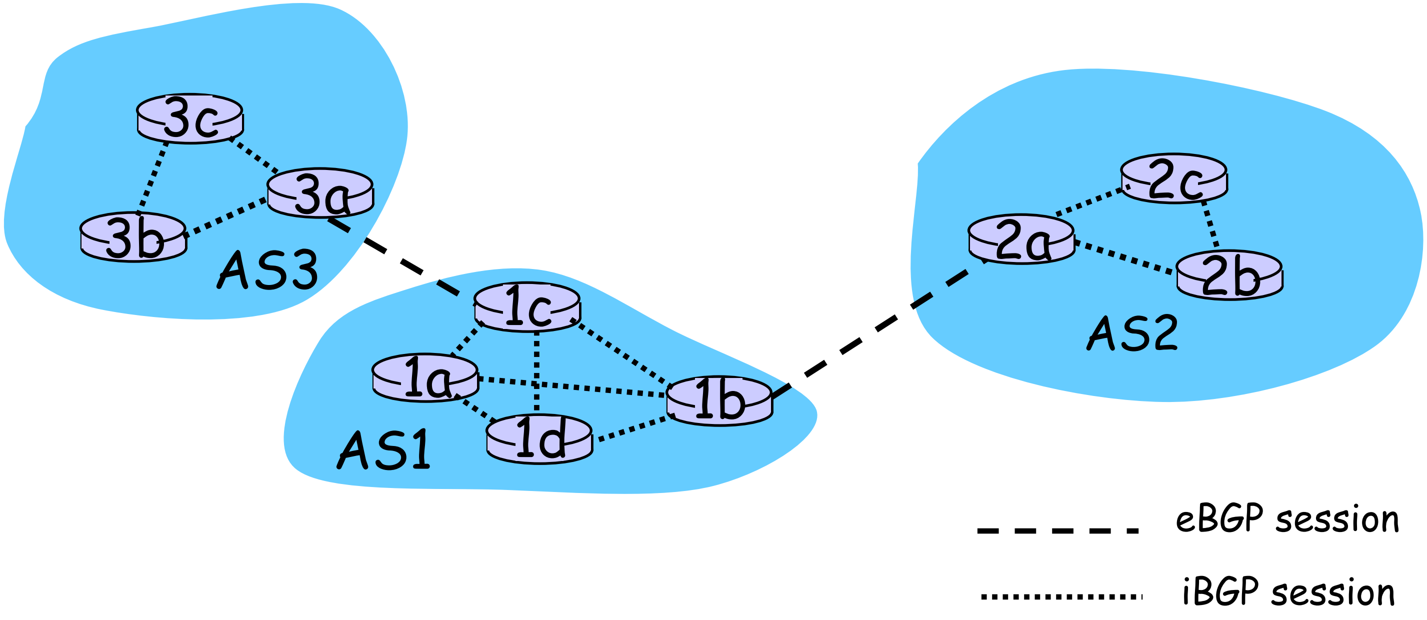
BGP basics

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP conctns: BGP sessions
- Note that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement



Distributing reachability info

- With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
- 1c can then use iBGP do distribute this new prefix reach info to all routers in AS1
- 1b can then re-advertise the new reach info to AS2 over the 1b-to-2a eBGP session
- When router learns about a new prefix, it creates an entry for the prefix in its forwarding table.



Path attributes & BGP routes

- When advertising a prefix, advert includes BGP attributes.
  - prefix + attributes = "route"
- Two important attributes:
  - AS-PATH: contains the ASs through which the advert for the prefix passed: AS 67, AS 17, …
  - NEXT-HOP: Indicates the specific internal-AS router to next-hop AS. (There may be multiple links from current AS to next-hop-AS.)
- When gateway router receives route advert, uses import policy to accept/decline.

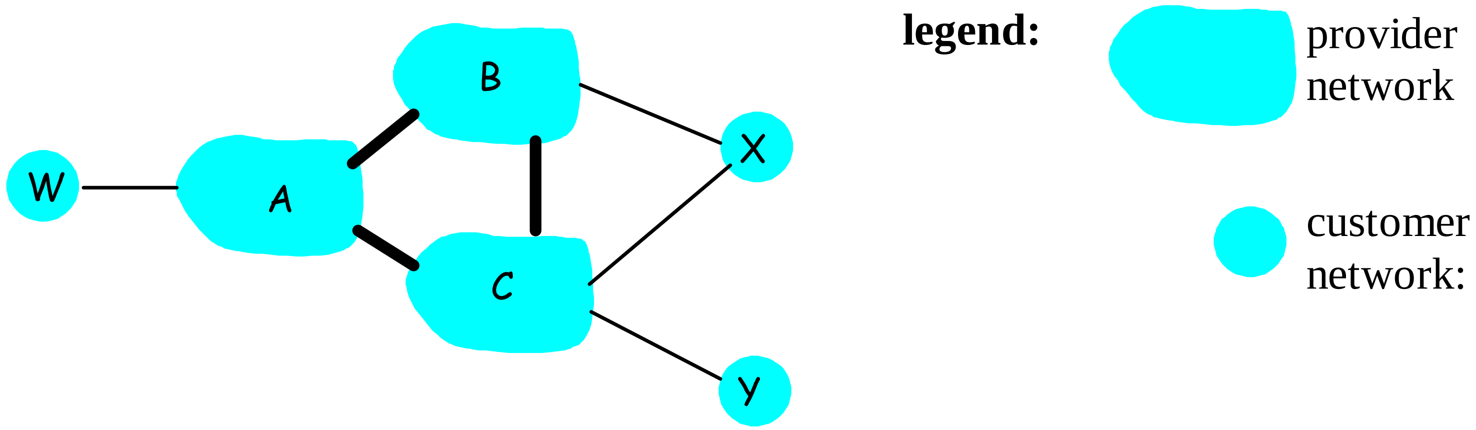
BGP route selection

- Router may learn about more than 1 route to some prefix. Router must select route.
- Elimination rules:
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria

BGP messages

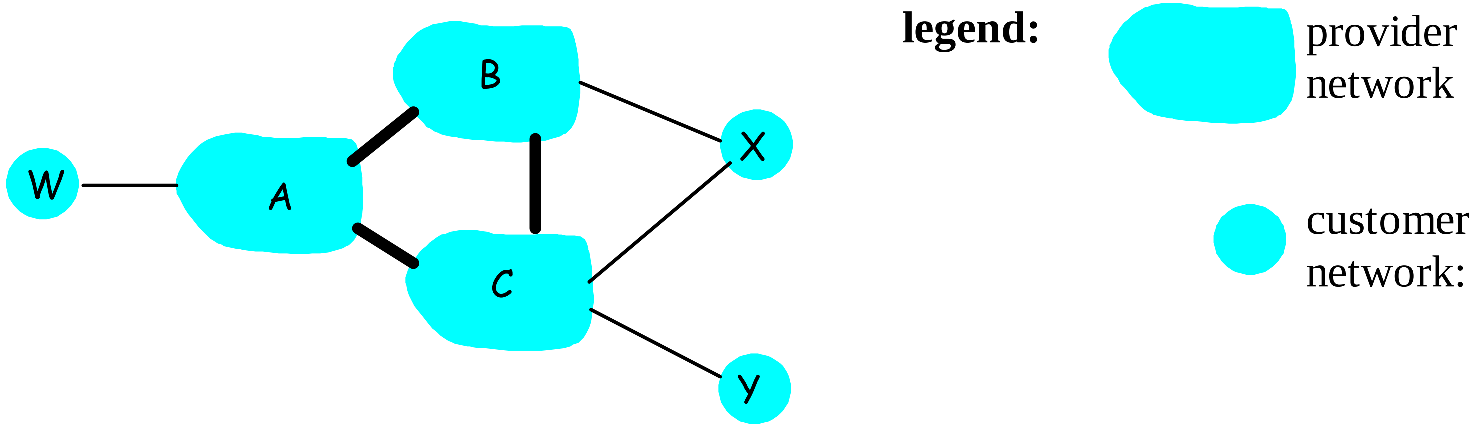
- BGP messages exchanged using TCP.
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP: Controlling who routes through you



- A,B,C are provider networks
- X,W,Y are customer (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C

BGP: Controlling who routes through you



- A advertises to B the path AW
- B advertises to X the path BAW
- Should B advertise to C the path BAW?
  - No way! B gets no "revenue" for routing CBAW since neither W nor C are B's customers
  - B wants to force C to route to w via A
  - B wants to route *only* to/from its customers!

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
    - Datagram format
    - IPv4 addressing
    - ICMP
    - IPv6
- 4.5 Routing algorithms
    - Link state
    - Distance Vector
    - Hierarchical routing
- 4.6 Routing in the Internet
    - RIP
    - OSPF
    - BGP
- 4.7 Broadcast and multicast routing (maybe: See textbook)

Network Layer: summary

What we've covered:

- network layer services
- routing principles: link state and distance vector
- hierarchical routing
- IP
- Internet routing protocols RIP, OSPF, BGP
- what's inside a router?
- IPv6

Next stop:

the Data link layer!