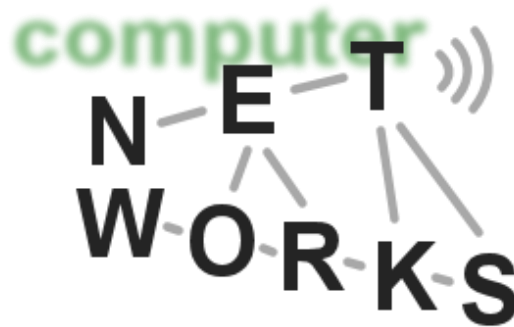# Network Layer – Part II

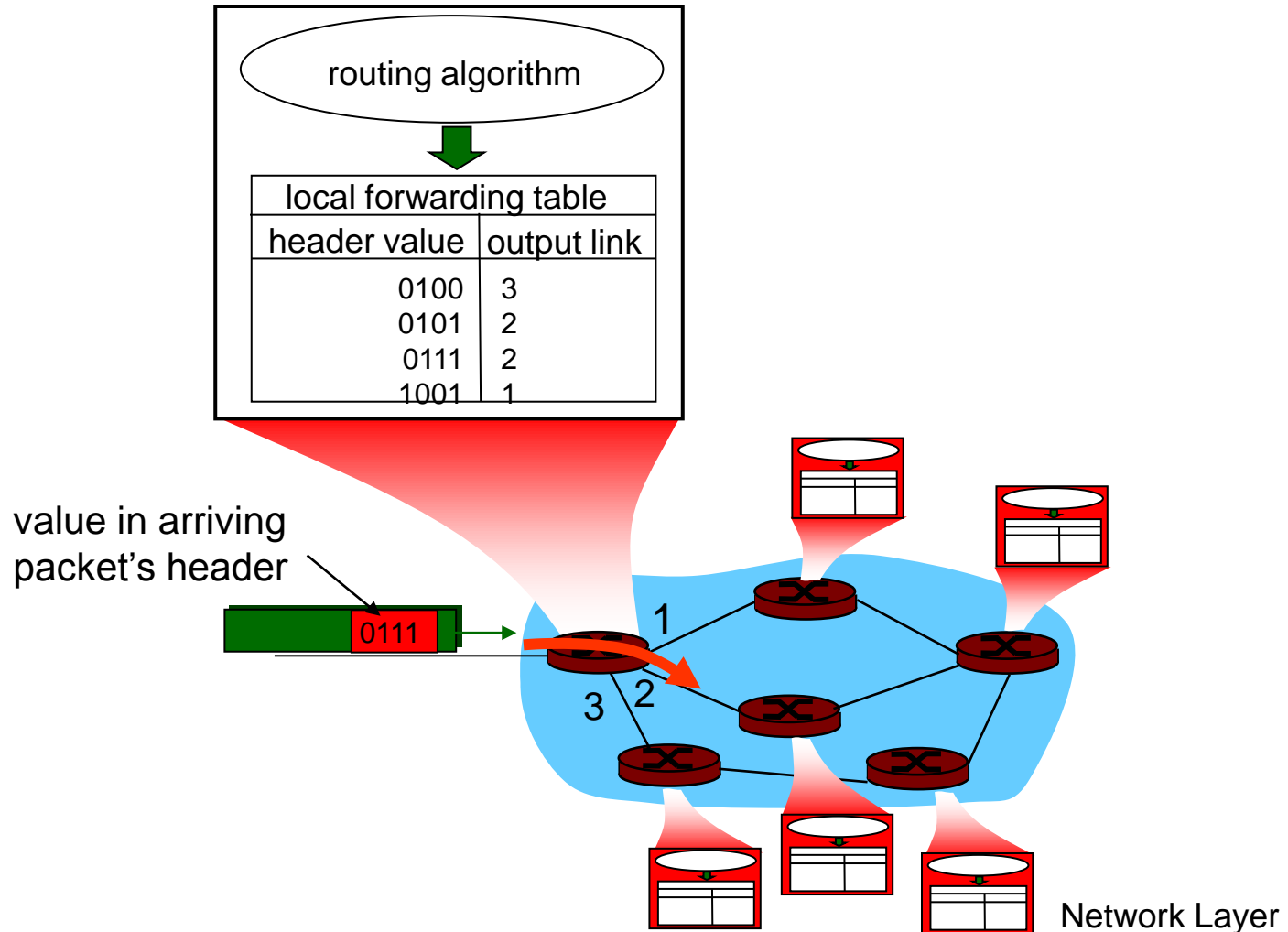## Computer Networks, Winter 2010/2011

Network Layer II

- 4.4 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.5 Routing protocols
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
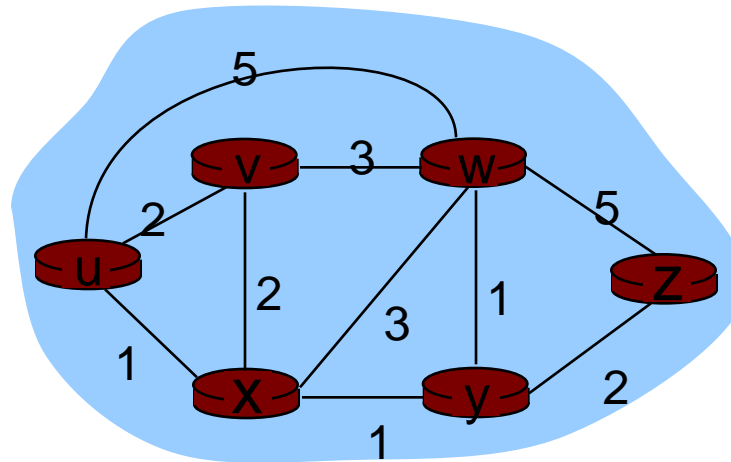 - Border Gateway Protocol (BGP)

Routing, Forwarding, and Switching

o Switching

- o Hardware-based relaying of packets
- o Relies on switching fabric
- o Layer-2 switches, Layer-3 switches

o Forwarding

- o Relaying of packets from input port to output port(s)
- o Based on forwarding table

o Routing

- o Process of configuring the forwarding of a node

Interplay between routing and forwarding

routing algorithm

local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

value in arriving packet's header

0111

Graph abstraction

Graph: G = (N,E)
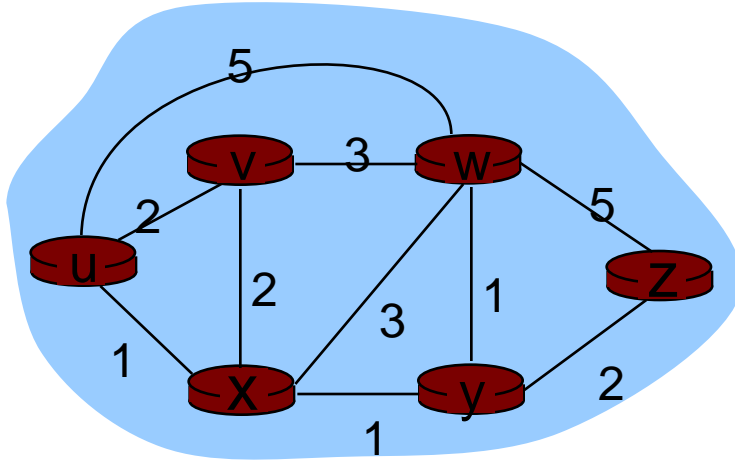
N = set of routers = { u, v, w, x, y, z }

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?

o Global:
  o all routers have complete topology, link cost info
  o "link state" algorithms

o Decentralized:
  o router knows physically-connected neighbors, link costs to neighbors
  o iterative process of computation, exchange of info with neighbors
  o "distance vector" algorithms

Static or dynamic?

o Static:
  o routes change slowly over time

o Dynamic:
  o routes change more quickly
    • periodic update
    • in response to link cost changes

Network Layer II

- 4.4 Routing algorithms
    - Link state
    - Distance Vector
    - Hierarchical routing
- 4.5 Routing protocols
    - Routing Information Protocol (RIP)
    - Open Shortest Path First (OSPF)
    - Border Gateway Protocol (BGP)

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; $= \infty$ 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 is definitively known

Dijsktra'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) = ∞
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'
```
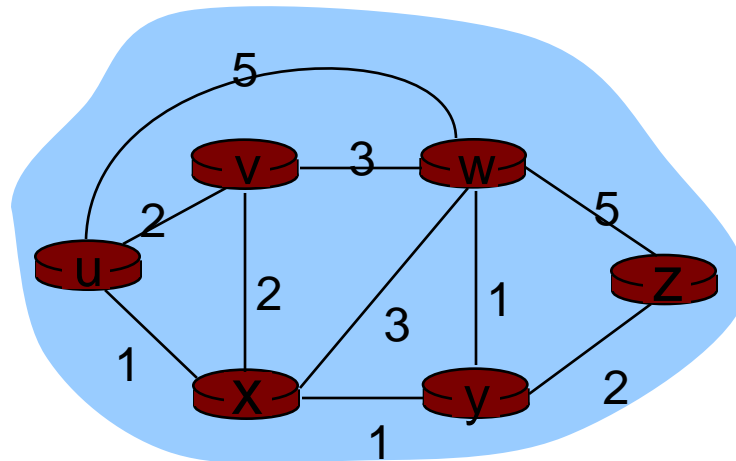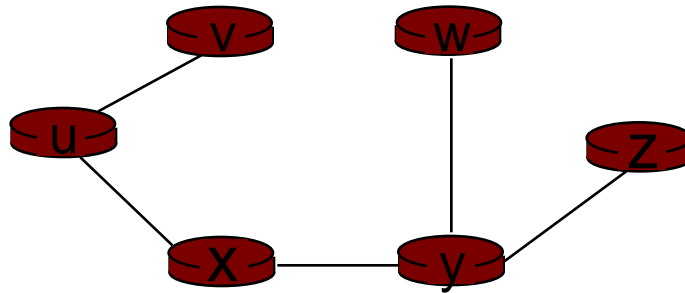
Dijkstra's algorithm: example

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:

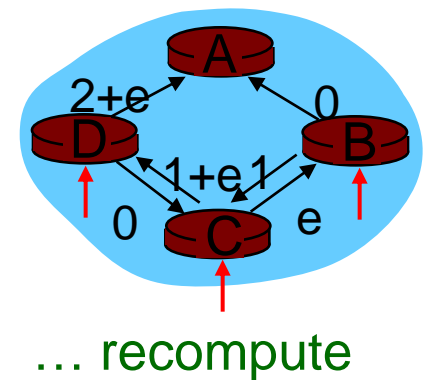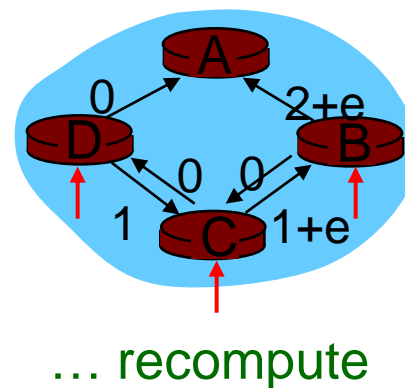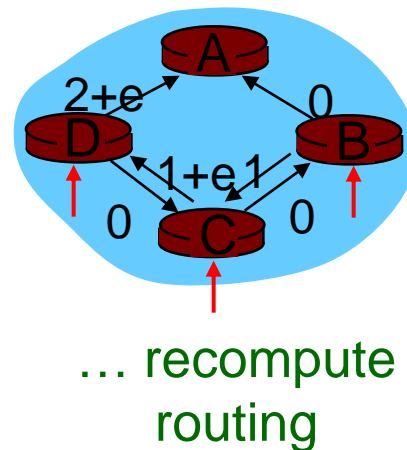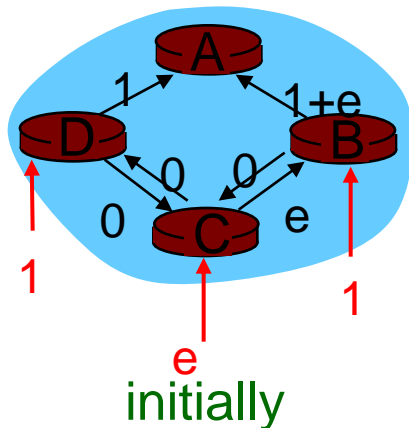| destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Dijkstra's algorithm, discussion

Algorithm complexity: n nodes

o each iteration: need to check all nodes, w, not in N

o n(n+1)/2 comparisons: $O(n^2)$

o more efficient implementations possible: $O(n*\log(n))$

Oscillations possible:

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

initially

… recompute routing

… recompute

… recompute

Network Layer II

o 4.4 Routing algorithms
- o Link state
- o Distance Vector
- o Hierarchical routing

o 4.5 Routing protocols
- o Routing Information Protocol (RIP)
- o Open Shortest Path First (OSPF)
- o Border Gateway Protocol (BGP)

Bellman–Ford algorithm

o Finds shortest paths in weighted, directed graph for given source

o Approach: relax all edges repeatedly until stable ($|V| - 1$ times)

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

o Then
$d_x(y) = min(c(x,v) + d_v(y))$
where min is taken over all neighbors v of x

Bellman-Ford example
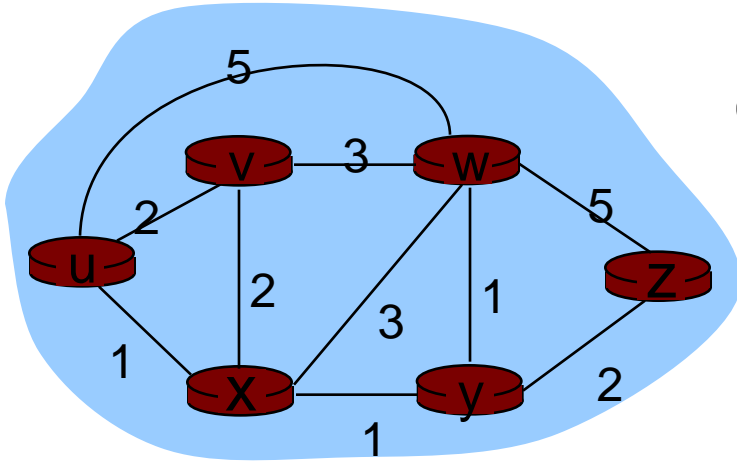
Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

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

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

Distance Vector Algorithm

- $D_x(y)$ = estimate of least cost from x to y

- Node x knows cost to each neighbor v: $c(x,v)$

- Node x maintains distance vector $\mathbf{D}_x = [D_x(y): y \in N]$

- Node x also maintains its neighbors' distance vectors
  - For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \in N]$

Distance vector algorithm – Basic Idea

o From time-to-time, each node sends its own distance vector estimate to neighbors

o Asynchronous

o When a node x receives new DV estimate from neighbor, it updates its own DV using B-F equation

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

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

Distance Vector Algorithm (cont'd)

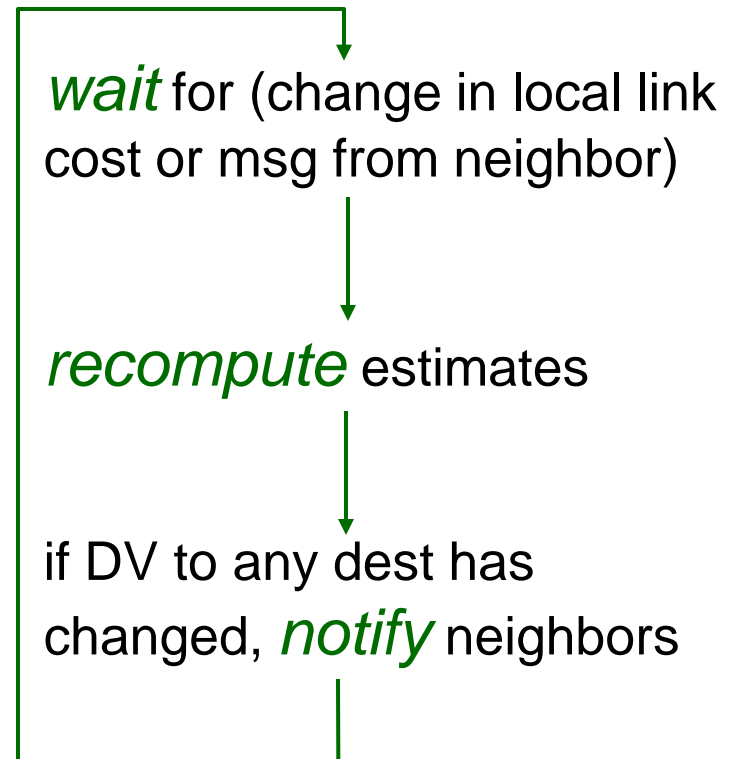## Iterative, asynchronous:

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

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from

cost to

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

from

node y table

cost to

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from

node z table

cost to

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from

time

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

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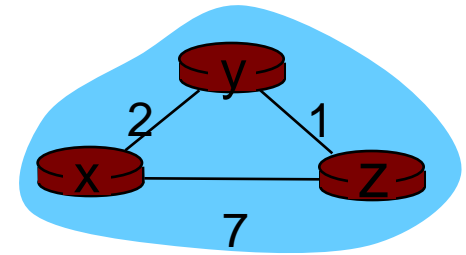
node x table

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

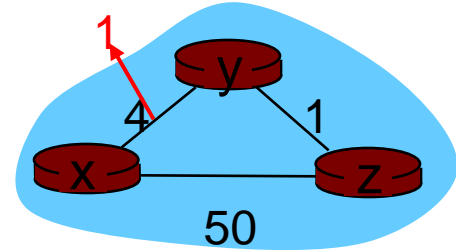| from	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

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

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.

"good news travels fast"

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.

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
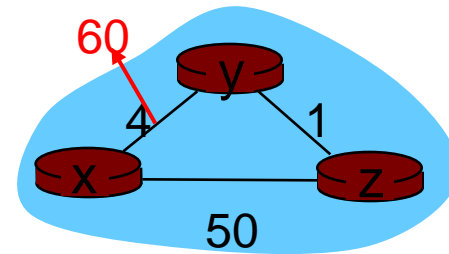
Poisoned reverse:

- 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?

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(n²) 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 through network

Network Layer II

o 4.4 Routing algorithms
- o Link state
- o Distance Vector
- o Hierarchical routing

o 4.5 Routing protocols
- o Routing Information Protocol (RIP)
- o Open Shortest Path First (OSPF)
- o Border Gateway Protocol (BGP)

Hierarchical Routing

o Our routing study thus far - idealization

   o all routers identical

   o network "flat"

   o … not true in practice

o Scale: with 200 million destinations:

   o can't store all dest's in routing tables

   o routing table exchange would swamp links

o Administrative autonomy

   o internet = network of networks

   o each network admin may want to control routing in its own network

Hierarchical Routing

o Aggregate routers into regions

  o Autonomous Systems (AS)

o Routers in same AS run same routing protocol

  o "intra-AS" routing protocol

  o routers in different AS can run different intra-AS routing protocol

o Gateway router

  o Direct link to router in another AS

Interconnected ASes
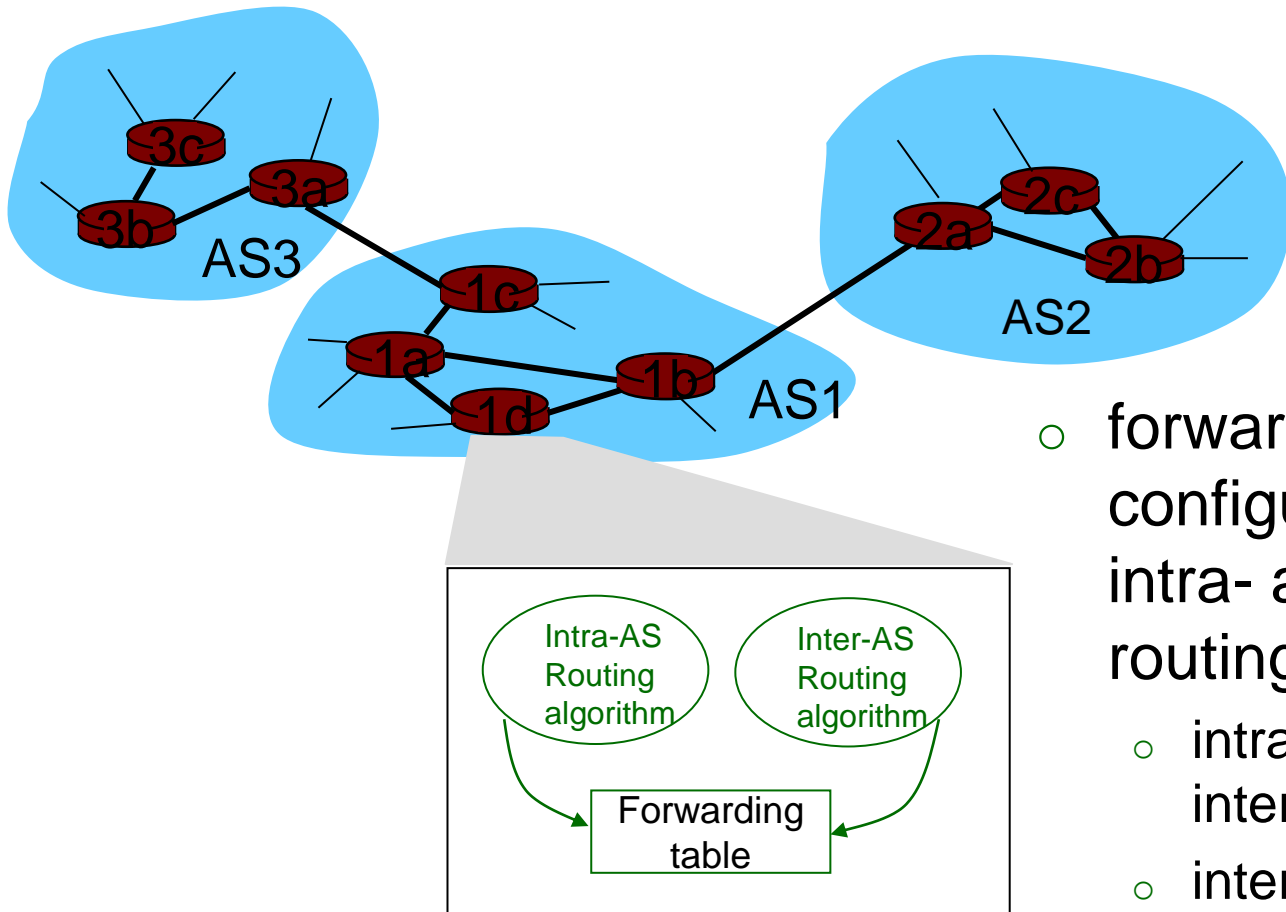
- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
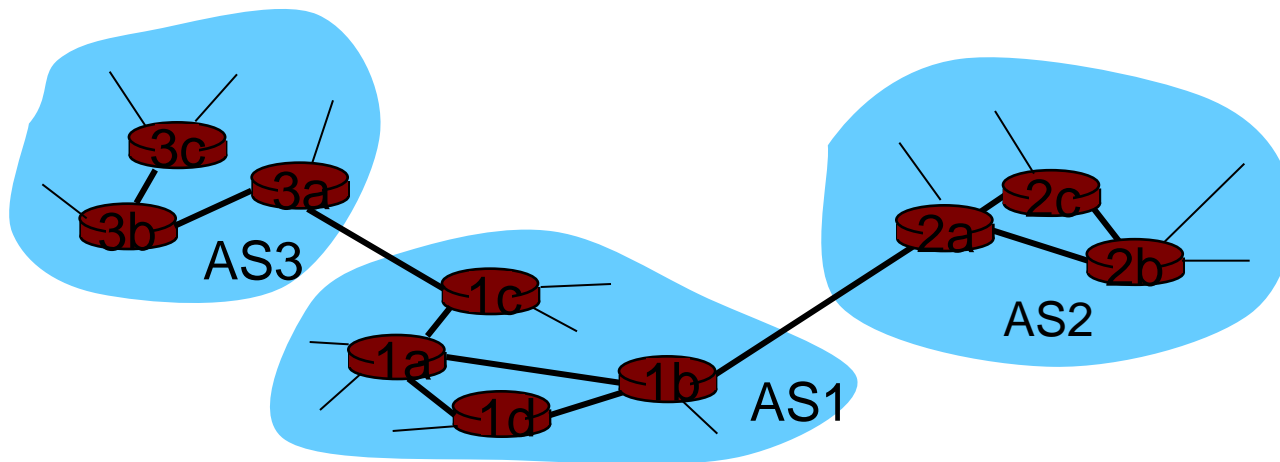 - inter-AS & intra-As sets entries for external dests

Inter-AS tasks

o suppose router in AS1 receives datagram destined outside of AS1:

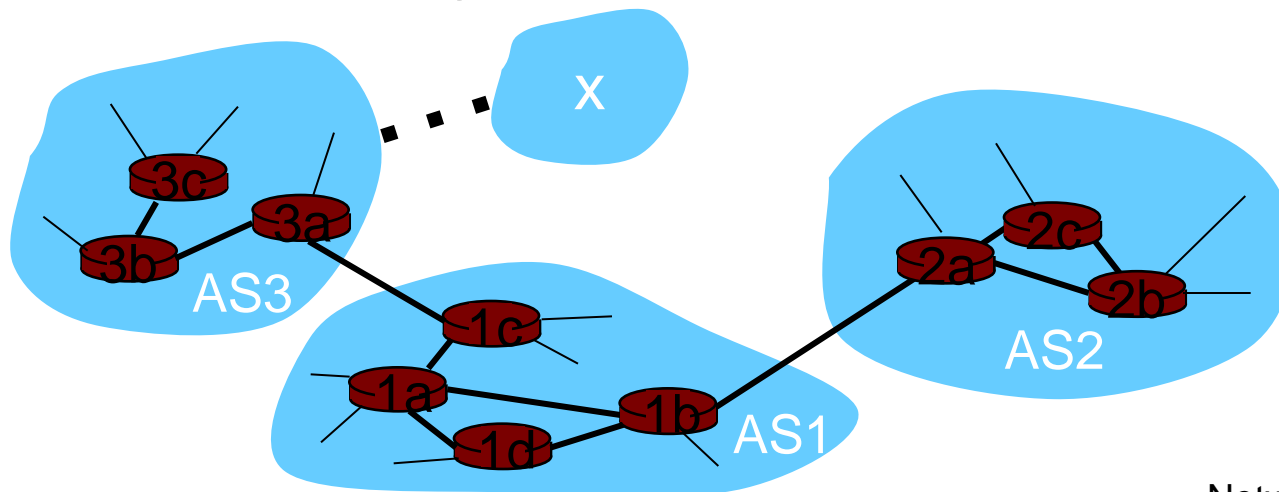 o router should forward packet to gateway router, but which one?

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!

Example: Setting forwarding table in router 1d

○ suppose AS1 learns (via inter-AS protocol) that subnet *x* reachable via AS3 (gateway 1c) but not via 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.

    ○ installs forwarding table entry *(x,I)*

Example: Choosing among multiple ASes

o now suppose AS1 learns from inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.

o to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.

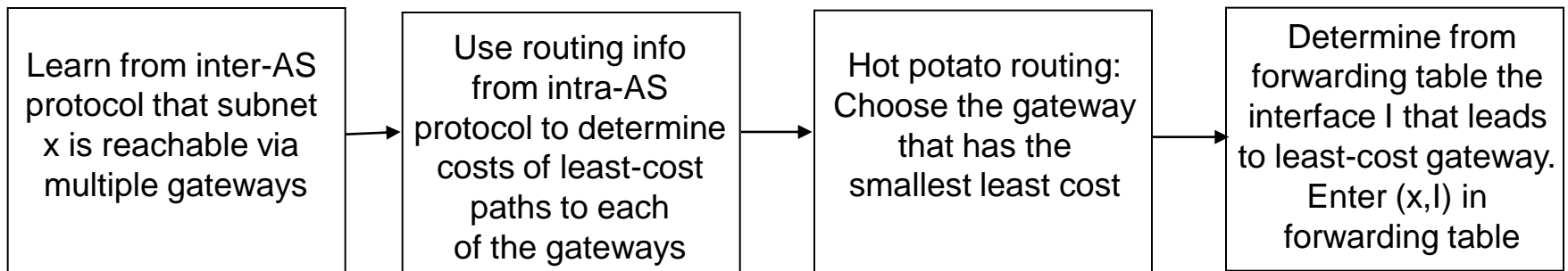 o this is also job of inter-AS routing protocol!

Example: Choosing among multiple ASes

o now suppose AS1 learns from inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.

o to configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.

   o this is also job of inter-AS routing protocol!
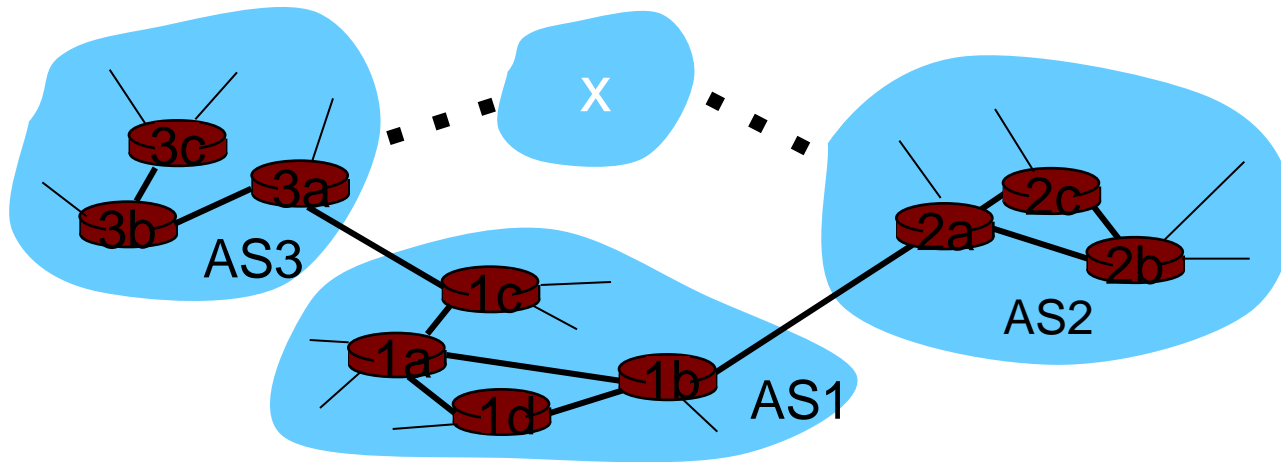
o hot potato routing: send packet towards closest of two routers.

| 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 |

Network Layer II

o 4.4 Routing algorithms

  o Link state

  o Distance Vector

  o Hierarchical routing

o 4.5 Routing protocols

  o Routing Information Protocol (RIP)

  o Open Shortest Path First (OSPF)

  o Border Gateway Protocol (BGP)

Routing Protocols

o Intra-AS routing aka Interior Gateway Protocols (IGP)

- o Routing Information Protocol (RIP)
- o Open Shortest Path First (OSPF)
- o Interior Gateway Routing Protocol (IGRP) (Cisco proprietary)

o Inter-AS routing

- o Border Gateway Protocol (BGP)
- o Exterior Gateway Protocol (EGP)

Network Layer II

- 4.4 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.5 Routing protocols
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Border Gateway Protocol (BGP)

Routing Information Protocol (RIP)

o distance vector algorithm

o included in BSD-UNIX Distribution in 1982
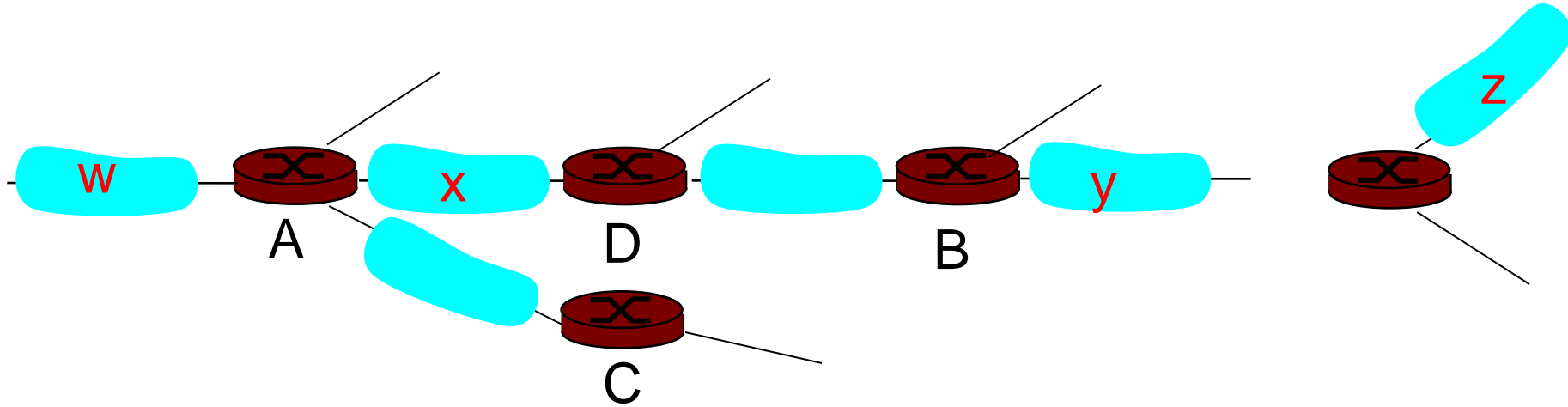
o distance metric: # of hops (max = 15 hops)
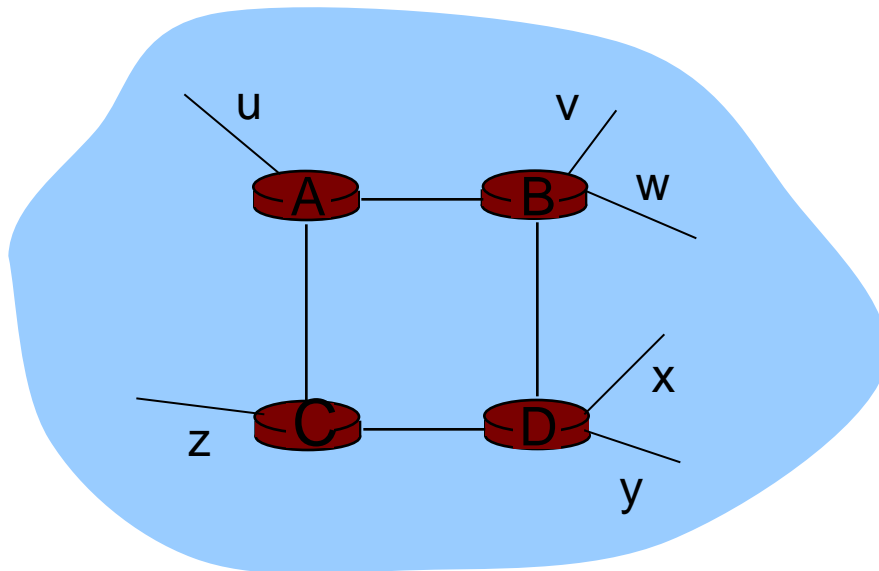
From router A to subnets:

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

RIP advertisements

o distance vectors

  o exchanged among neighbors every 30 sec via Response Message (also called *advertisement*)

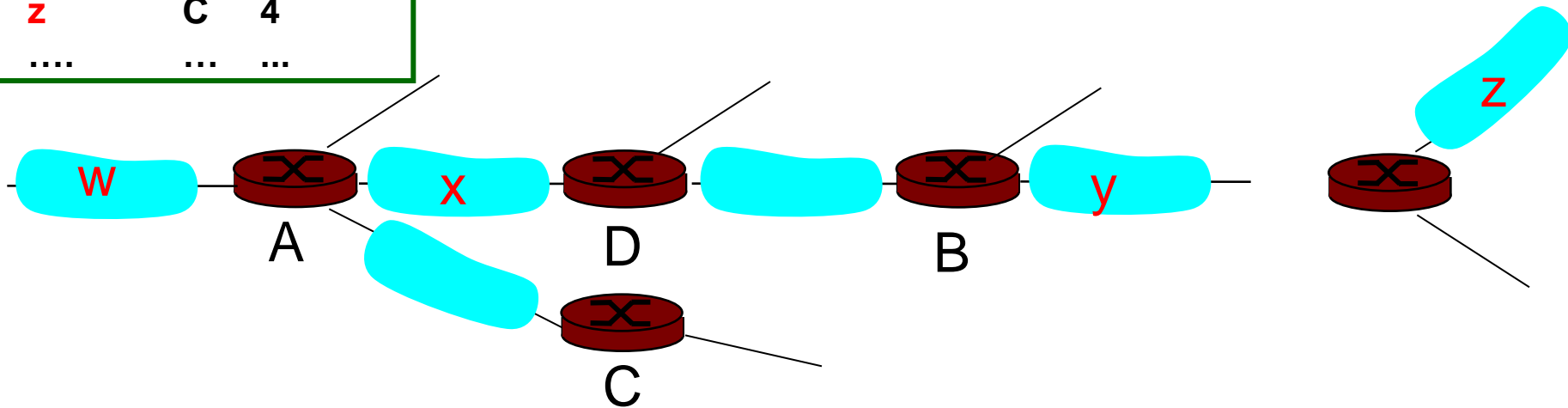o each advertisement: list of up to 25 destination subnets within AS

RIP: Example

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

Routing/Forwarding table in D

RIP: Example

Dest	Next	hops
w	-	1
x	-	1
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/Forwarding table in D

RIP: Link Failure and Recovery

o If no advertisement heard after 180 sec: neighbor/link declared dead

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

RIP Table processing

o RIP routing tables managed by **application-level** process called route-d (daemon)

o advertisements sent in UDP packets, periodically repeated
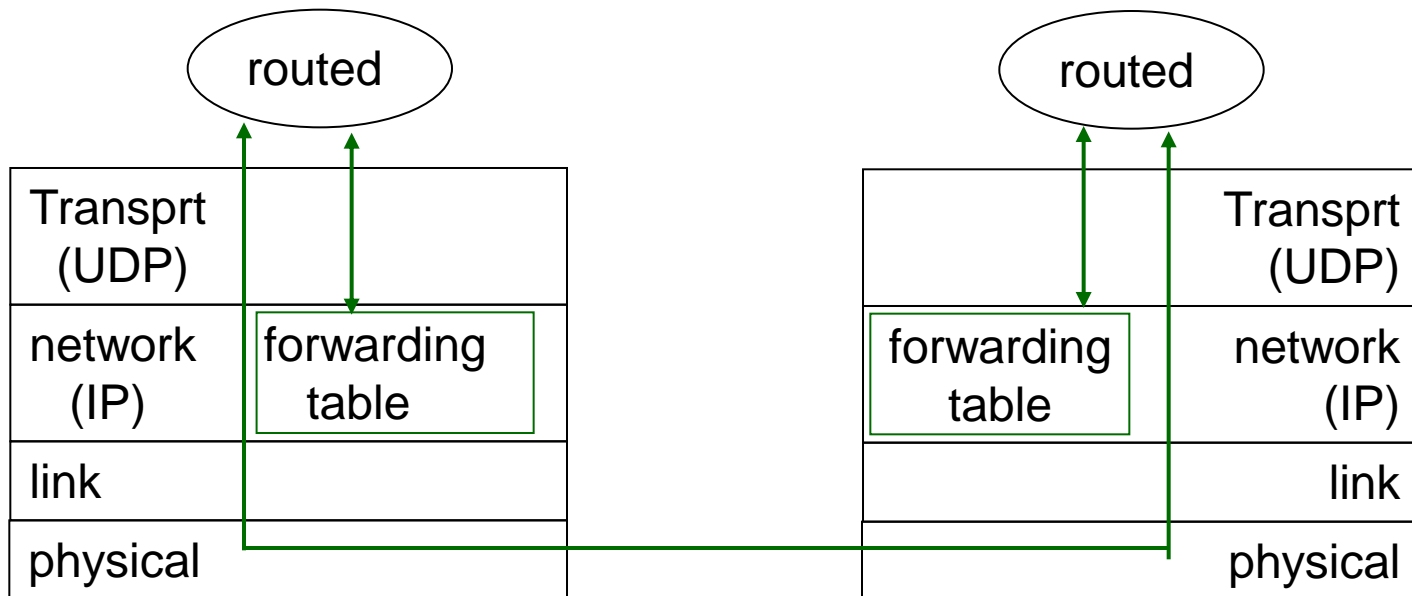
Network Layer II

- 4.4 Routing algorithms
    - Link state
    - Distance Vector
    - Hierarchical routing
- 4.5 Routing protocols
    - Routing Information Protocol (RIP)
    - Open Shortest Path First (OSPF)
   - Border Gateway Protocol (BGP)

Open Shortest Path First (OSPF)

o "open": publicly available

o uses Link State algorithm
  o LS packet dissemination
  o topology map at each node
  o route computation using Dijkstra's algorithm

o OSPF advertisement carries one entry per neighbor router

o advertisements disseminated to entire AS (via flooding)
  o carried in OSPF messages directly over IP (rather than TCP or UDP

OSPF "advanced" features (not in RIP)

o Security: all OSPF messages authenticated (to prevent malicious intrusion)

o multiple same-cost paths allowed

o For each link, multiple cost metrics for different TOS

o Integrated uni- and multicast support:

  o Multicast OSPF (MOSPF) uses same topology data base as OSPF

o Hierarchical OSPF in large domains.

Hierarchical OSPF
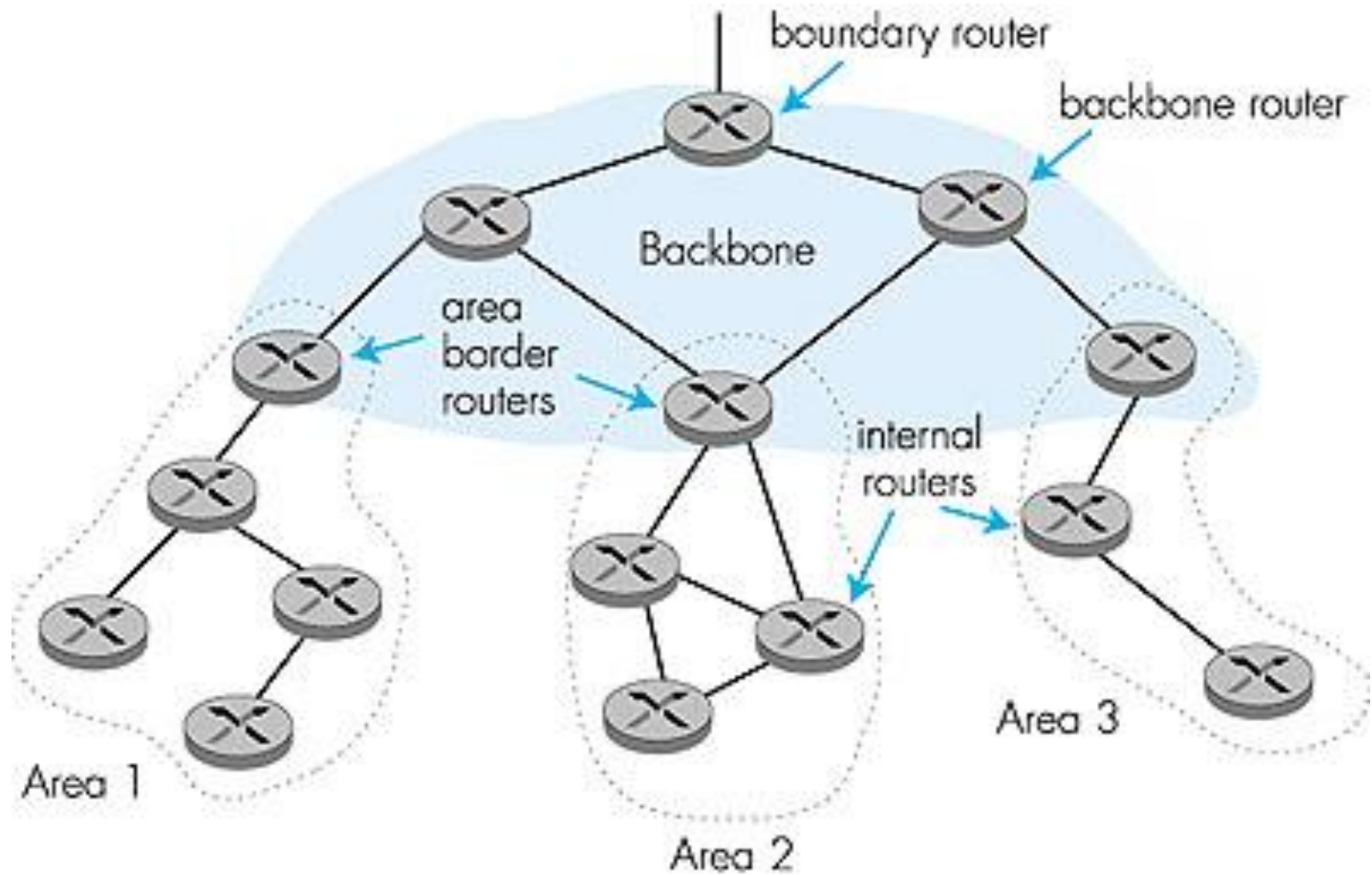
Hierarchical OSPF

o two-level hierarchy: local area, backbone.
  - o Link-state advertisements only in area
  - o each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.

o area border routers: "summarize" distances to nets in own area, advertise to other Area Border routers.

o backbone routers: run OSPF routing limited to backbone.

o boundary routers: connect to other AS's.

Network Layer II

o 4.4 Routing algorithms

  o Link state

  o Distance Vector

  o Hierarchical routing

o 4.5 Routing protocols

  o Routing Information Protocol (RIP)

  o Open Shortest Path First (OSPF)

  o Border Gateway Protocol (BGP)

Inter-AS routing: BGP

○ BGP (Border Gateway Protocol): the de facto standard

○ BGP provides each AS a means to:

  ○ Obtain subnet reachability information from neighboring ASs.

  ○ Propagate reachability information to all AS-internal routers.

  ○ Determine "good" routes to subnets based on reachability information and policy.

○ allows subnet to advertise its existence to rest of Internet: "I am here"

BGP basics

○ pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions

  ○ BGP sessions need not correspond to physical links.

○ when AS2 advertises a prefix to AS1:

  ○ AS2 *promises* it will forward datagrams towards that prefix.

  ○ AS2 can aggregate prefixes in its advertisement

- - - - - eBGP session

·············· iBGP session

3c   3a   3b   AS3

1c   1a   1b   1d   AS1

2a   2c   2b   AS2
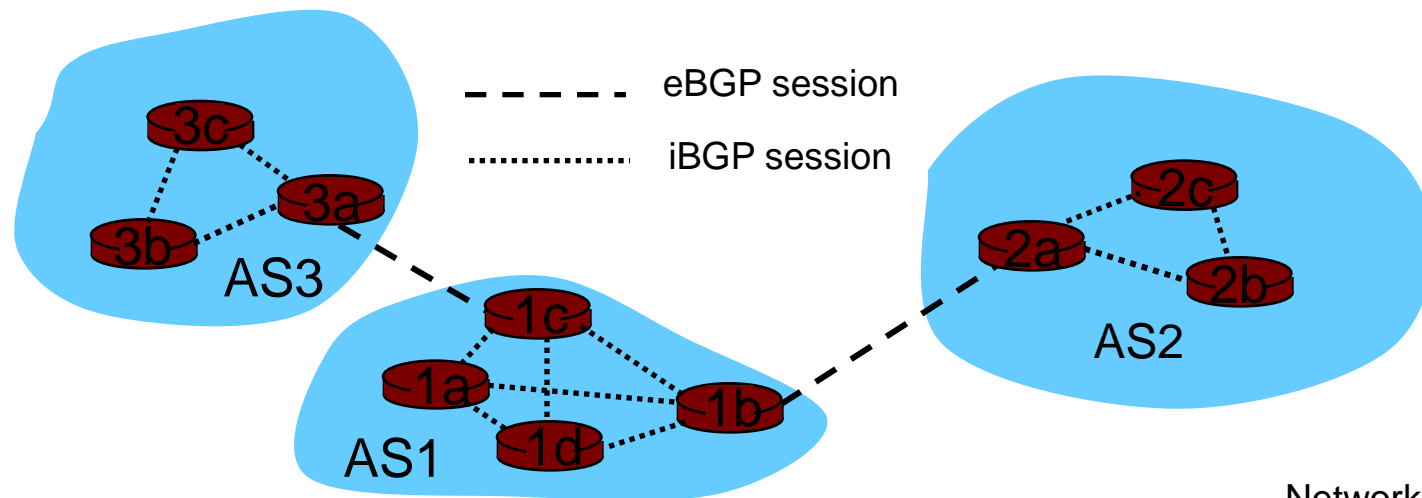
Distributing reachability info

○ using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.

    ○ 1c can then use iBGP do distribute new prefix info to all routers in AS1

    ○ 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session

○ when router learns of new prefix, it creates entry for prefix in its forwarding table.
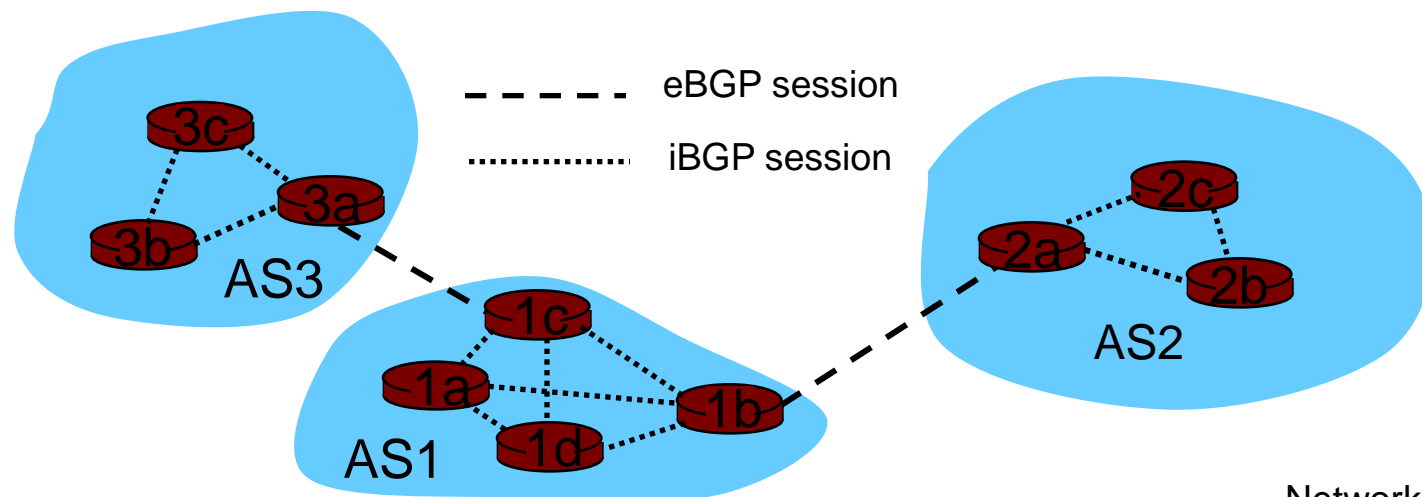
Path attributes & BGP routes

o advertised prefix includes BGP attributes.
  - o prefix + attributes = "route"
o two important attributes:
  - o AS-PATH: contains ASs through which prefix advertisement has passed: e.g, AS 67, AS 17
  - o NEXT-HOP: indicates specific internal-AS router to next-hop AS. (may be multiple links from current AS to next-hop-AS)
o when gateway router receives route advertisement, uses import policy to accept/decline.

BGP route selection

o Router may learn about more than 1 route to same prefix:

  o Router must select route

o Elimination rules:

  o Local preference value attribute: policy decision

  o Shortest AS-PATH

  o Closest NEXT-HOP router: hot potato routing

  o Additional criteria

BGP messages

o BGP messages exchanged using TCP

o BGP messages:

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

BGP routing policy

legend:

provider network
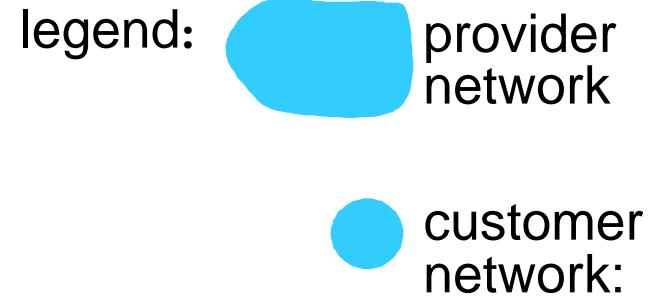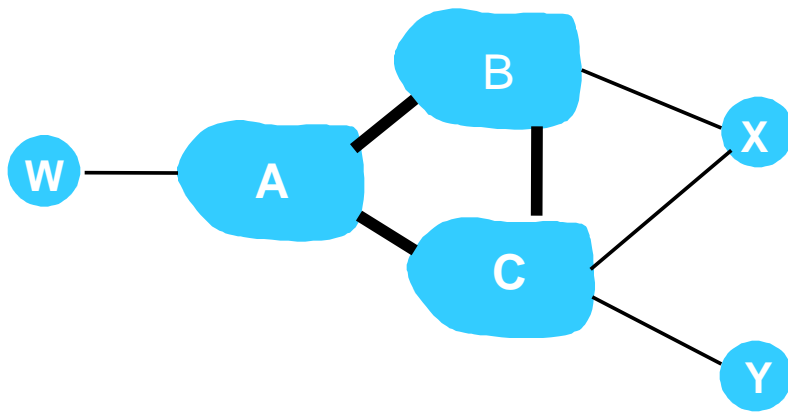
customer network:

○ A,B,C are provider networks

○ X,W,Y are customer 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 routing policy (2)

legend:

provider
network
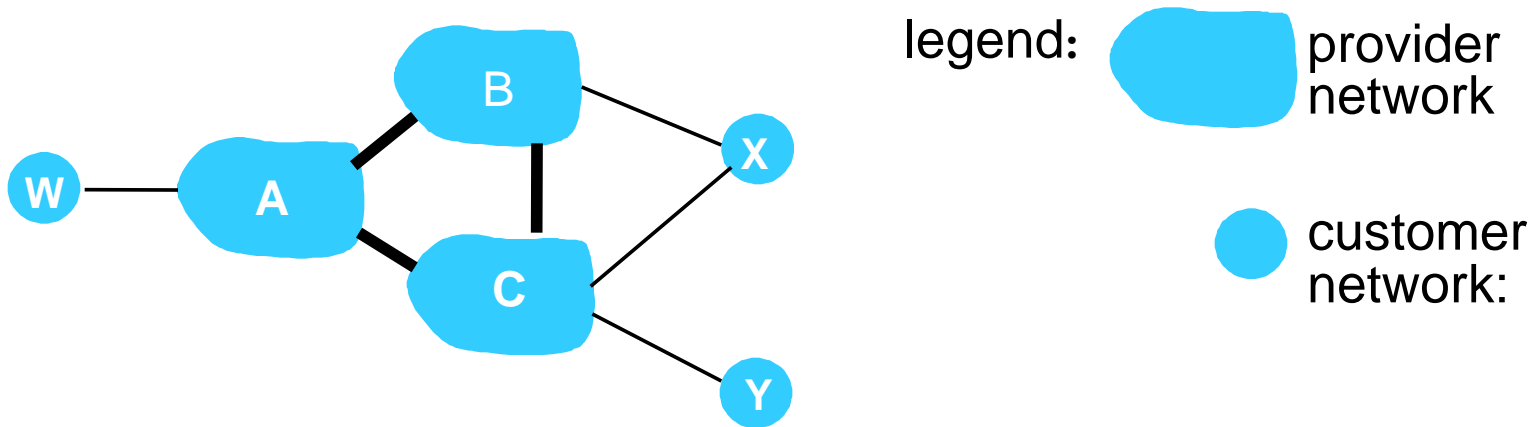
customer
network:

- A advertises path AW to B

- B advertises path BAW to X

- Should B advertise path BAW to C?

  - 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

Thank you

Any questions?