Chapter 4: outline

Interplay between routing, forwarding

routing algorithm

routing algorithm determines
end-end-path through network

forwarding table determines
local forwarding at this router

local forwarding table

dest address	output link
address-range 1	3
address-range 2	2
address-range 3	2
address-range 4	1

IP destination address in
arriving packet's header

1

3 2
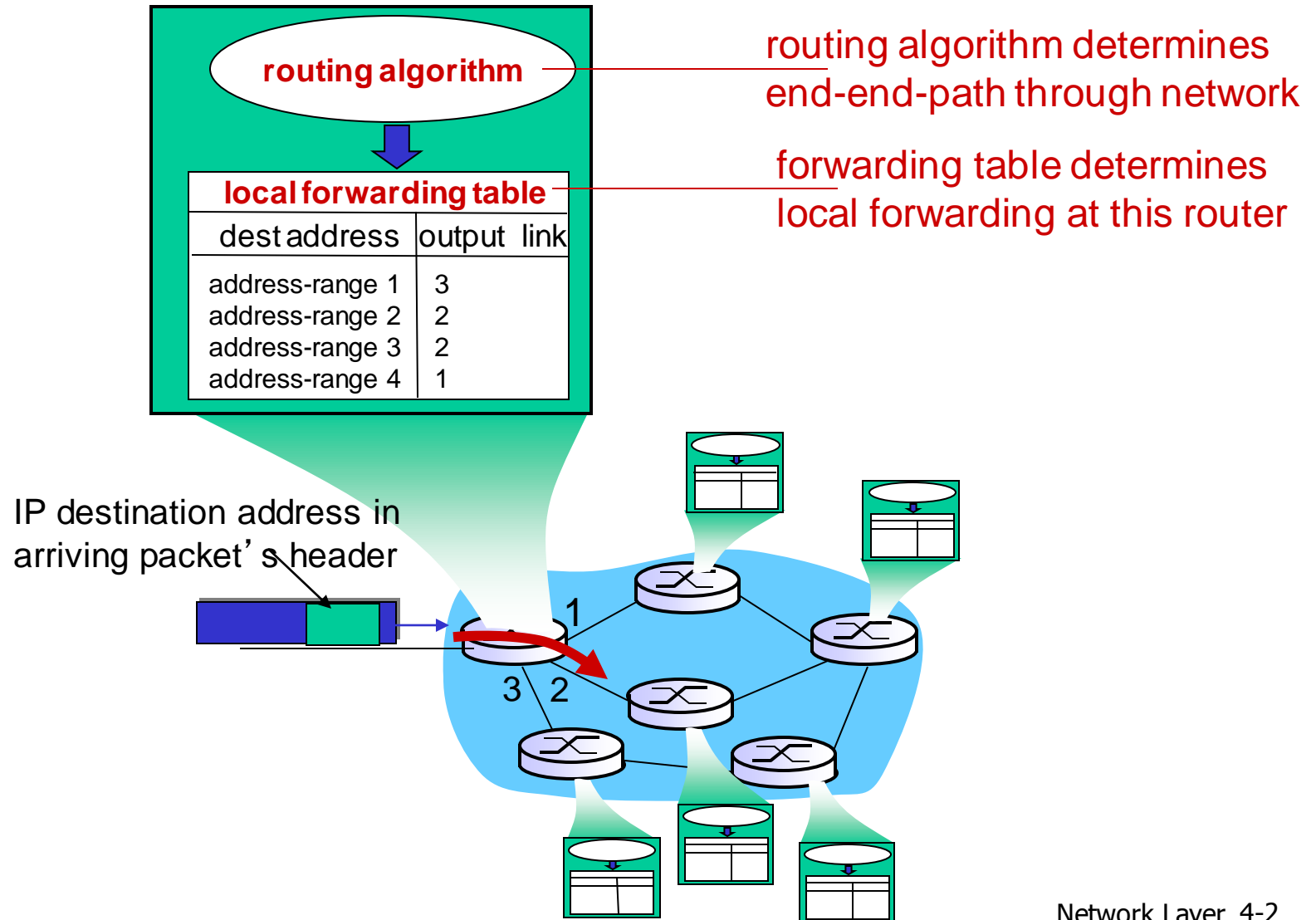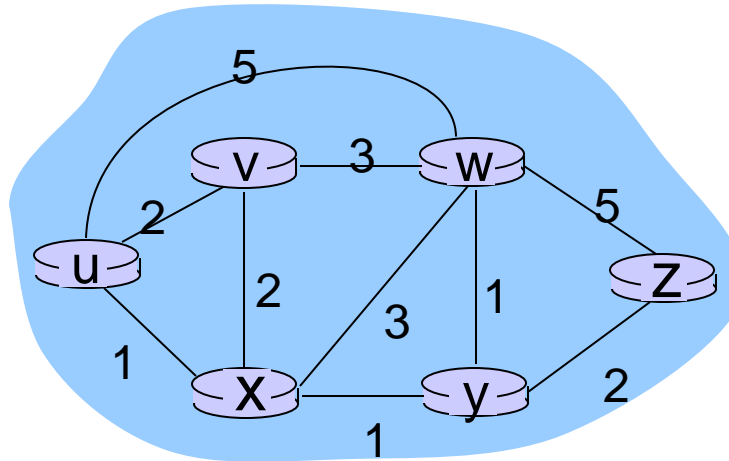
Graph abstraction

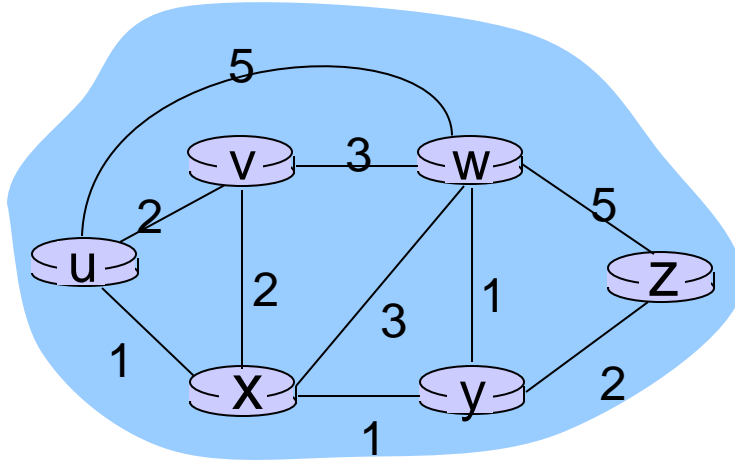

graph: G = (N,E)

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) }

aside: graph abstraction is useful in other network contexts, e.g., 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 $(x_1, x_2, x_3, ..., x_p) = c(x_1,x_2) + c(x_2,x_3) + ... + c(x_{p-1},x_p)$

key question: what is the least-cost path between 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

Q: static or dynamic?

static:

- ❖ routes change slowly over time

dynamic:

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

Chapter 4: outline

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

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 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' :
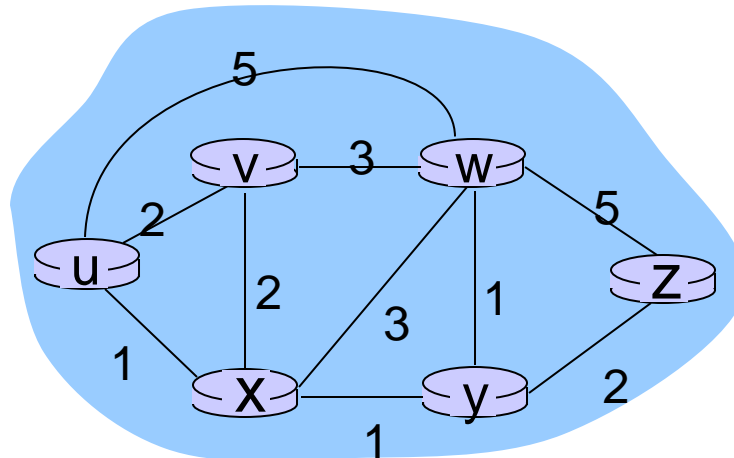12   **D(v) = min(D(v), D(w) + c(w,v))**
13   /* new cost to v is either old cost to v or known
14   shortest path cost to w plus cost from w to v */
15 *until all nodes in N'*
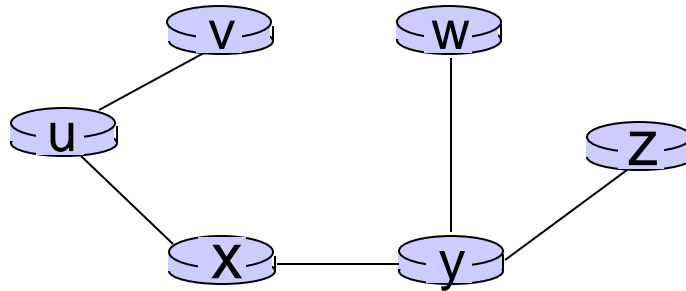
Dijkstra's algorithm: another example

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)

Chapter 4: outline

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

Bellman-Ford equation (dynamic programming)

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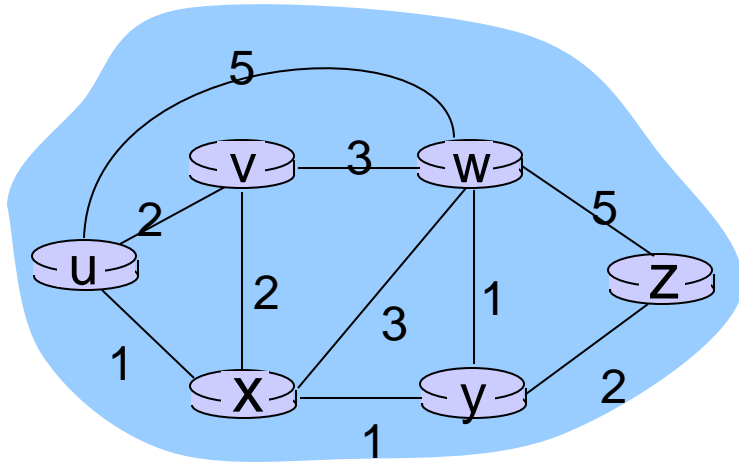then

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

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

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 achieving minimum is next
hop in shortest path, used in forwarding table

Distance vector algorithm

❖ $D_x(y)$ = estimate of least cost from x to y
  ▪ x maintains distance vector $D_x = [D_x(y): y \in N]$
❖ node x:
  ▪ knows cost to each neighbor v: $c(x,v)$
  ▪ maintains its neighbors' distance vectors. For each neighbor v, x maintains
    $D_v = [D_v(y): y \in N]$

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 B-F equation:

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

* ❖ under minor, natural conditions, the estimate $D_x(y)$ *converge to the actual least cost* $d_x(y)$
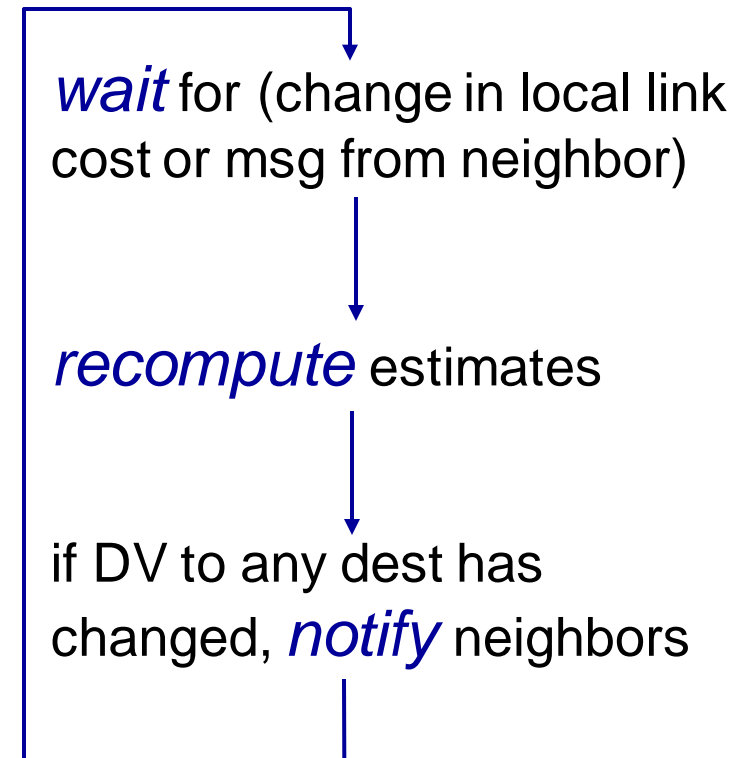
Distance vector algorithm

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
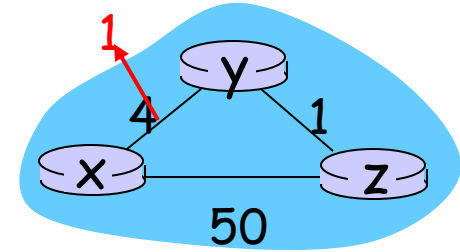 ▪ neighbors then notify their neighbors if necessary

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: 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"

t_0: *y* detects link-cost change, updates its DV, informs its neighbors.

t_1: *z* receives update from *y*, updates its table, computes new least cost to *x*, sends its neighbors its DV.

t_2: *y* receives *z*'s update, updates its distance table. *y*'s least costs do *not* change, so *y* does *not* send a message to *z*.

Distance vector: link cost changes

link cost changes:

❖ node detects local link cost change

❖ *bad news travels slow* - "count to infinity" problem!

❖ 44 iterations before algorithm stabilizes: see text
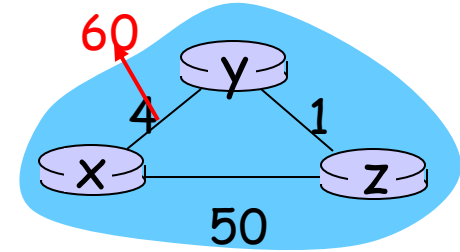
poisoned reverse:

❖ 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(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 propagate thru network

Chapter 4: outline

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

Hierarchical routing

our routing study thus far - idealization
- ❖ all routers identical
- ❖ network "flat"

… *not* true in practice

scale: with 600 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 router:

❖ at "edge" of its own AS

❖ has link to router in another AS

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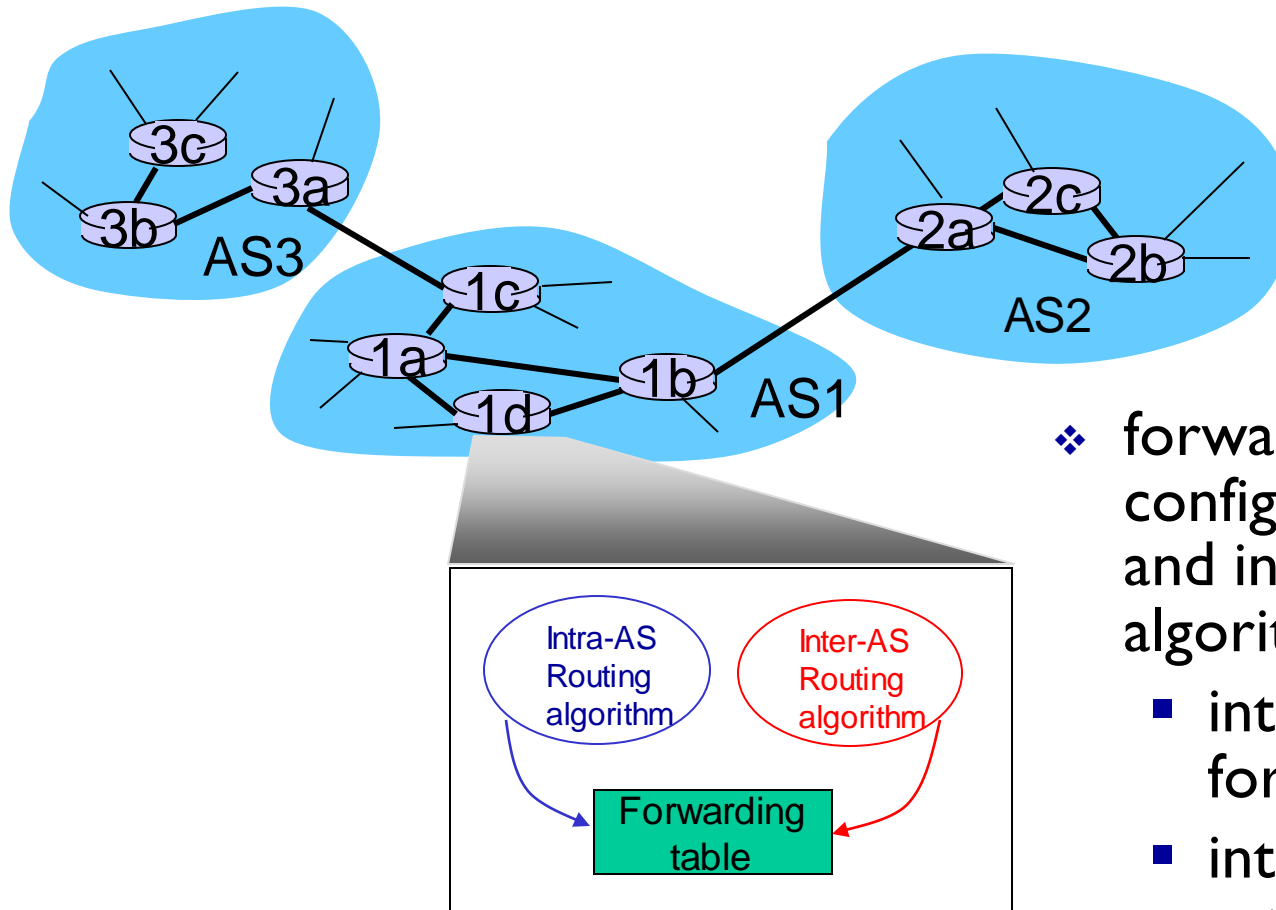
- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
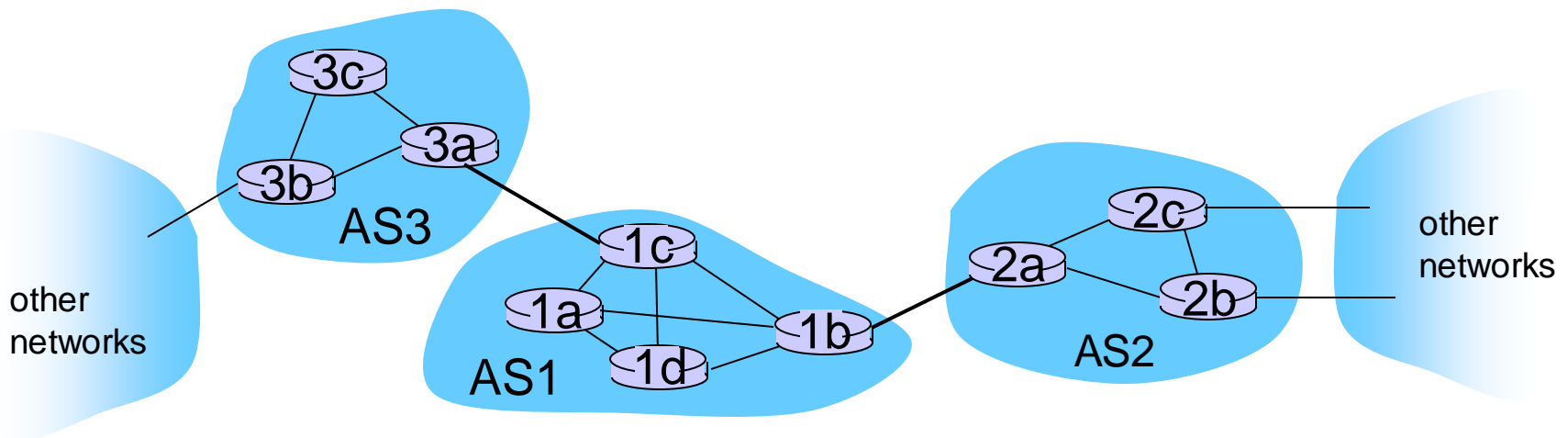 - inter-AS & intra-AS sets entries for external dests

Inter-AS tasks

❖ 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!

Chapter 4: outline

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

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)

RIP (Routing Information Protocol)

- ❖ included in BSD-UNIX distribution in 1982
- ❖ distance vector algorithm
  - ▪ distance metric: # hops (max = 15 hops), each link has cost 1
  - ▪ DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - ▪ each advertisement: list of up to 25 destination *subnets* (in IP addressing sense)
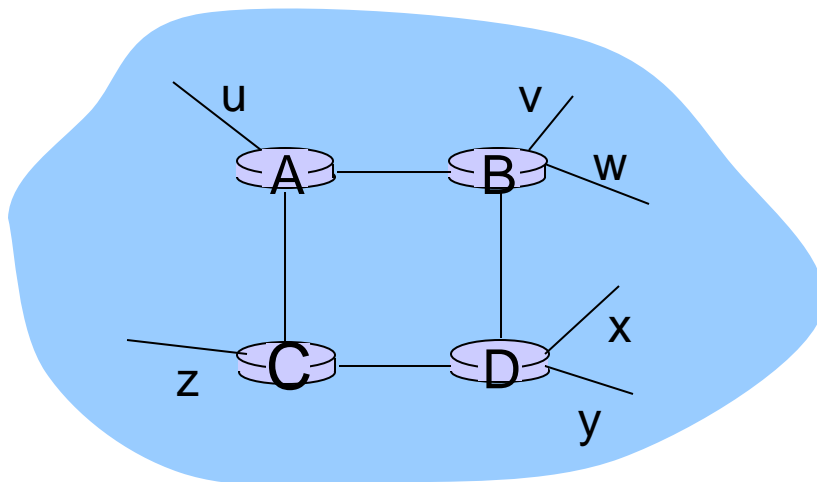
from router A to destination *subnets:*

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

RIP: example
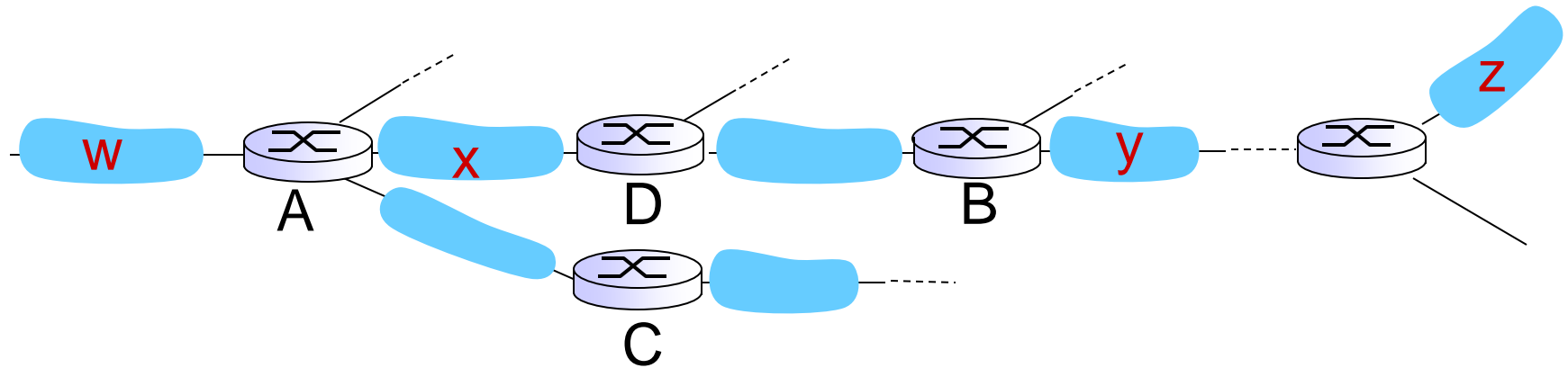
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

RIP: example

A-to-D advertisement

dest	next	hops
w	-	1
x	-	1
z	C	4
....	....	....
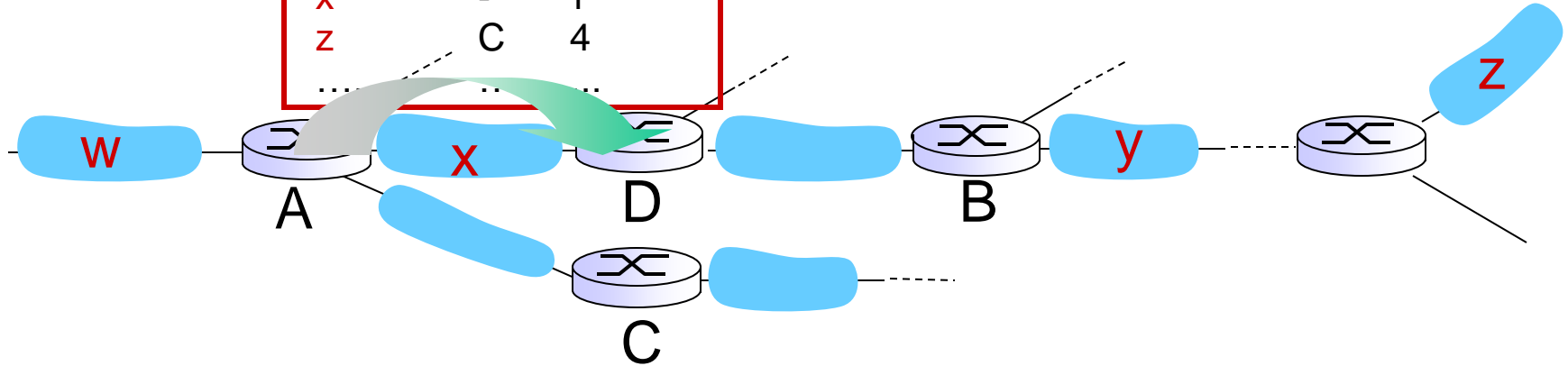
routing table in router D

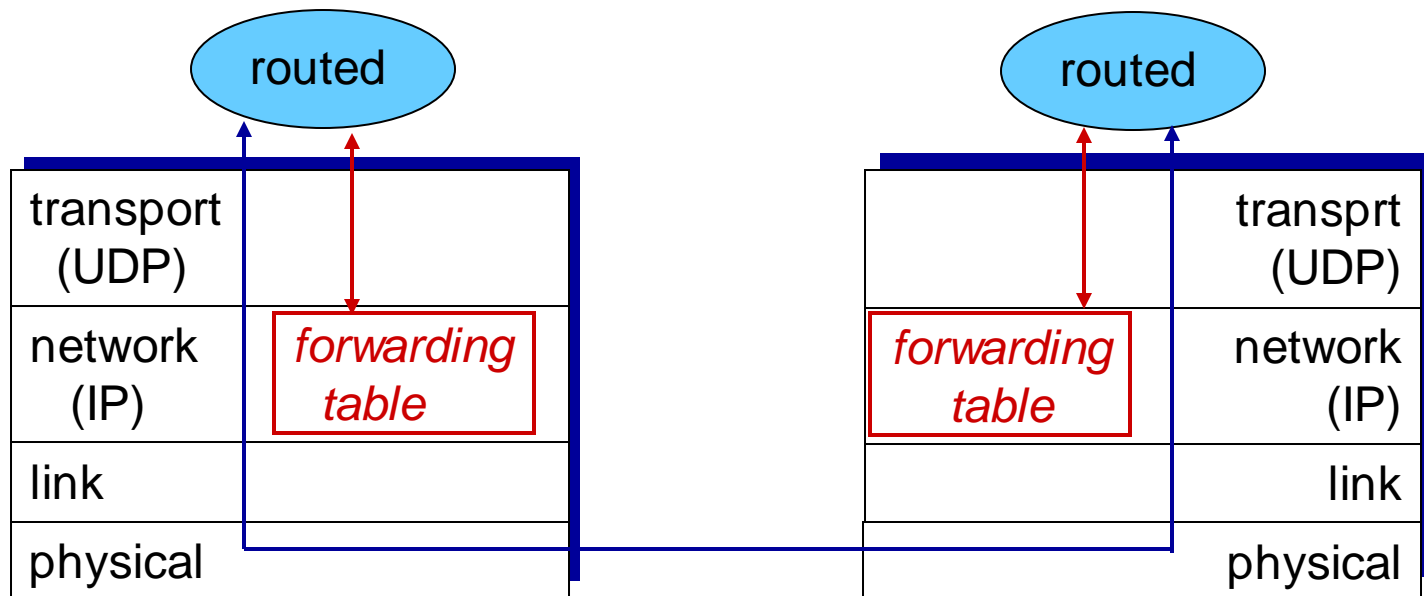| destination subnet	next router	# hops to dest
w	A	2
y	B	2 5
z	B A	7
x	--	1
....	....	....

RIP: link failure, 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)
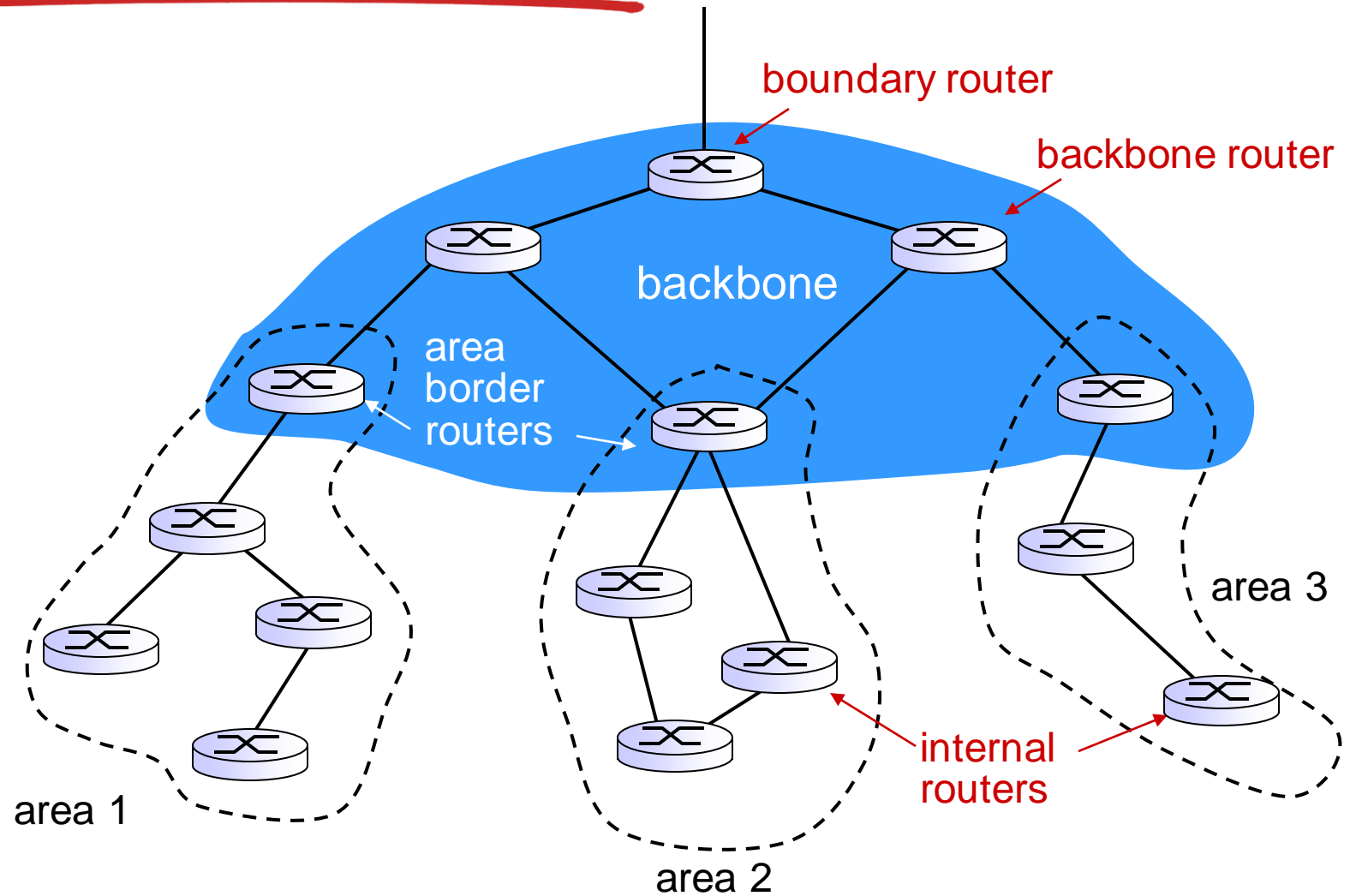❖ advertisements sent in UDP packets, periodically repeated

routed

routed

transport (UDP)	
network (IP)	*forwarding table*
link	
physical	

	transprt (UDP)
forwarding table	network (IP)
	link
	physical

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
* ❖ advertisements flooded to *entire* AS
    * ▪ carried in OSPF messages directly over IP (rather than TCP or UDP
* ❖ *IS-IS routing* protocol: nearly identical to OSPF

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 ToS; high for real time ToS)

❖ integrated uni- and multicast support:
   ▪ Multicast OSPF (MOSPF) uses same topology data base as OSPF

❖ hierarchical OSPF in large domains.

Hierarchical OSPF

boundary router

backbone router

backbone

area border routers

internal routers

area 1

area 2

area 3

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.

Internet inter-AS routing: BGP

❖ BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  ▪ "glue that holds the Internet together"
❖ BGP provides each AS a means to:
  ▪ eBGP: obtain subnet reachability information from neighboring ASs.
  ▪ iBGP: propagate reachability information to all AS-internal routers.
  ▪ determine "good" routes to other networks based on reachability information and policy.
❖ allows subnet to advertise its existence to rest of Internet: *"I am here"*

BGP basics

❖ **BGP session:** two BGP routers ("peers") exchange BGP messages:
- advertising *paths* to different destination network prefixes ("path vector" protocol)
- exchanged over semi-permanent TCP connections

❖ when AS3 advertises a prefix to AS1:
- AS3 *promises* it will forward datagrams towards that prefix
- AS3 can aggregate prefixes in its advertisement
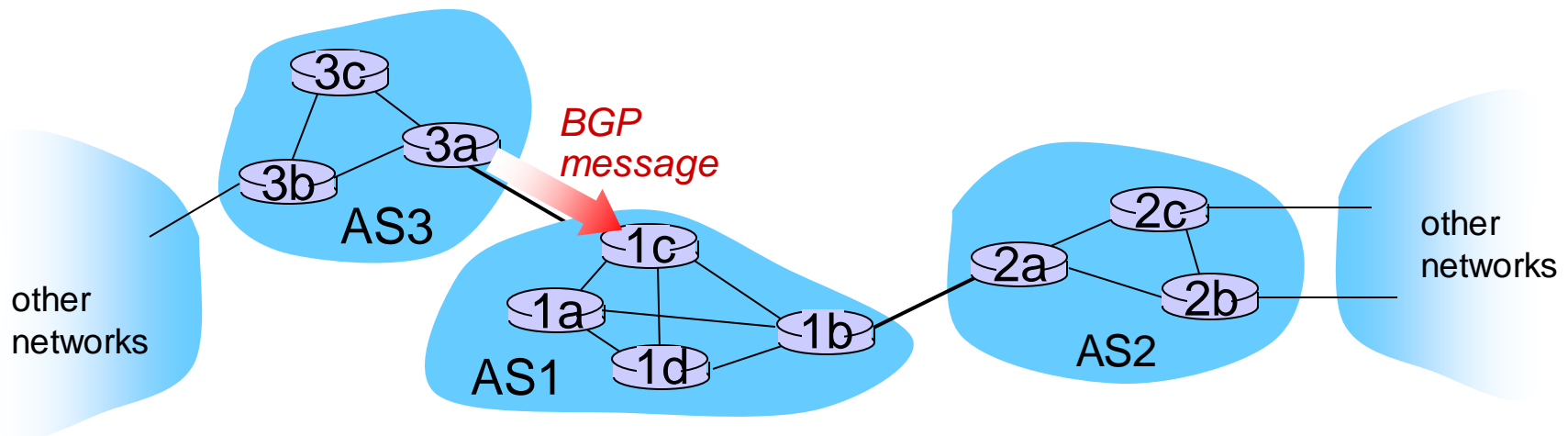
BGP basics: distributing path information

❖ 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.

eBGP session

iBGP session
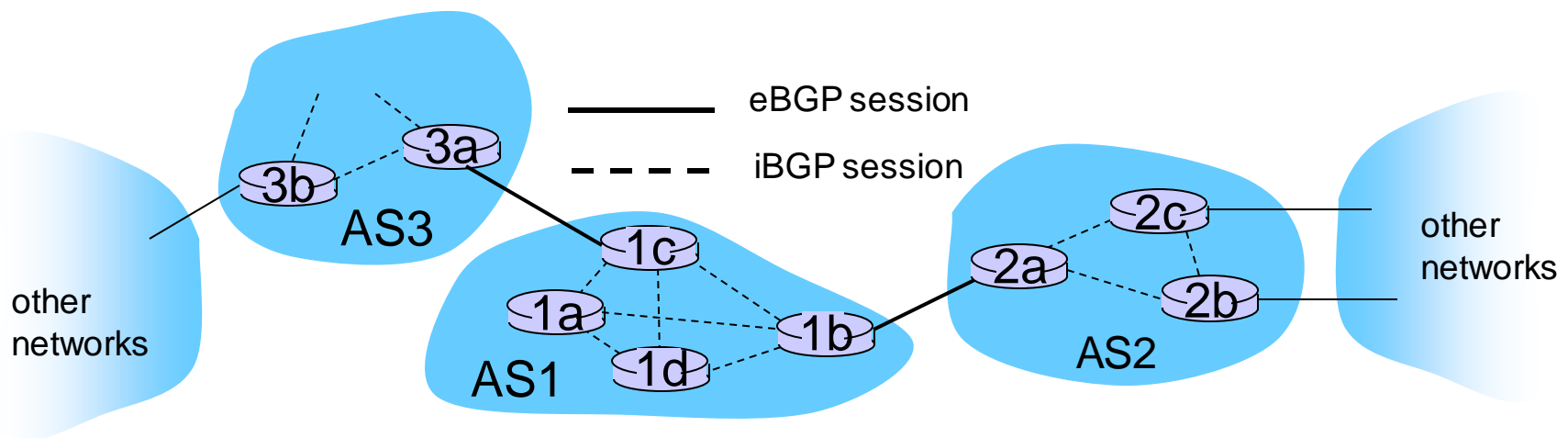
Why different Intra-, Inter-AS routing ?

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