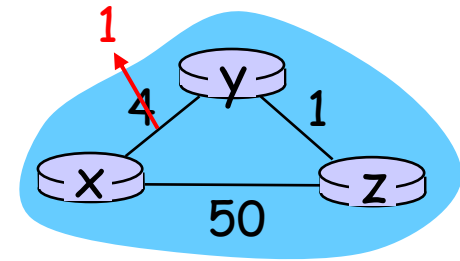# CS 3205 COMPUTER NETWORKS

JAN-MAY 2020

LECTURES 25: 23RD APR 2020

Text book and section(s) covered in this lecture:
Book Kurose and Ross – Sections 4.5

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
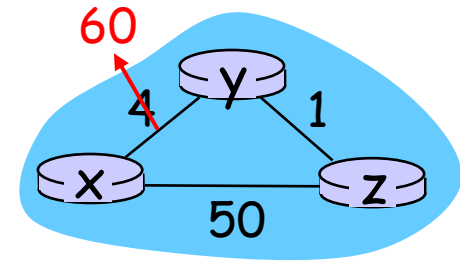
poisoned reverse:

- 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(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 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
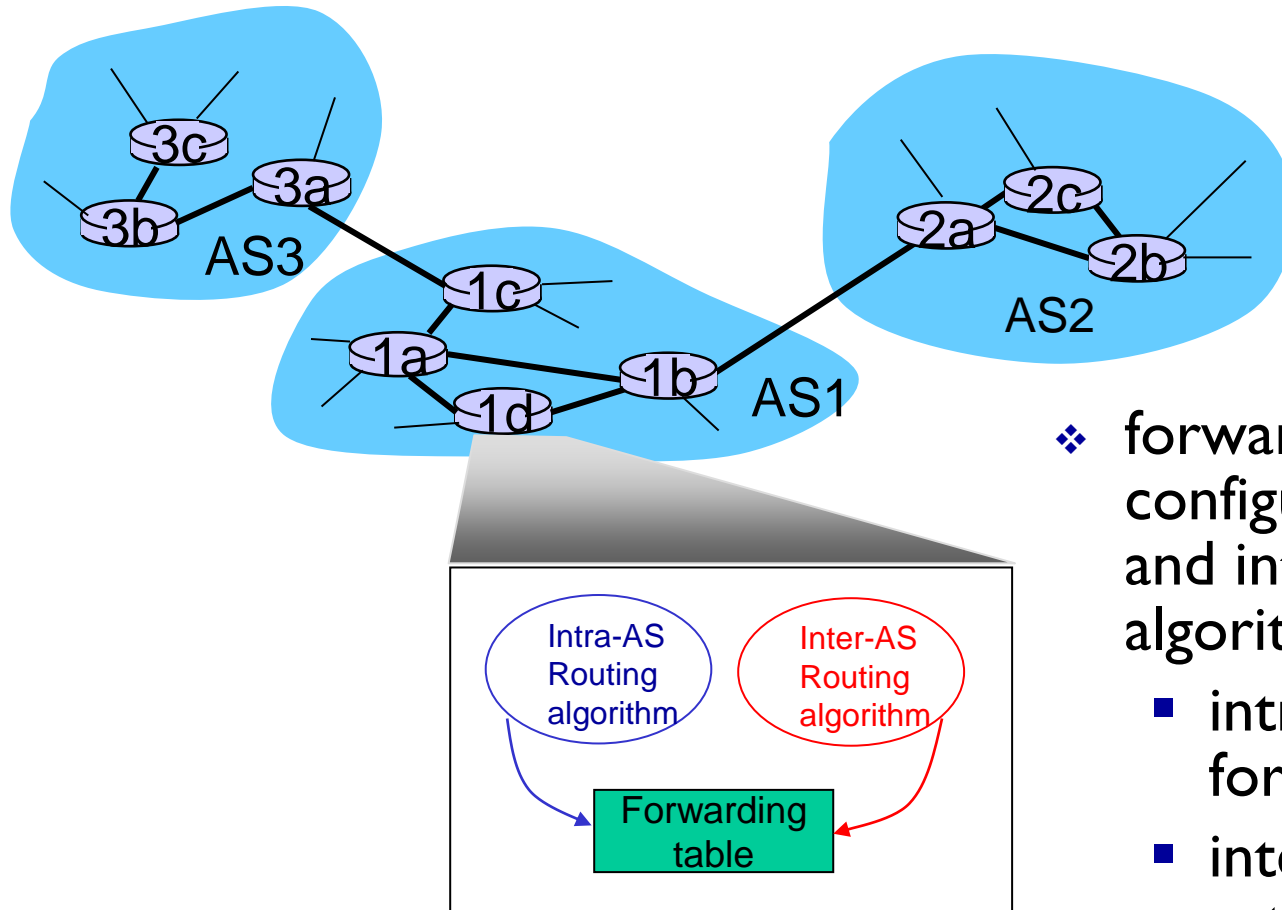
- ❖ forwarding table configured by both intra- and inter-AS routing algorithm
  - ▪ intra-AS sets entries for internal dests
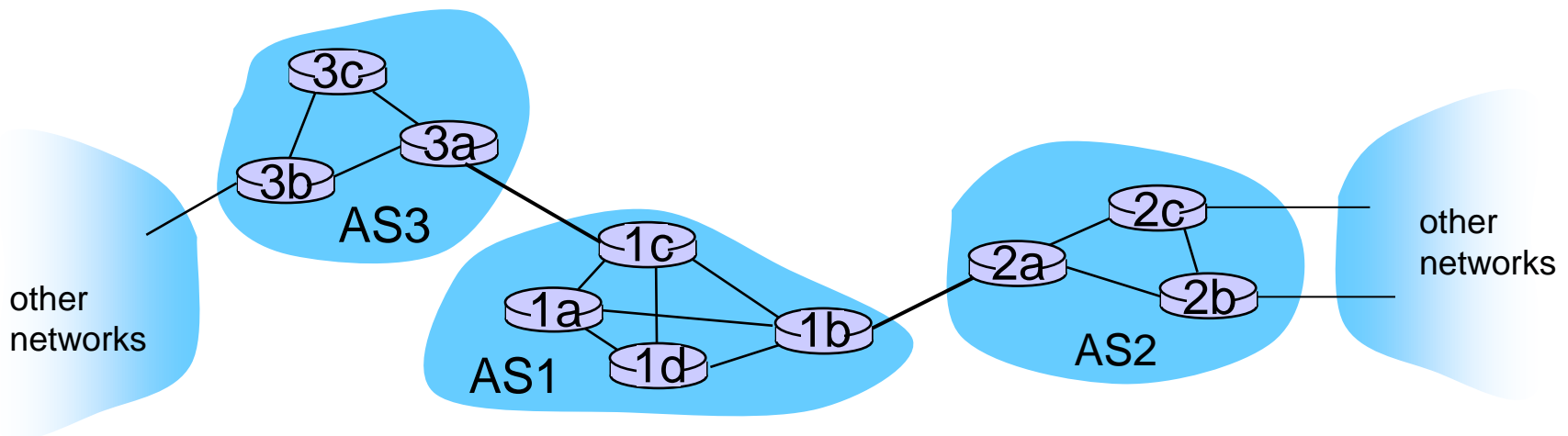 - ▪ inter-AS & intra-AS sets entries for external dests

Inter-AS tasks

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

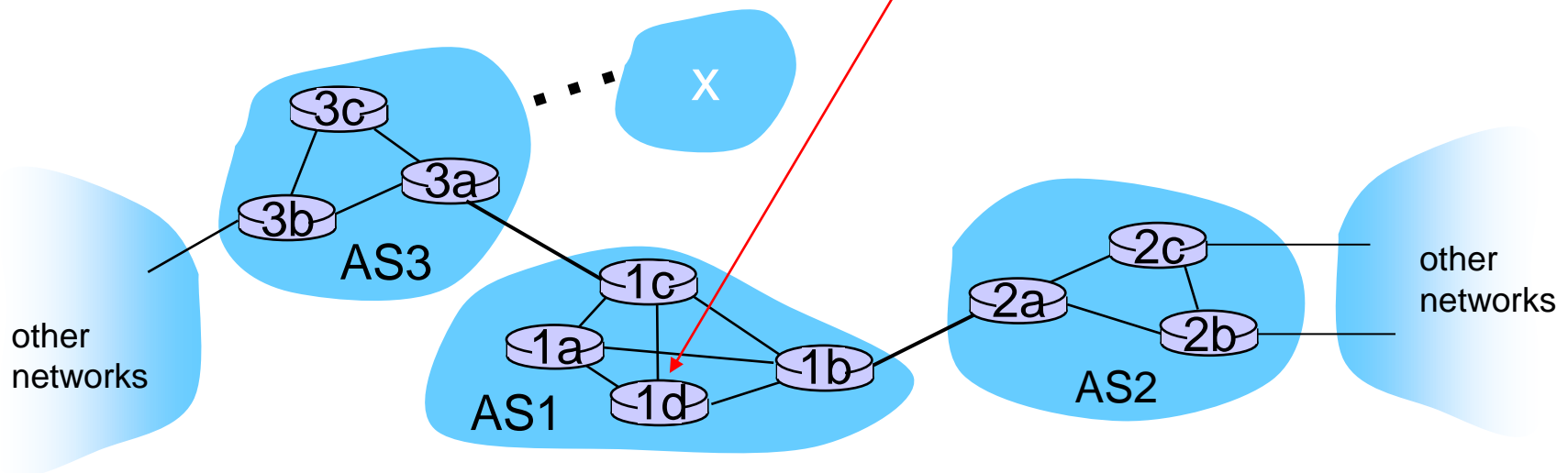- router should forward packet to gateway router, but which one?

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!

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

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

❖ to configure forwarding table, router 1d must determine which gateway it should forward packets towards for dest x
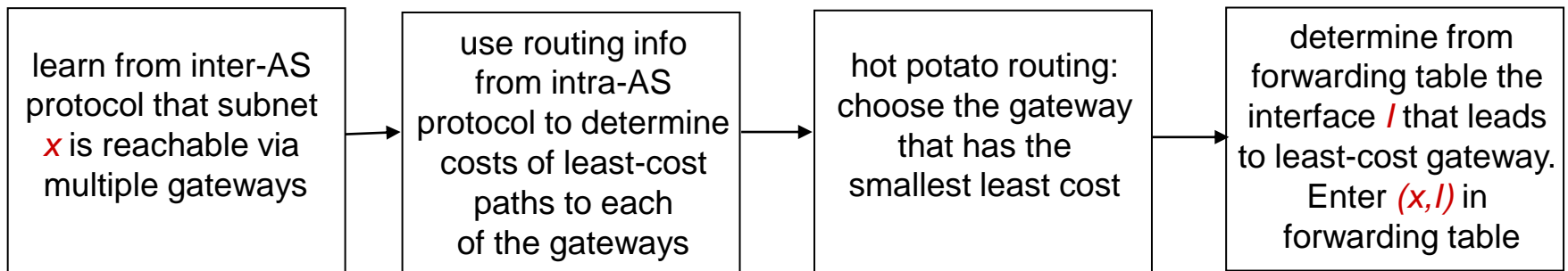 ▪ this is also job of inter-AS routing protocol!
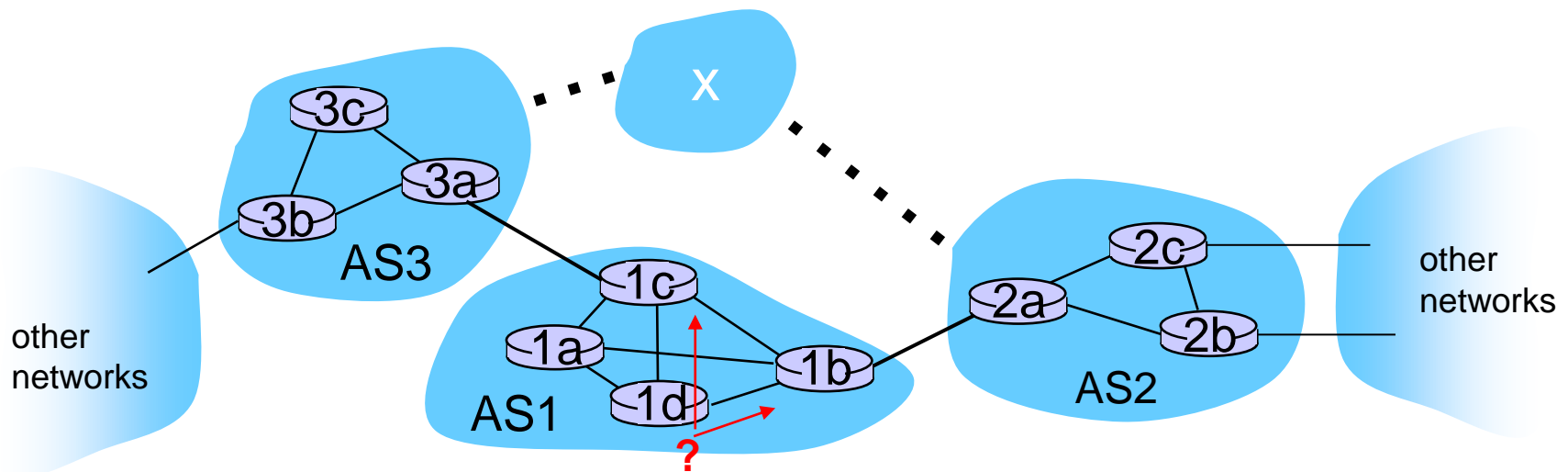
Example: choosing among multiple ASes

❖ now suppose AS1 learns from 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 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: 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)
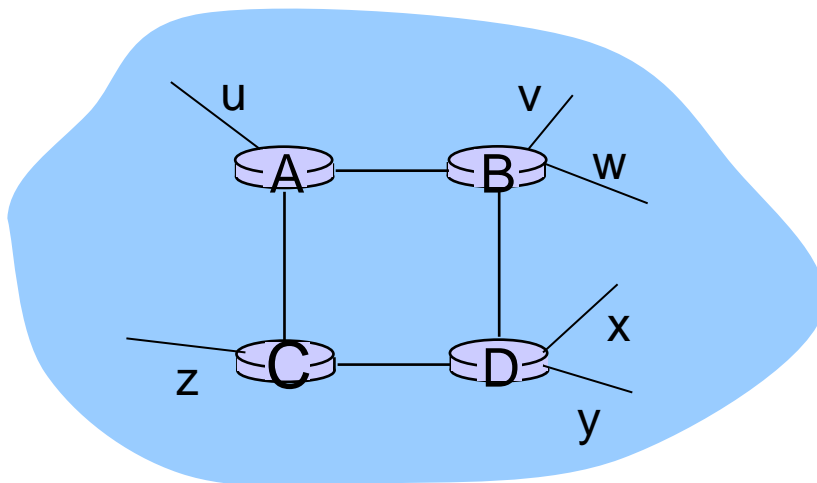
from router A to destination *subnets:*

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

RIP: example
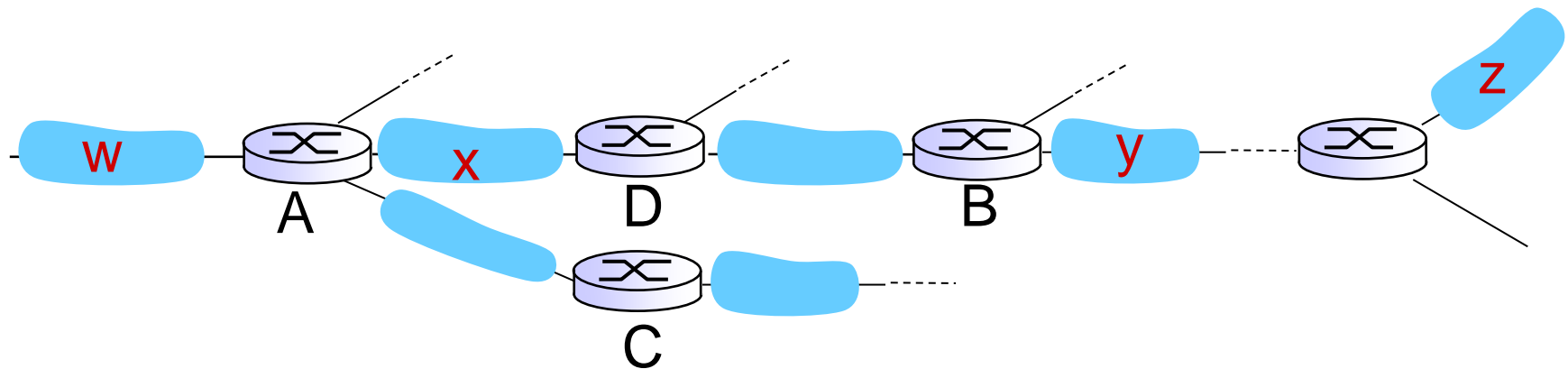
routing table in router D

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
....	....	....

W   A   x   D   B   y   z

C
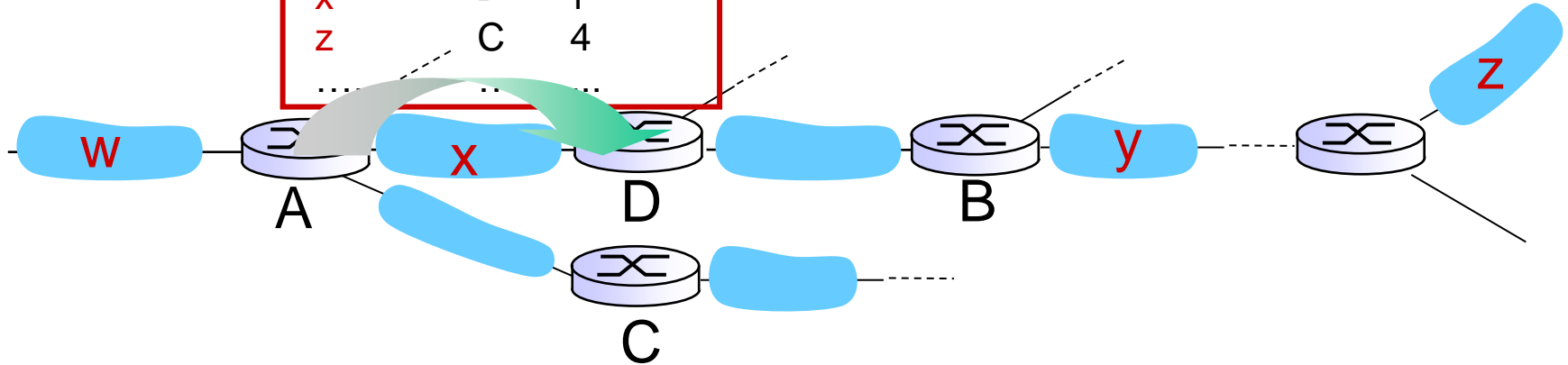
routing table in router D

| destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B A	7 5
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)

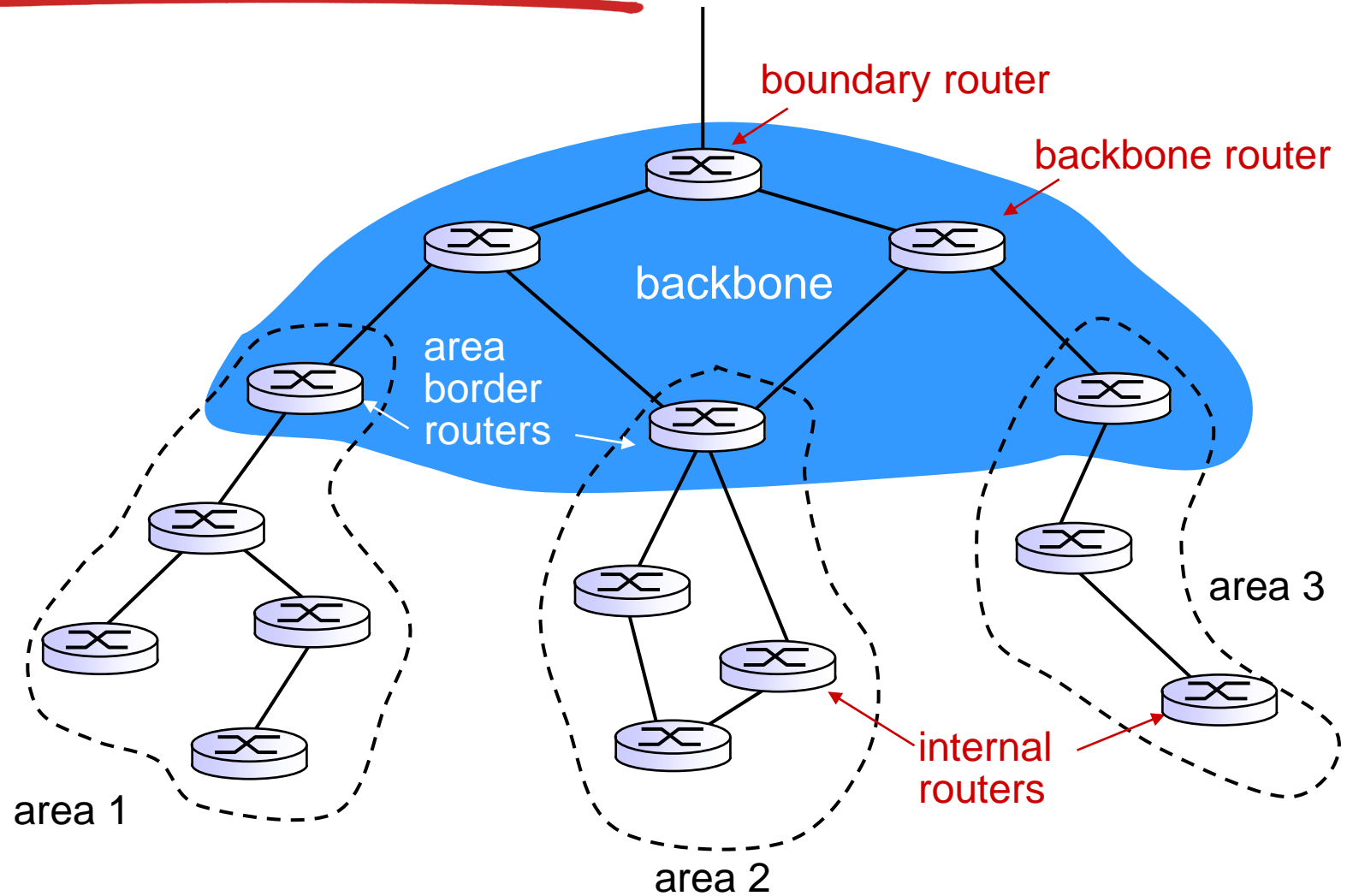❖ advertisements sent in UDP packets, periodically repeated
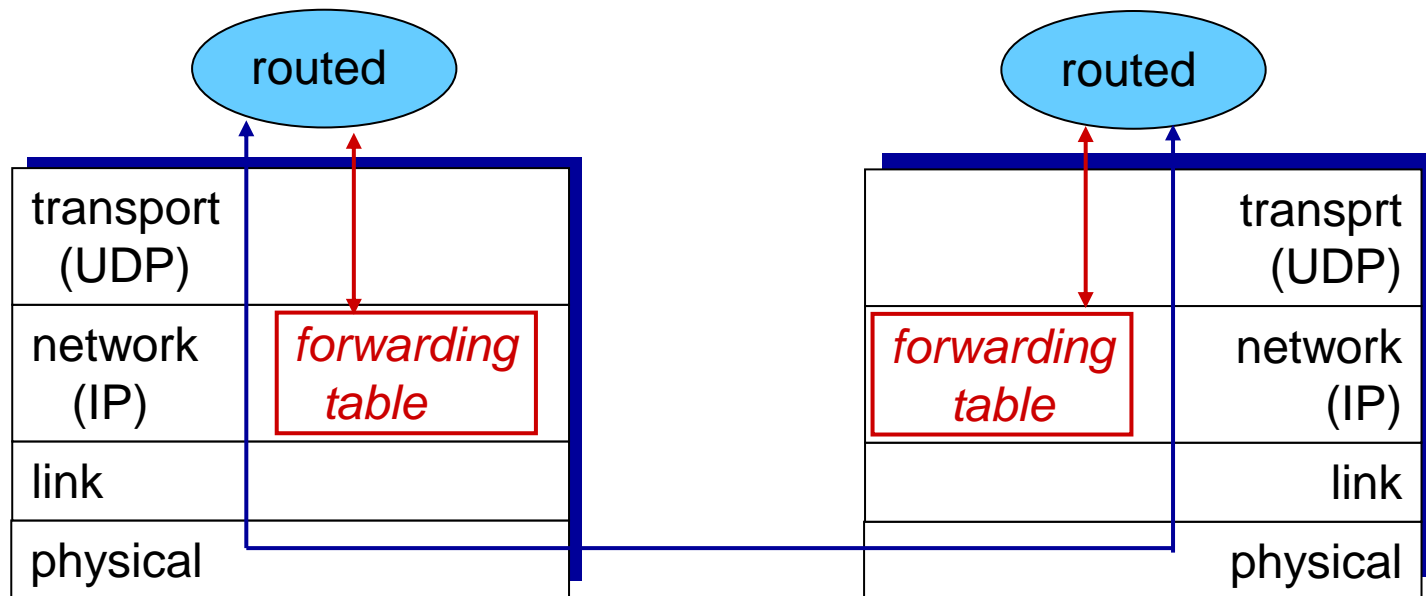
OSPF (Open Shortest Path First)

❖ "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.