

# Chapter 4

## Network Layer

### A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- ❖ If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- ❖ If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

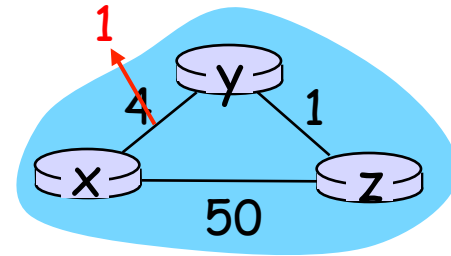
All material copyright 1996-2010  
J.F Kurose and K.W. Ross, All Rights Reserved

*Computer Networking:  
A Top Down Approach  
5<sup>th</sup> edition.  
Jim Kurose, Keith Ross  
Addison-Wesley, April  
2009.*

# 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



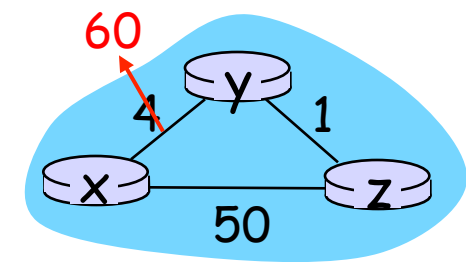
“good  
news  
travels  
fast”

$t_0$ : 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 increases



## node x table

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

## node y 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	4	5
	y	4	0	1
	z	5	1	0

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

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

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

		cost to		
		x	y	z
from	x	idem		
	y	idem		
	z	idem		

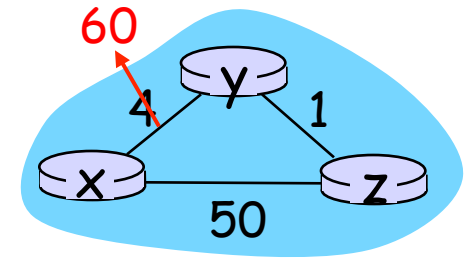
		cost to		
		x	y	z
from	x	0	51	50
	y	6	0	1
	z	5	1	0

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

		cost to		
		x	y	z
from	x	0	4	5
	y	8	0	1
	z	7	1	0

time →

Same with poison reverse!



node x table

		cost to		
		x	y	z
from	x	0	4	5
	y	$\infty$	0	1
	z	5	1	0

node y table

		cost to		
		x	y	z
from	x	0	$\infty$	$\infty$
	y	4	0	1
	z	$\infty$	$\infty$	0

node z table

		cost to		
		x	y	z
from	x	0	4	5
	y	4	0	$\infty$
	z	5	1	0

		cost to		
		x	y	z
from	x	0	51	50
	y	$\infty$	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x	0	$\infty$	$\infty$
	y	60	0	1
	z	$\infty$	$\infty$	0

		cost to		
		x	y	z
from	x	0	4	5
	y	4	0	$\infty$
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	idem		
	z			

		cost to		
		x	y	z
from	x			
	y	idem		
	z			

		cost to		
		x	y	z
from	x	0	$\infty$	$\infty$
	y	60	0	1
	z	50	1	0

		cost to		
		x	y	z
from	x	0	51	50
	y	51	0	1
	z	50	$\infty$	0

time

# 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: 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

# 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

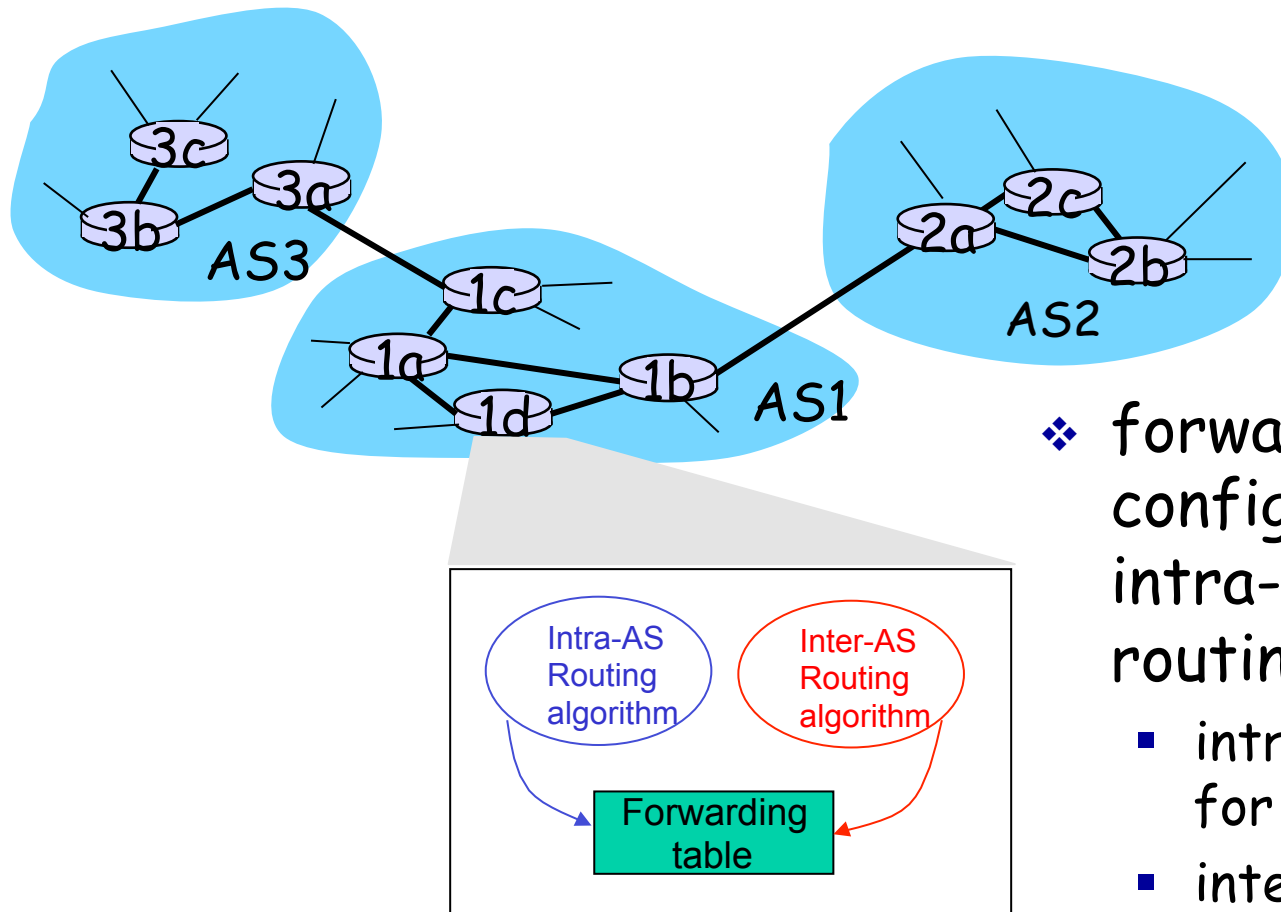
- ❖ aggregate routers into regions, “**autonomous systems**” (AS)
- ❖ routers in same AS run same routing protocol
  - “**intra-AS**” routing protocol
  - routers in different AS can run different intra-AS routing protocol

## gateway 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

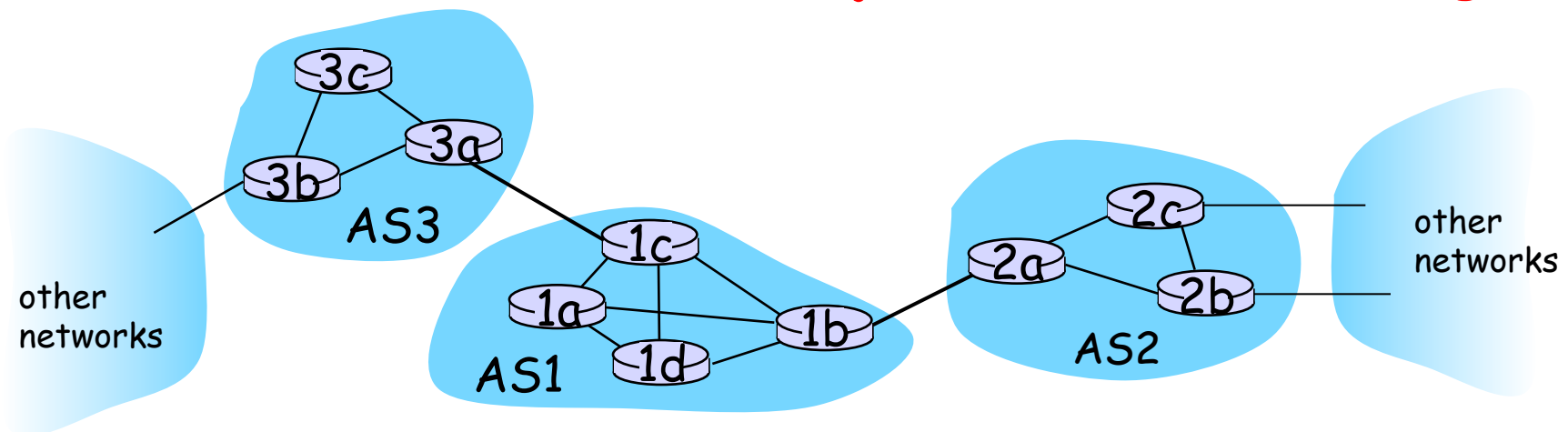
# Inter-AS tasks

- ❖ suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

## AS1 must:

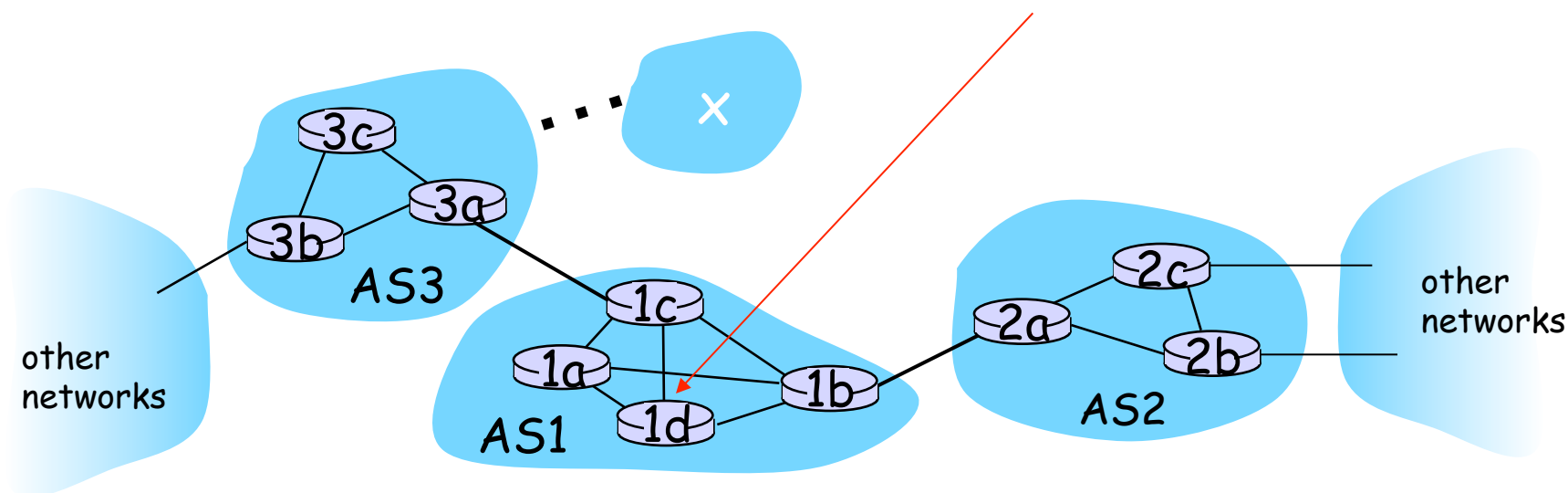
1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1

job of inter-AS routing!



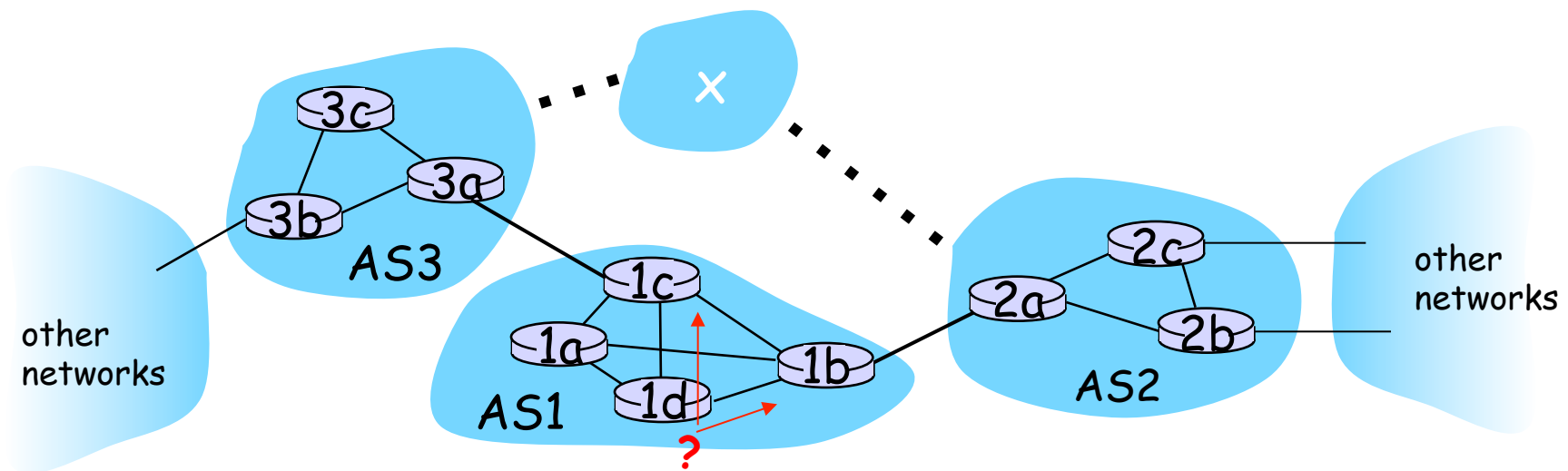
## Example: Setting forwarding table in router 1d

- ❖ 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  $(x, I)$



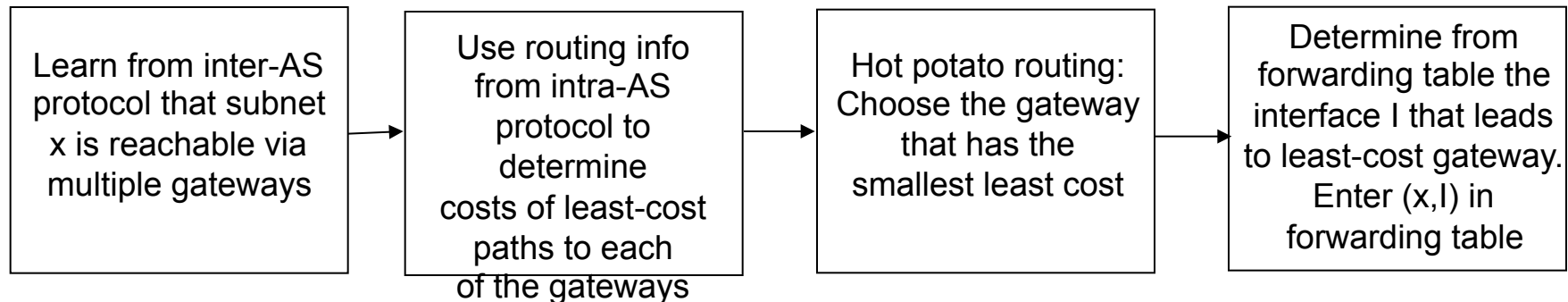
# Example: Choosing among multiple ASes

- ❖ 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.



# 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

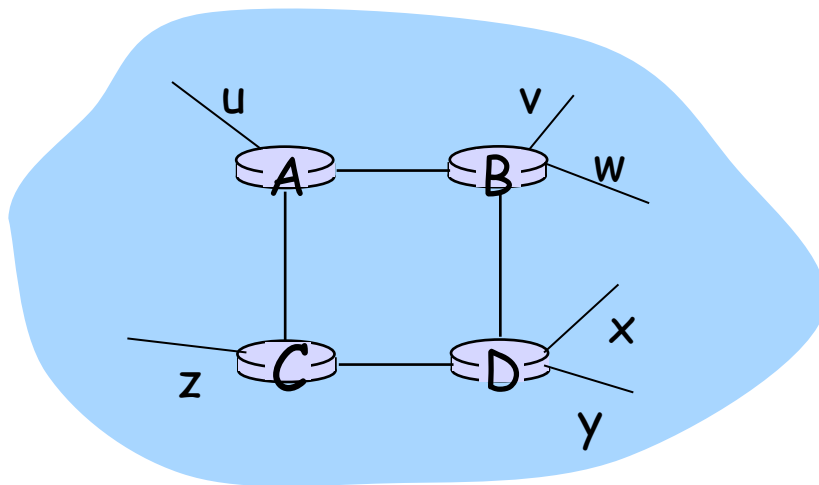
## 4.7 Broadcast and multicast routing

# 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)

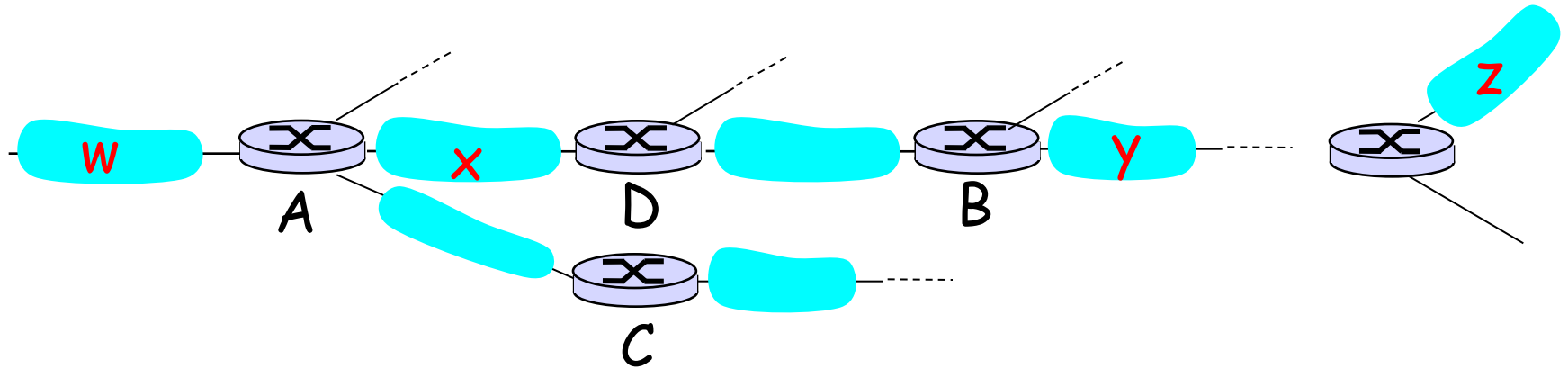


from router A to destination **subnets**:

<u>subnet</u>	<u>hops</u>
u	1
v	2
w	2
x	3
y	3
z	2



# RIP: Example



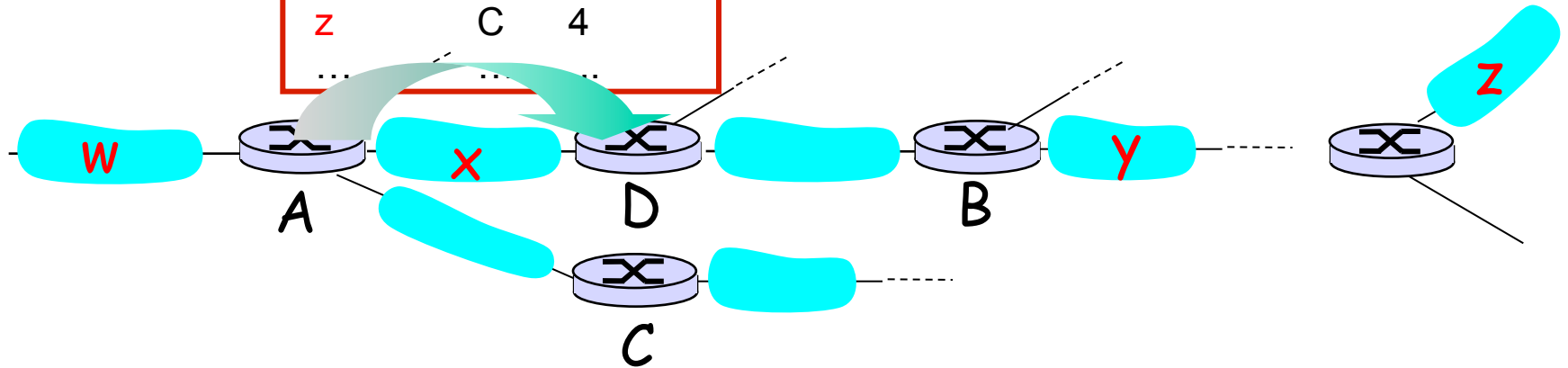
routing table in router D

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
Z	<del>B</del> → A	<del>7</del> → 5
X	--	1
....	....	....

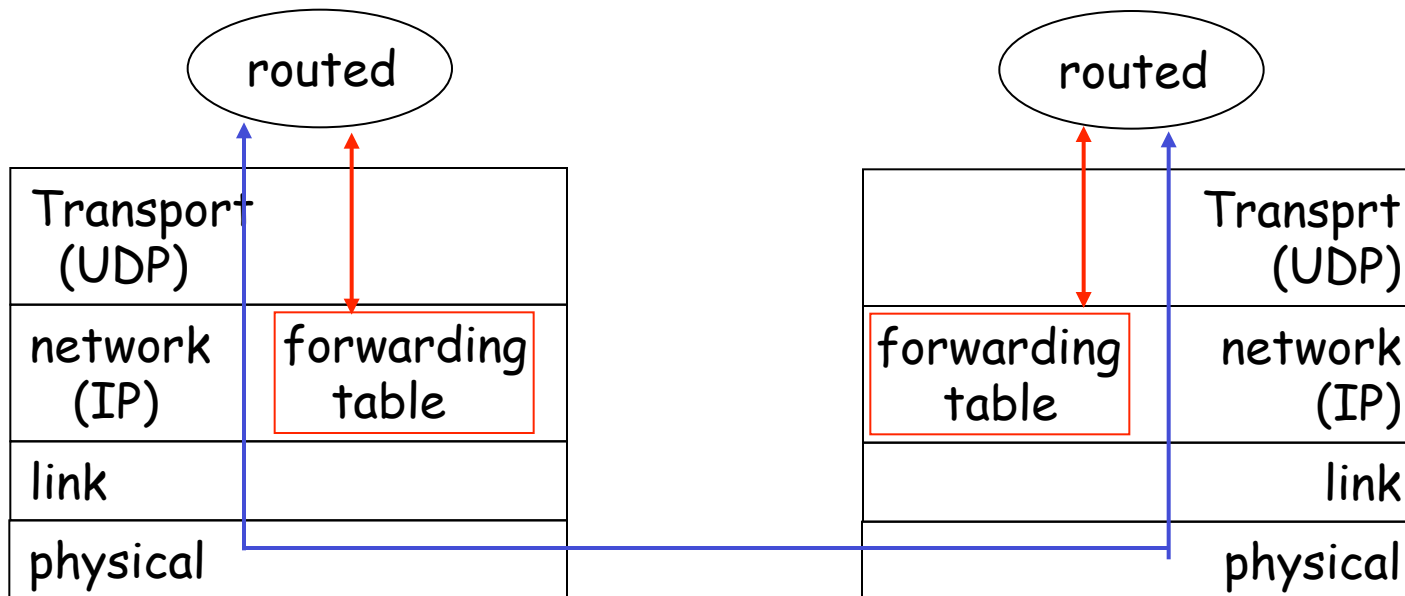
# RIP: Link Failure and Recovery

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



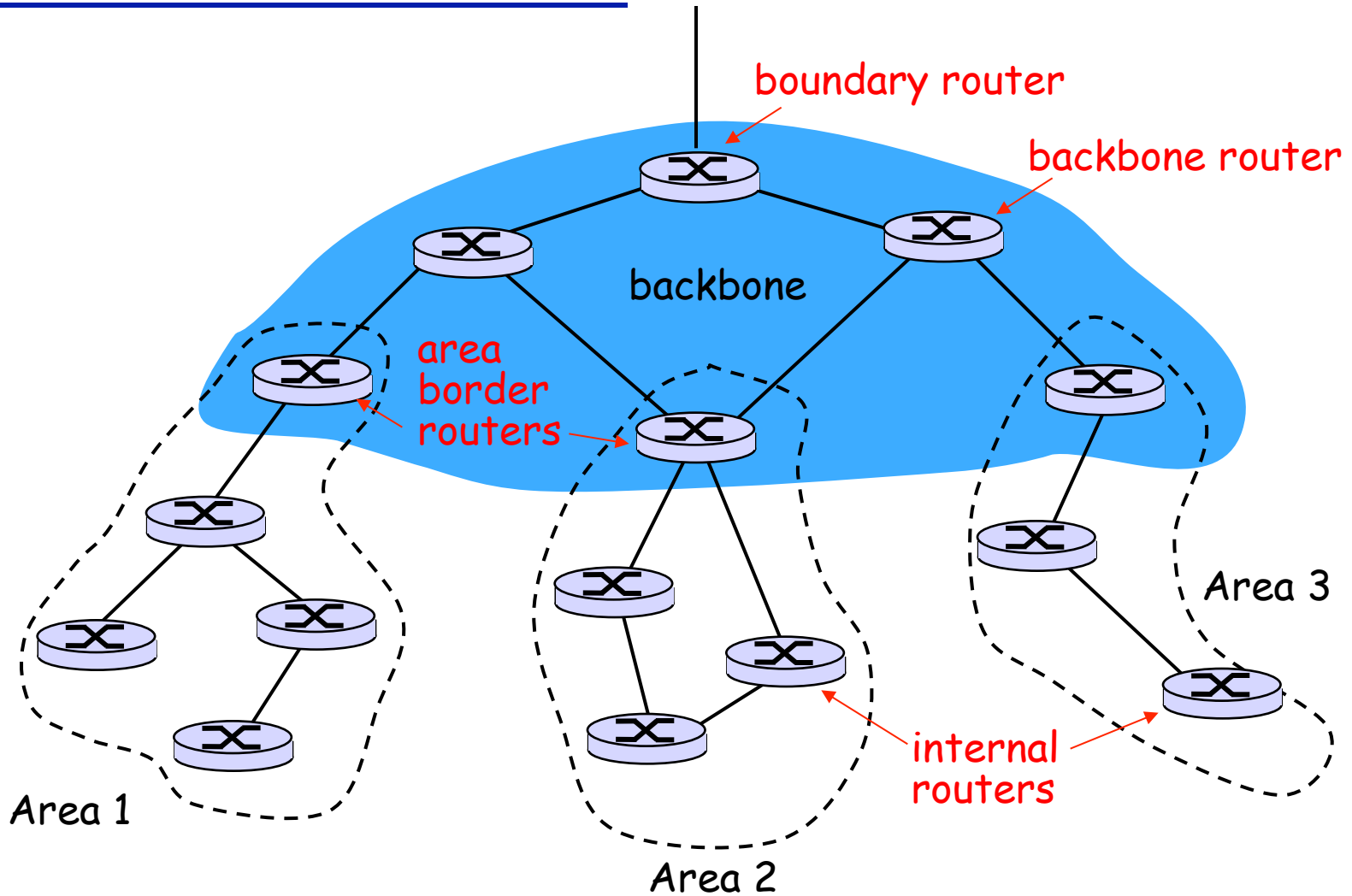
# OSPF (Open Shortest Path First)

- ❖ “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 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



# 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.