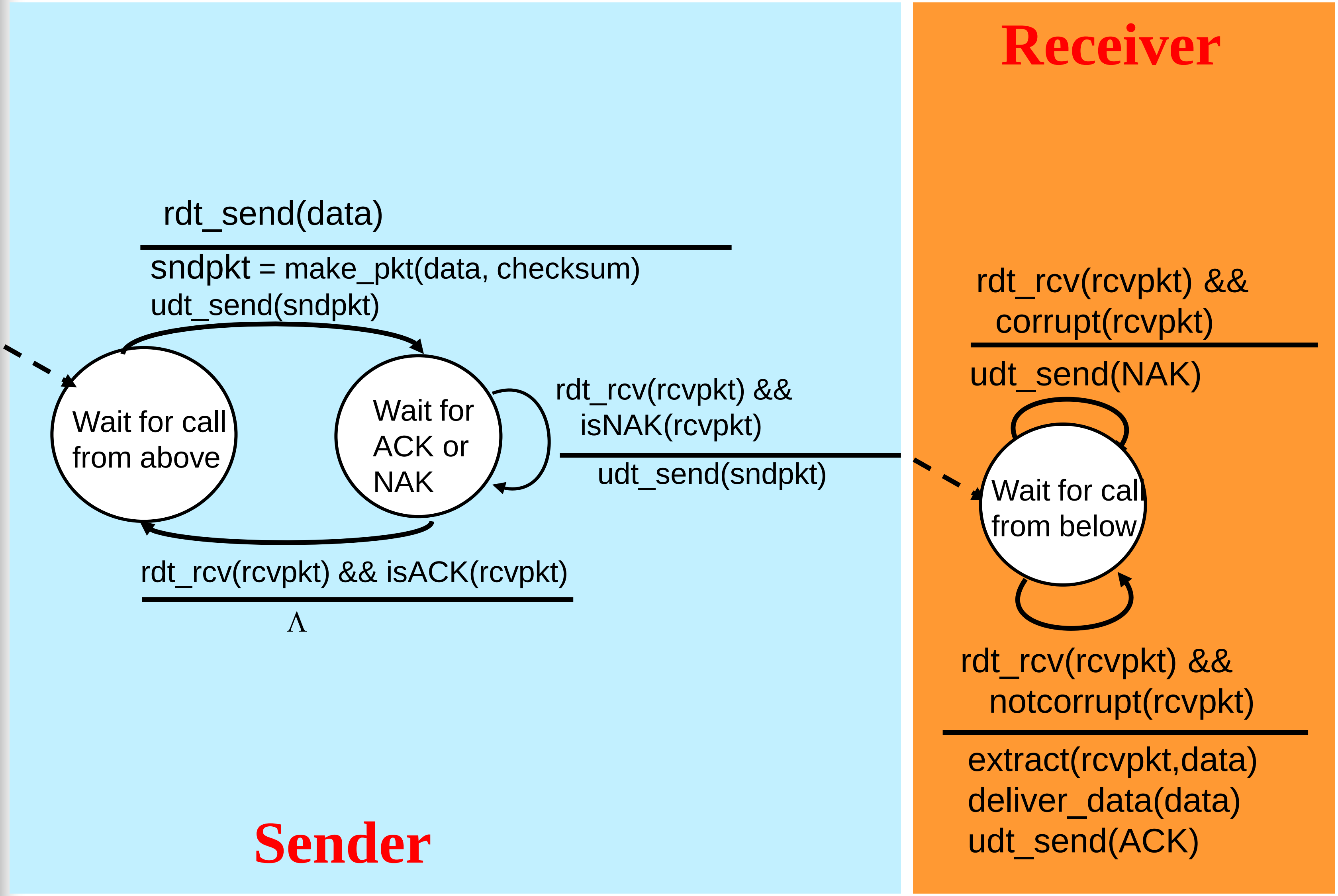
Protocoles et Interconnexions

Course Overview and Introduction

Dario Vieira
Department of Computer Science
EFREI

Rdt (Reliable Data Transfer) 2.0: FSM specification

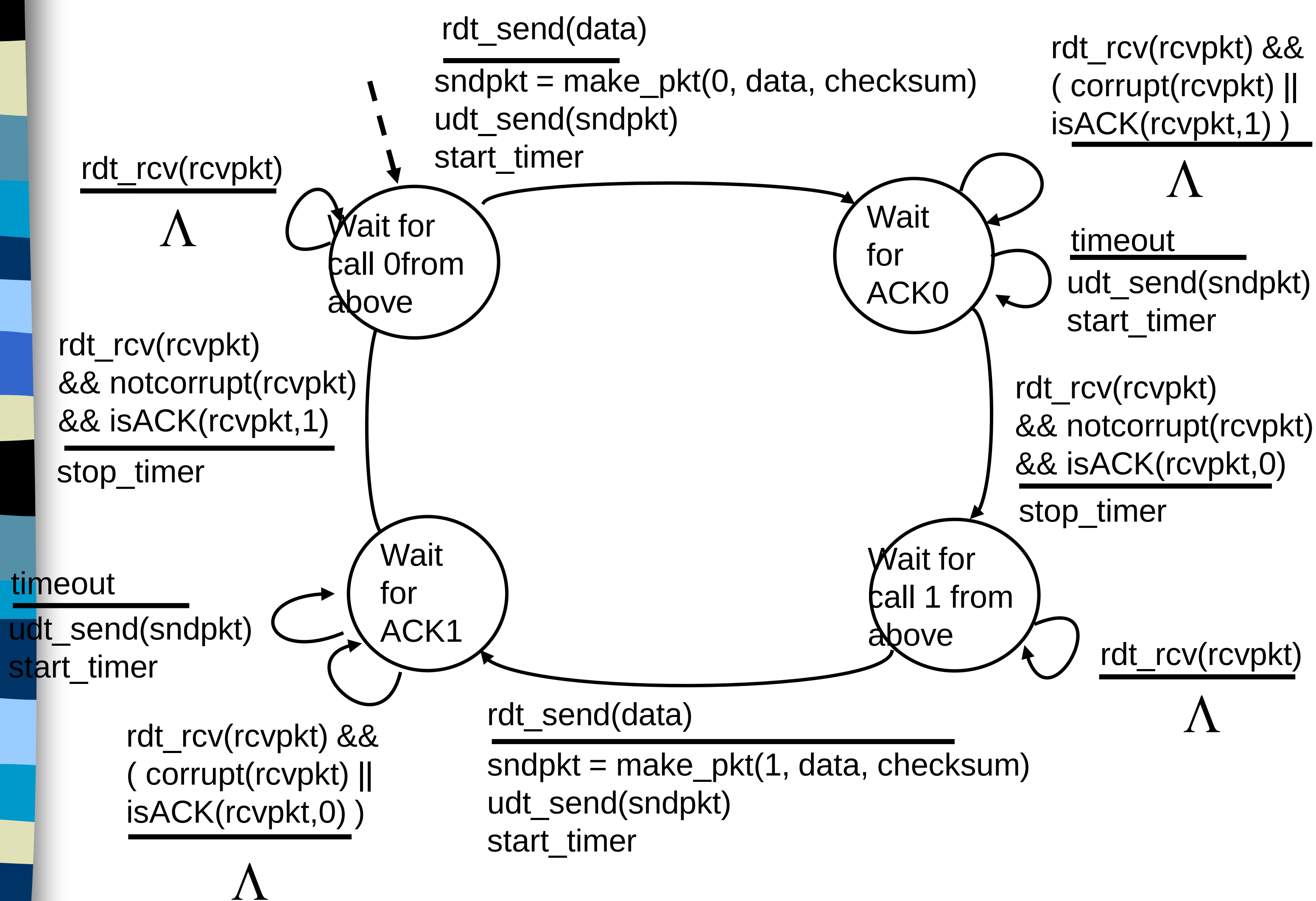


rdt (Reliable Data Transfer) 3.0: channels with errors *and* loss

New assumption:
underlying channel can also lose packets (data or ACKs)
- checksum, seq. #, ACKs, retransmissions will be of help, but not enough

Approach: sender waits "reasonable" amount of time for ACK
- retransmits if no ACK received in this time
- if pkt (or ACK) just delayed (not lost):
  - retransmission will be duplicate, but use of seq. #'s already handles this
  - receiver must specify seq # of pkt being ACKed
- requires countdown timer

rdt3.0 sender



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

Autonomous Systems (ASes)

A collection of physical networks glued together using IP, that have a unified administrative routing policy.

- **Campus networks**
- **Corporate networks**
- **ISP Internal networks**
- **…**

AS Numbers (ASNs)

ASNs are 16 bit values
64512 through 65535 are "private"

Currently over 11,000 in use

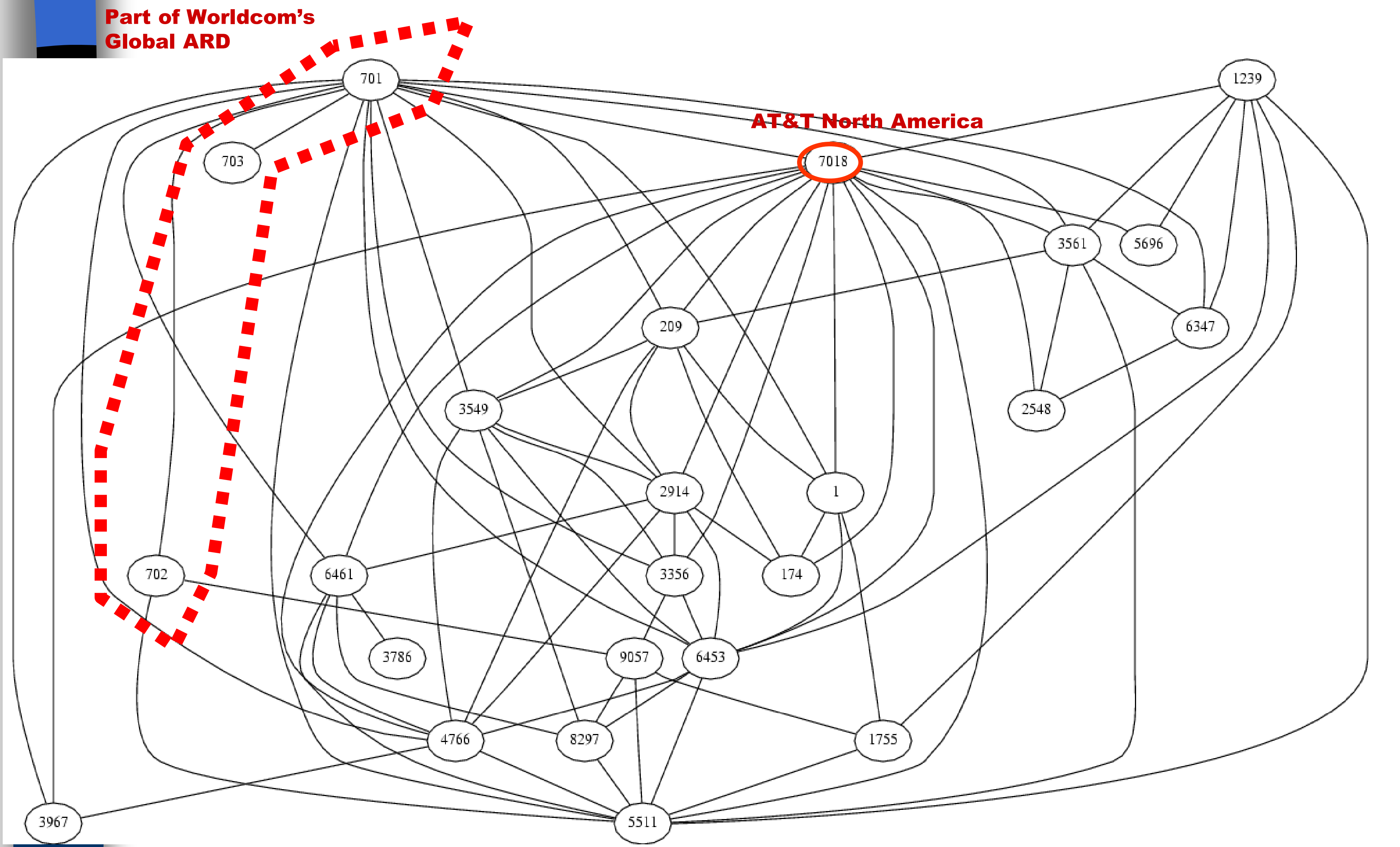
- **Genuity (f.k.a. BBN): 1**
- **MIT: 3**
- **Harvard: 11**
- **UC San Diego: 7377**
- **AT&T: 7018, 6341, 5074, …**
- **UUNET: 701, 702, 284, 12199, …**
- **Sprint: 1239, 1240, 6211, 6242, …**
- **…**

ASNs represent units of routing policy

Thanks to T. Griffin

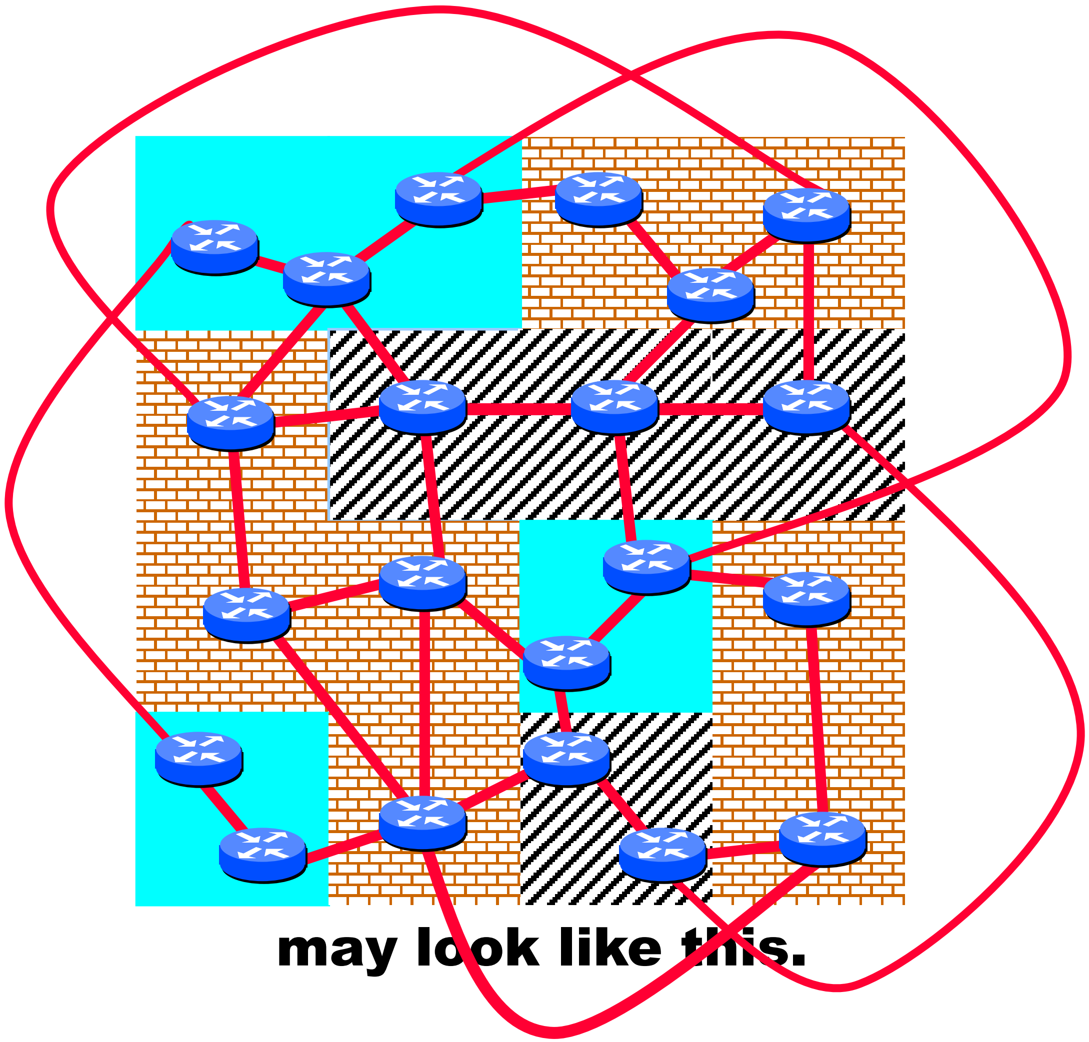
Thanks to T. Griffin

AS Graphs Can Be Fun



The subgraph showing all ASes that have more than 100 neighbors in full graph of 11,158 nodes. July 6, 2001. Point of view: AT&T route-server

AS Graph != Internet Topology

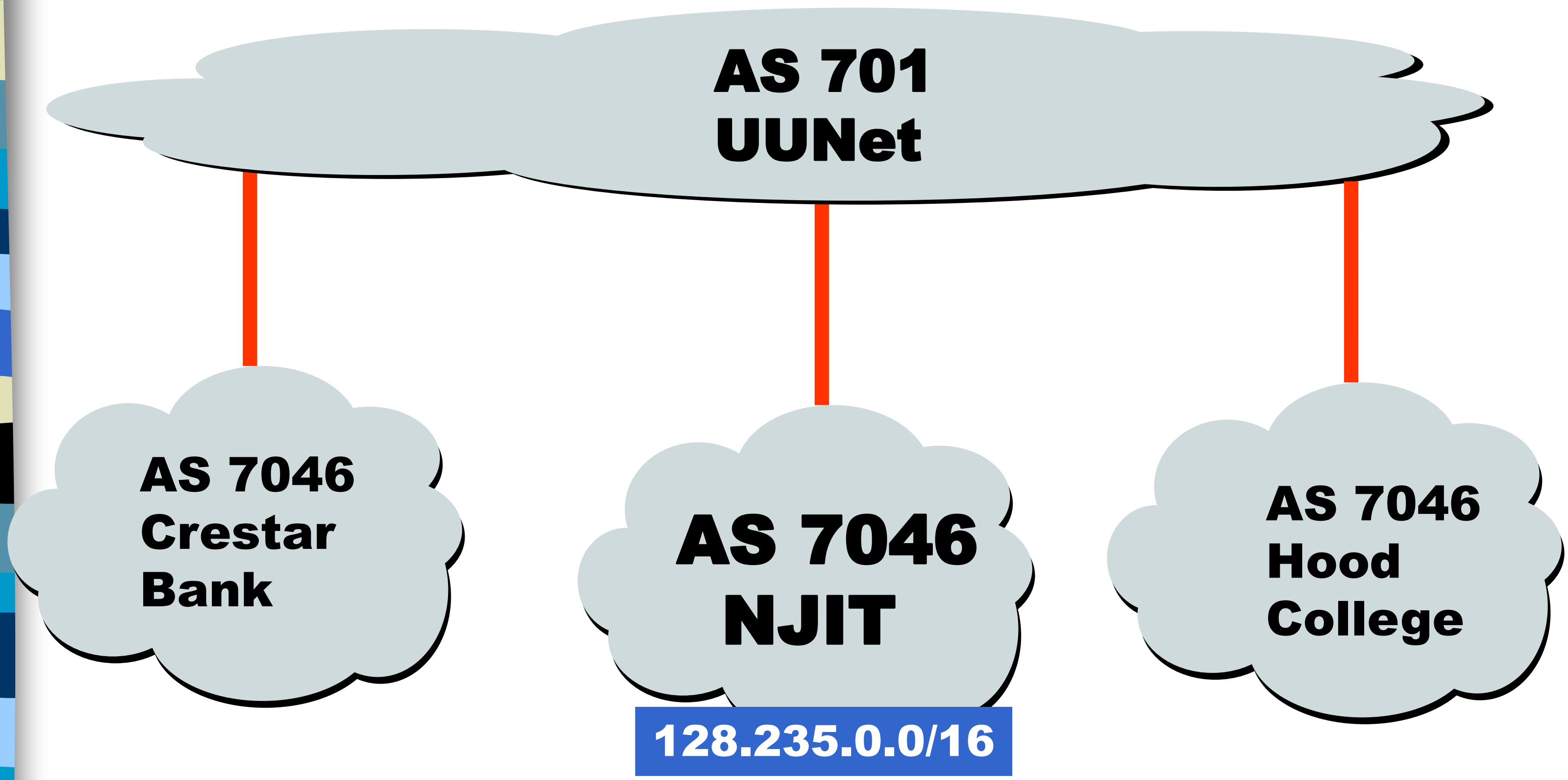


may look like this.

Reality may be closer to this…

Thanks to T. Griffin

ASNs Can Be “Shared” (RFC 2270)



ASN 7046 is assigned to UUNet. It is used by Customers single homed to UUNet, but needing BGP for some reason (load balancing, etc..) [RFC 2270]

Thanks to T. Griffin

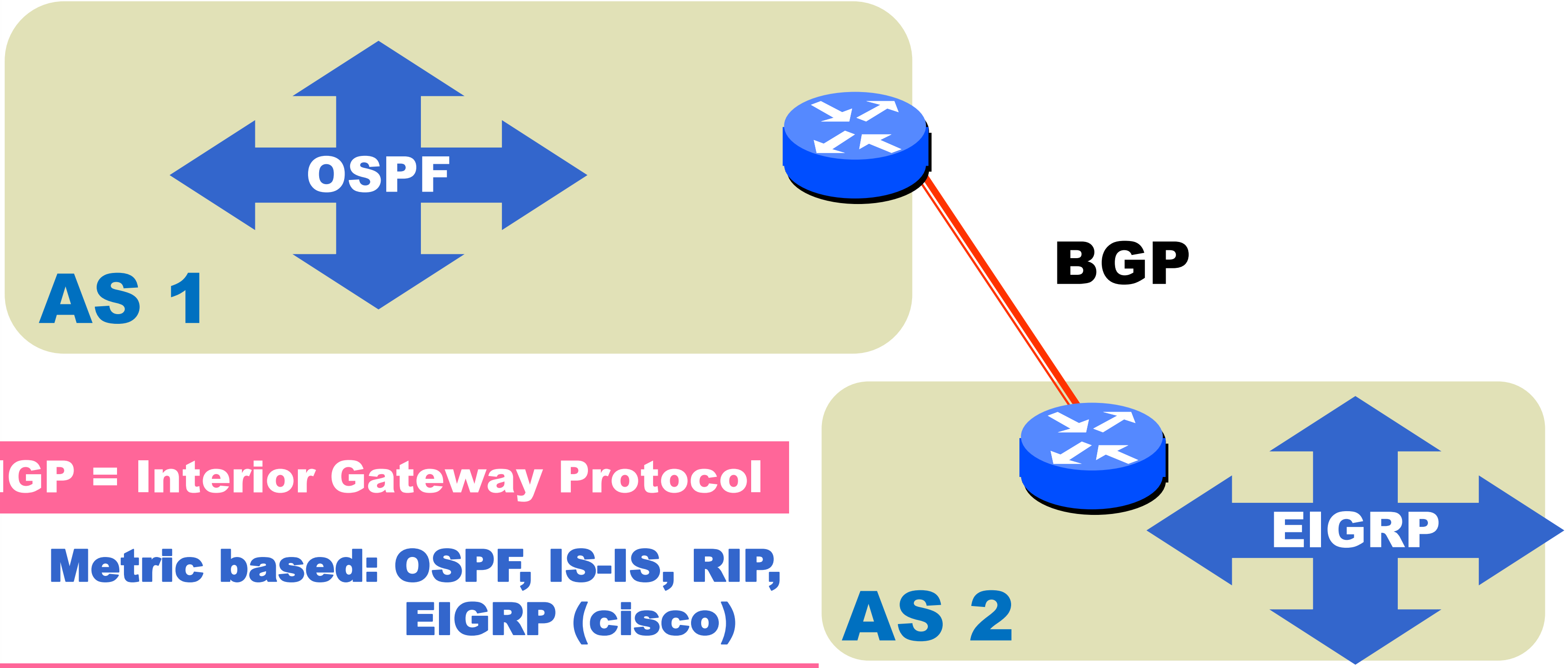
Hierarchical Routing

- Aggregate routers into regions, "Autonomous Systems" (AS)
- Routers in same AS run same routing protocol
  - Intra-AS routing protocol
    - IGP: Interior Gateway Protocol
    - e.g. , RIP
  - Routers in different AS can run **different** intra-AS routing protocol

Gateway router

- At "edge" of its own AS
- Direct link to router in another AS
- All ASes run the same Inter-AS routing protocol
  - EGP: Exterior Gateway Protocol
  - e.g., Border Gateway Protocol (BGP)

Architecture of Dynamic Routing



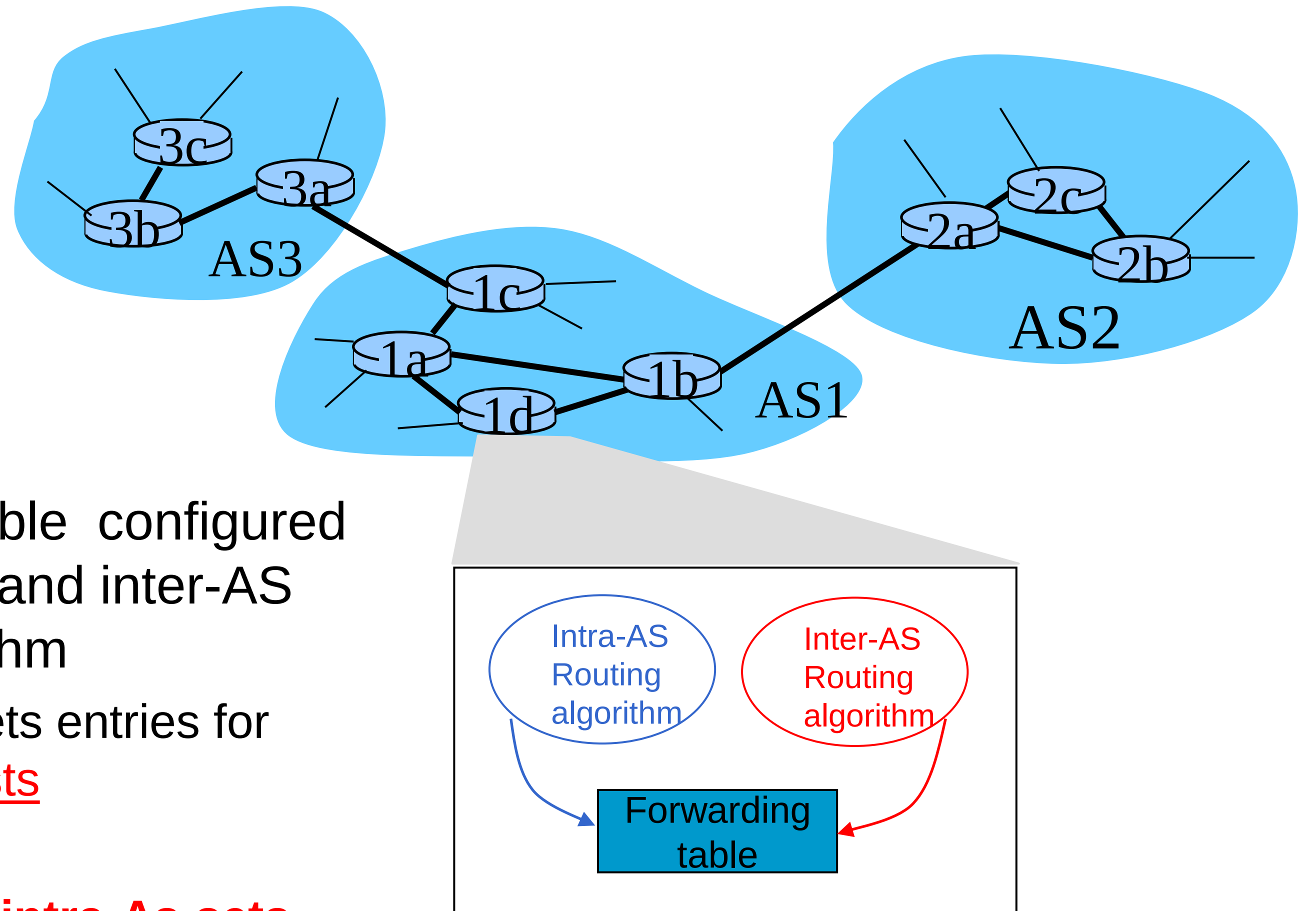
IGP = Interior Gateway Protocol

Metric based: OSPF, IS-IS, RIP, EIGRP (cisco)

EGP = Exterior Gateway Protocol

Policy based: BGP

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

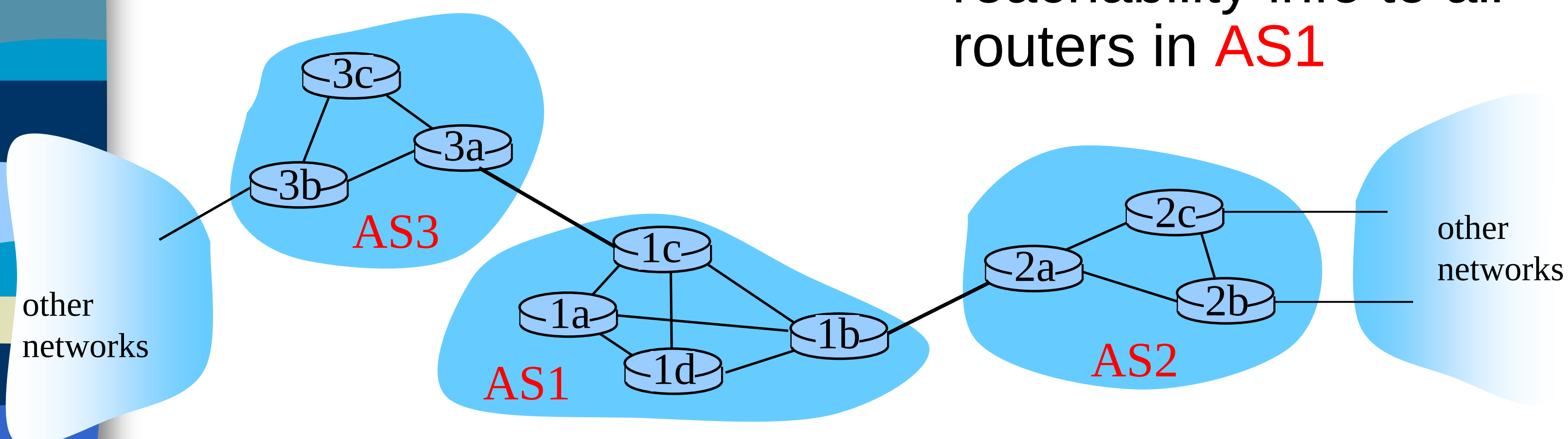
Inter-AS tasks

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

Job of inter-AS routing!

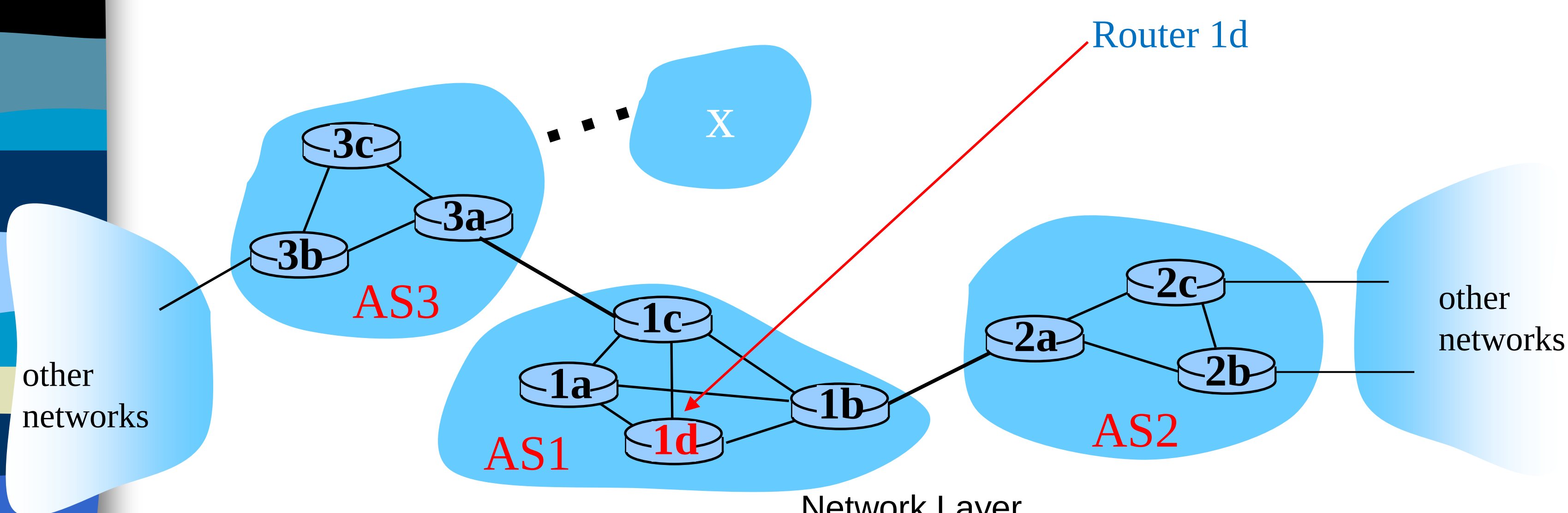
AS1 must

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



Example: Setting forwarding table in router 1d

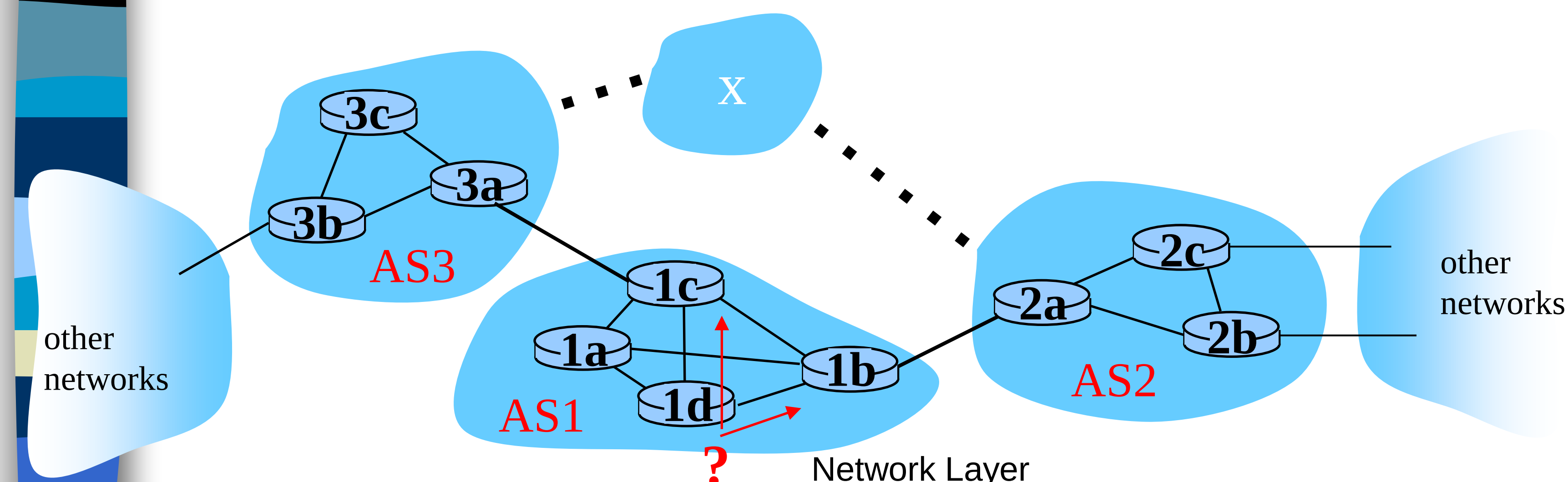
- Suppose AS1 learns (via inter-AS protocol) that subnet *x* reachable via AS3 (gateway 1c) but not via AS2.
  - inter-AS protocol propagates reachability info to all internal routers
- Router 1d determines from intra-AS routing info that its interface *I* is on the least cost path to 1c.
  - installs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



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

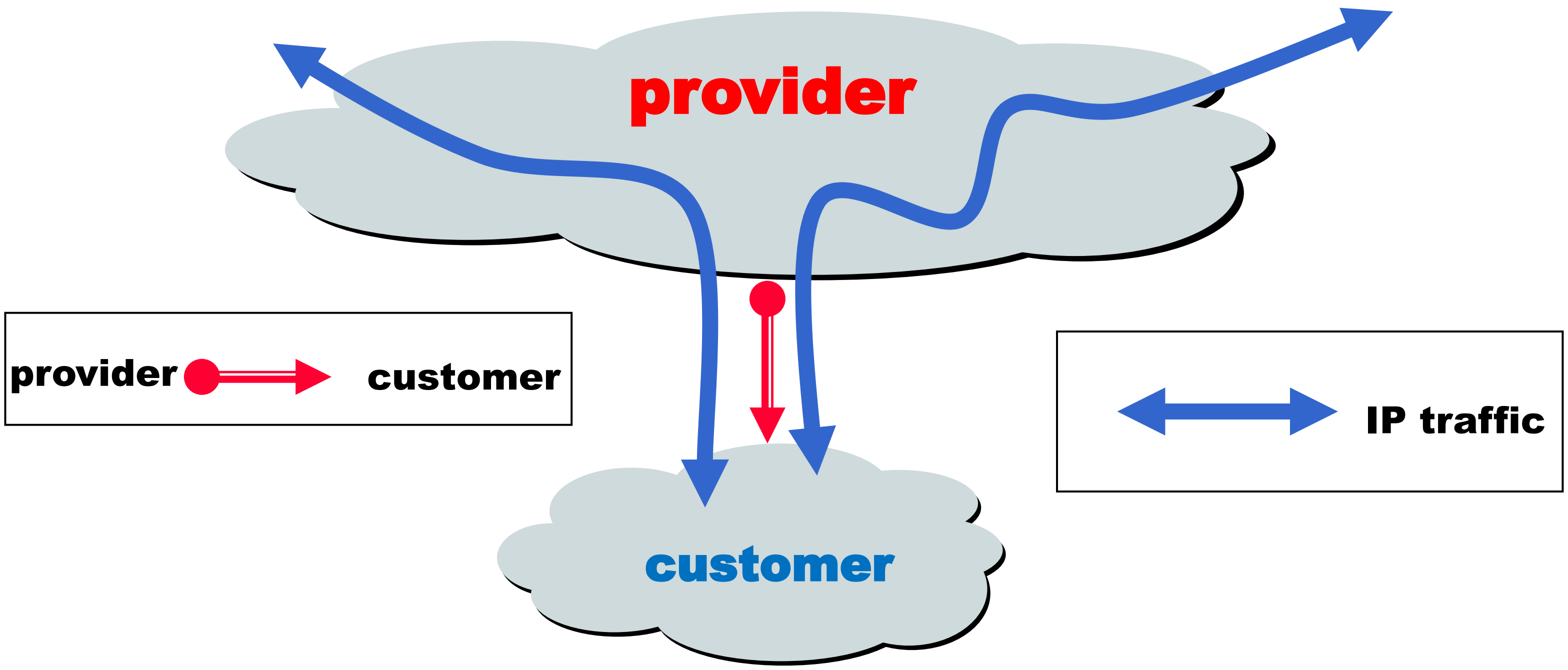
Intra-AS Routing

- 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)

Relationships Between Networks

Thanks to T. Griffin

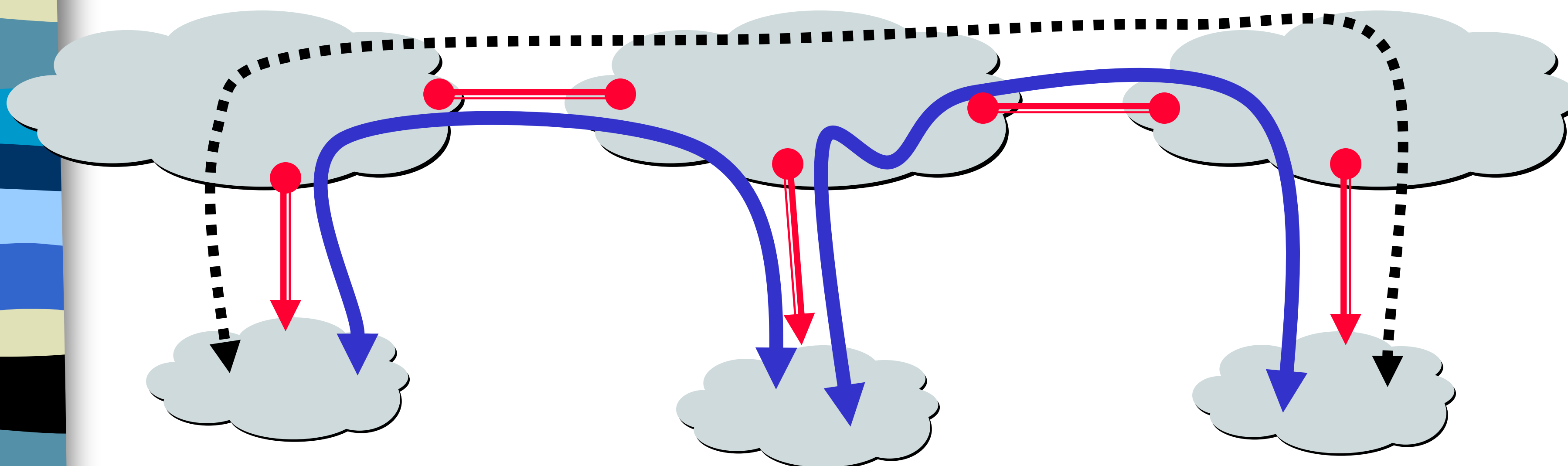
Customers and Providers

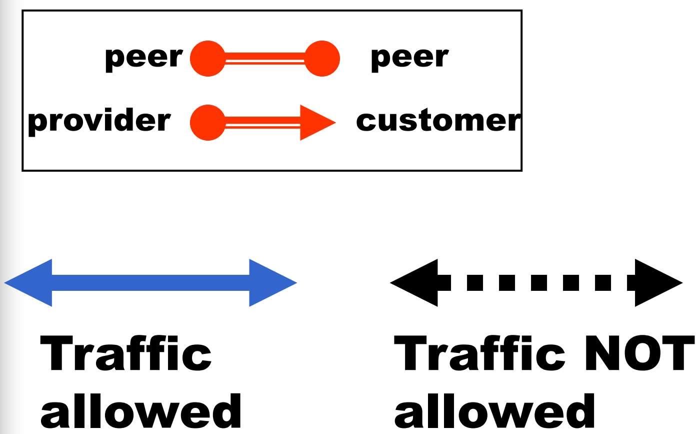


Customer pays provider for access to the Internet

Thanks to T. Griffin

The "Peering" Relationship





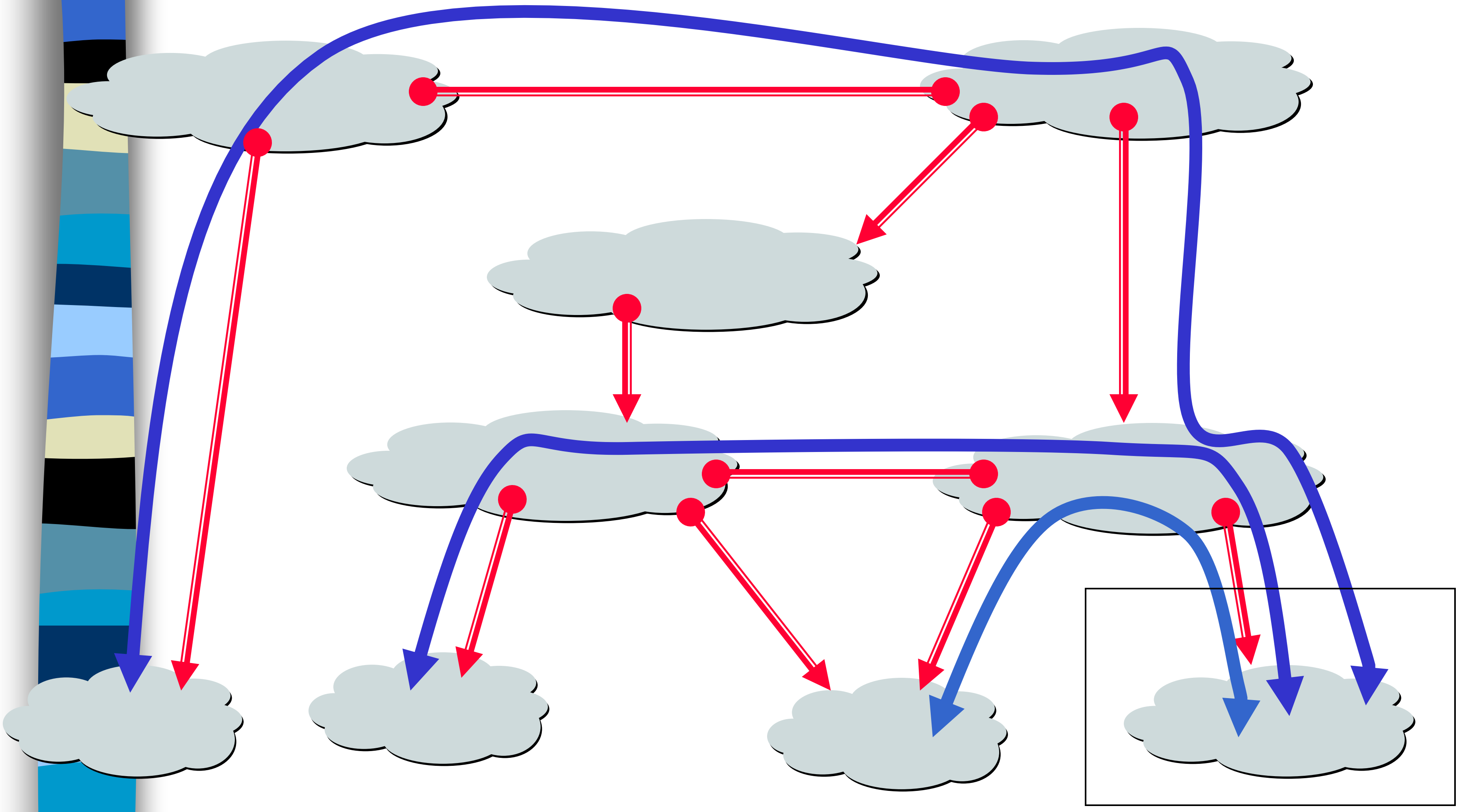
Peers provide transit between their respective customers

Peers do not provide transit between peers

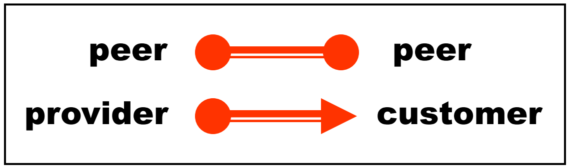
Peers (often) do not exchange $$$

Thanks T. Griffin

Peering Provides Shortcuts



Peering also allows connectivity between the customers of “Tier 1” providers.



Peering Wars

Peer

- Reduces upstream transit costs
- Can increase end-to-end performance
- May be the only way to connect your customers to some part of the Internet (“Tier 1”)

Don’t Peer

- You would rather have customers
- Peers are usually your competition
- Peering relationships may require periodic renegotiation

Peering struggles are by far the most contentious issues in the ISP world!

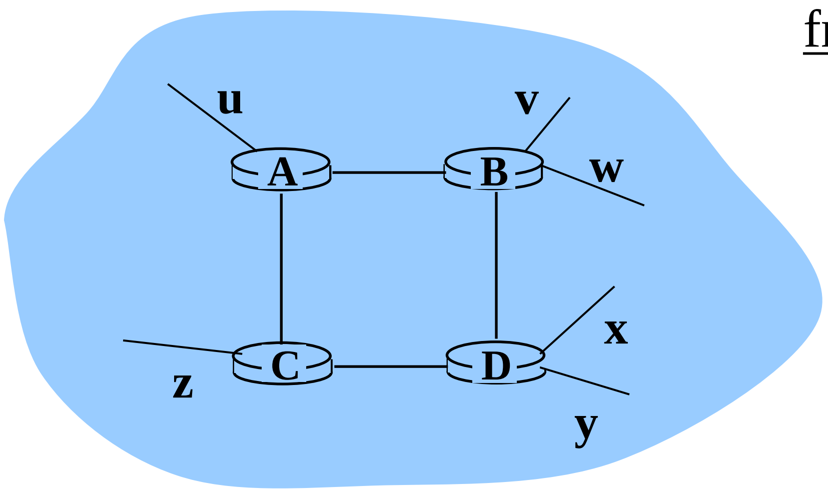
Peering agreements are often confidential.

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
- 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

RIP (Routing Information Protocol)

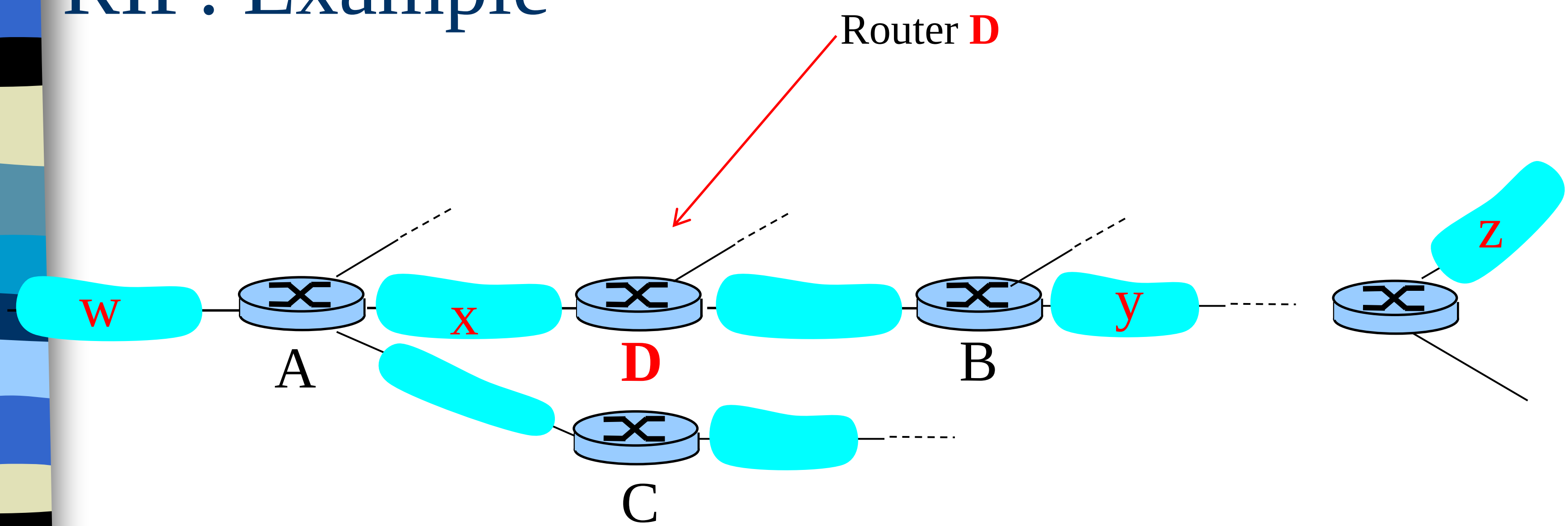
- Included in BSD-UNIX distribution in 1982
- Distance vector algorithm
  - distance metric: # hops (**max = 15 hops**), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing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



from router A to destination subnets

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

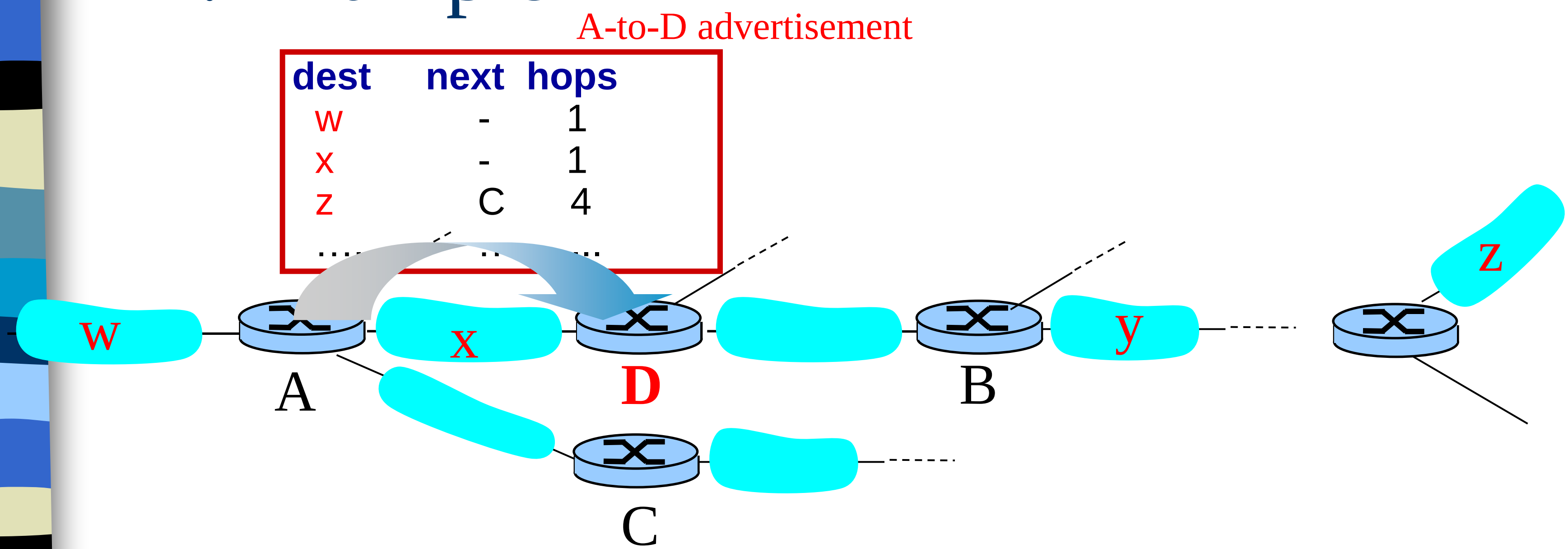
RIP: Example



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



Routing table in router **D**

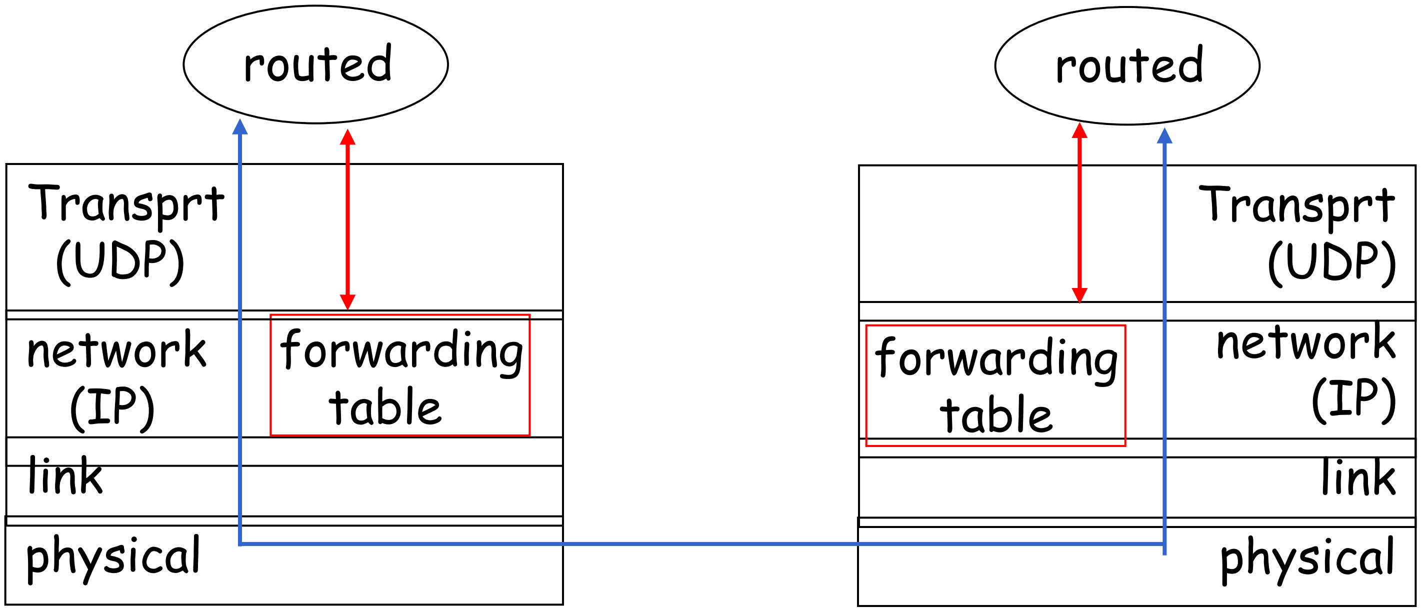
destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
....	....	....

RIP: Link Failure 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)

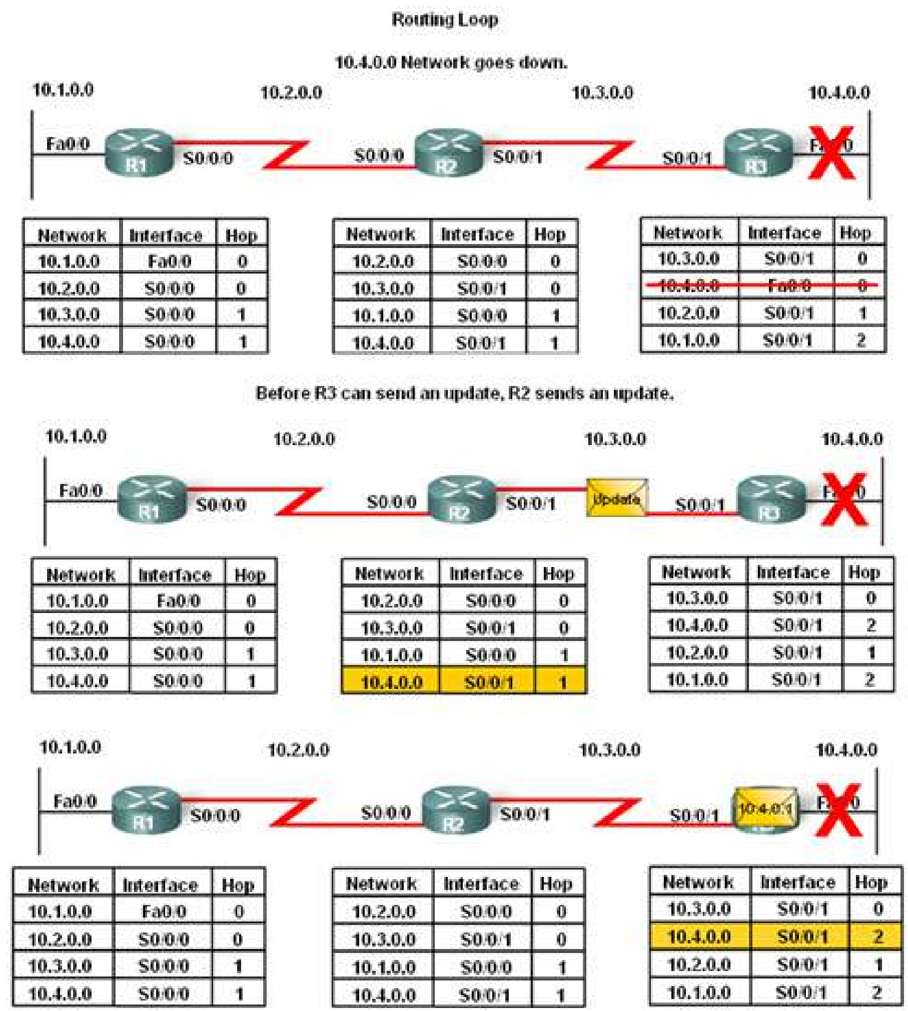
RIP Table Processing

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



Routing Loops

- **Routing loops** are
  - A condition in which a packet is continuously transmitted within a series of routers
  - without ever reaching its destination.

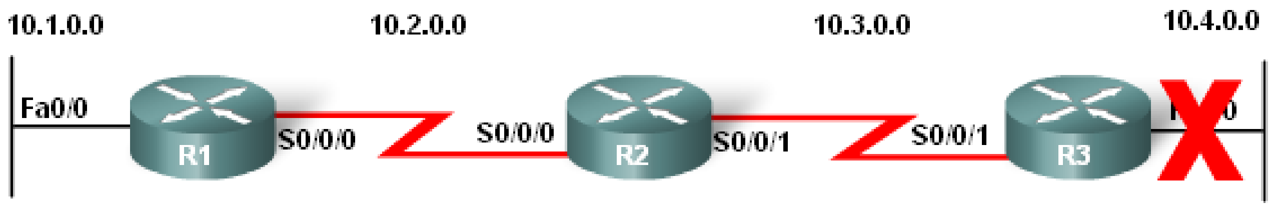


Routing Loops

- **Count to Infinity**
  - This is a routing loop whereby packets bounce infinitely around a network

Count to Infinity

Each round of updates continues to increase hop count.



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	24

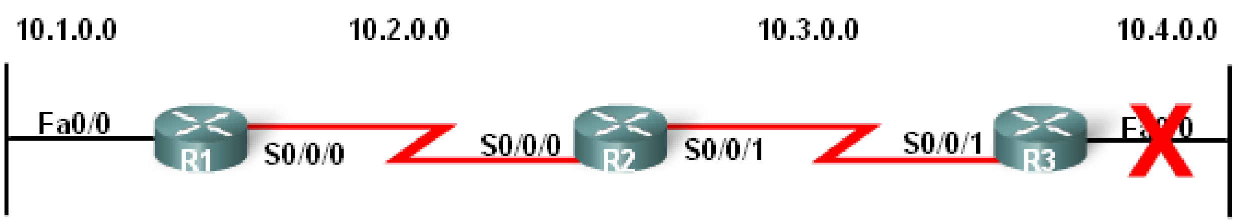
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	23

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	22
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Routing Loops

- Setting a maximum
  - **Distance Vector routing protocols** set a specified metric value to indicate infinity
    - Once a router “counts to infinity” it marks the route as unreachable

10.4.0.0 is unreachable. Hop count is 16.



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	16

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	16

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	16
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Routing Loops

- The **Split Horizon Rule** is used to prevent routing loops
- **Split Horizon rule**:
  - A router should not advertise a network through the interface from which the update came

Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	1

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

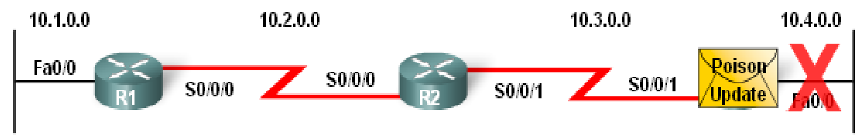
Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Routing Loops

- **Split horizon with poison reverse**
  – A router learns of an unreachable route through an interface
  – Advertise it as unreachable back through the same interface

Poison Reverse

Network 10.4.0.0 goes down.
R3 “poisons” route with an “infinite” metric.
R3 sends triggered Poison Update to R2.



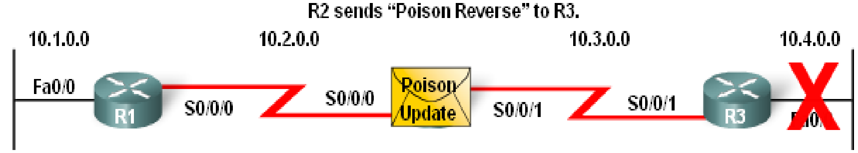
Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	16
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Poison Reverse

R2 “poisons” route with an “infinite” metric.
R2 sends “Poison Reverse” to R3.



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

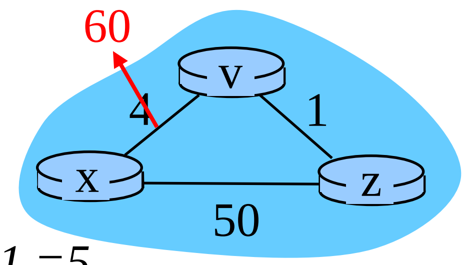
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	16

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	16
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

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



  - Bef: $D_v(x) = 4$, $D_v(z) = 1$, $D_z(v) = 1$, $D_z(x) = 4+1 = 5$
  - Aft: $D_v(x) = min\{60, c(v,z) + D_z(x)\} = 6$
  - Next: $D_z(x) = \ldots = c(z,v) + D_v(x) = 1+6 = 7$
  - Then: $D_v(x) = c(v,z) + D_z(x) = 1+7 = 8$

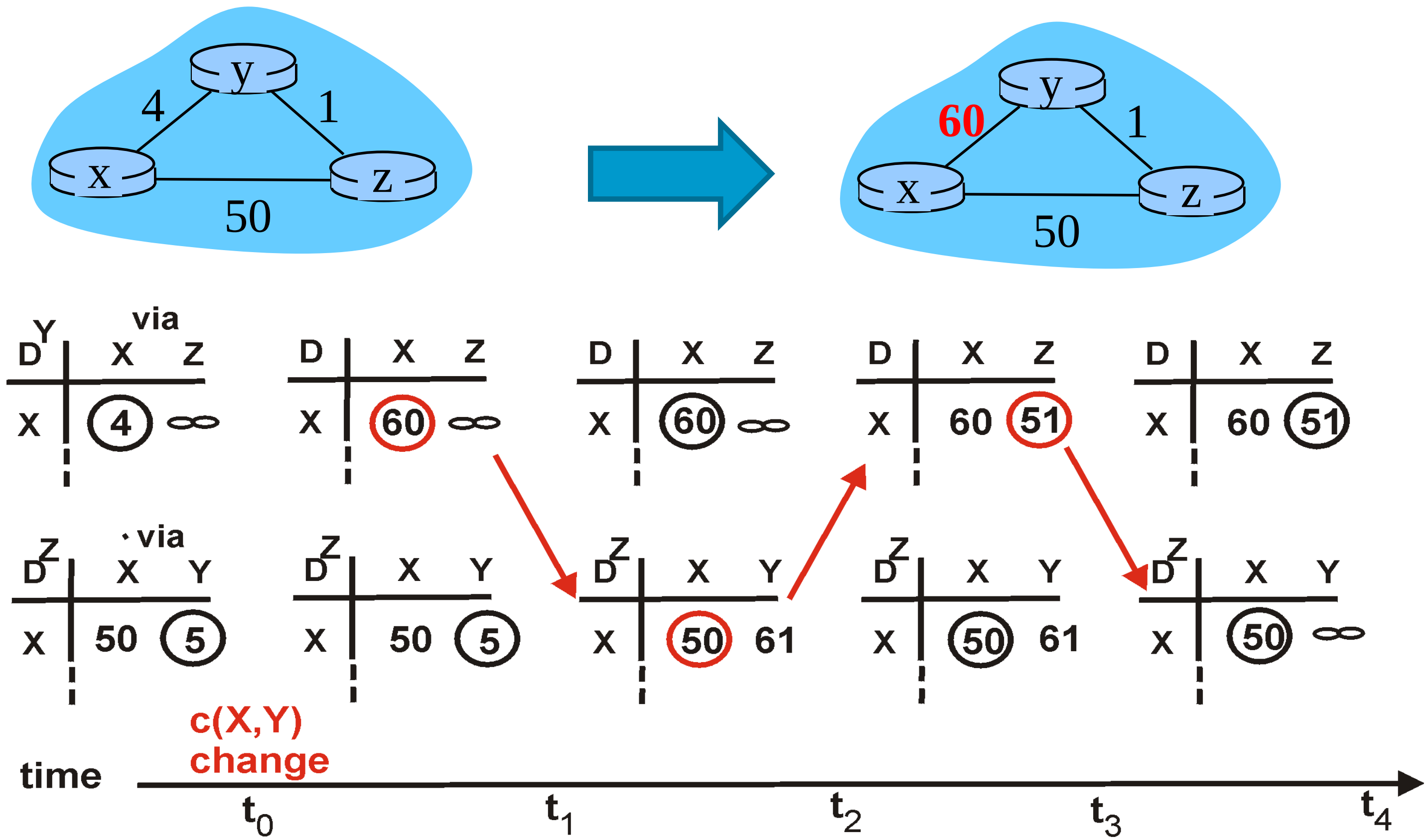
Poisoned reverse:

- If z routes through v to get to x :
  - z tells v: $D_z(x) = \infty$ *(so v won't route to x via z)*
    - $\longrightarrow D_v(x) = 60 \longrightarrow D_z(x) = 50 \longrightarrow D_v(x) = 51$
- Will this completely solve count to infinity problem?

Distance Vector: link cost changes

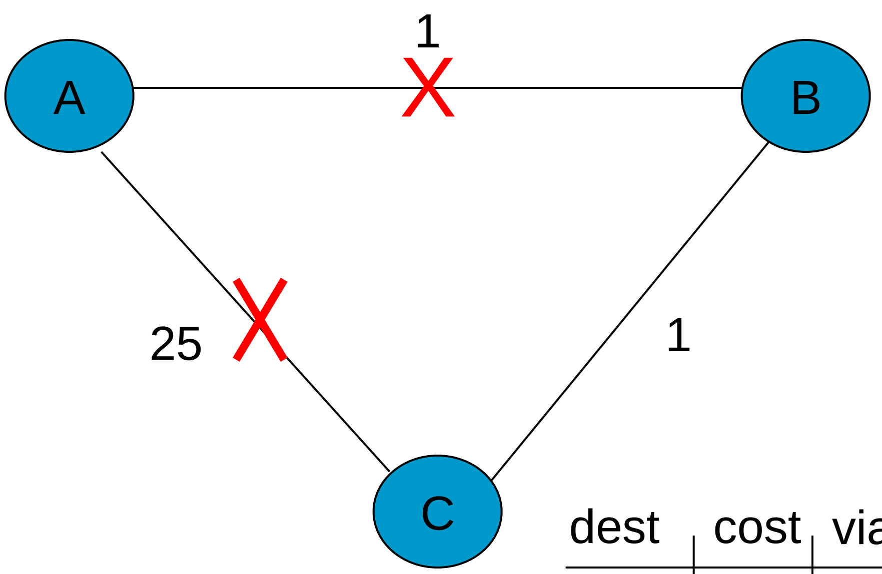
Poisoned reverse:

- If z routes through y to get to x :
  - z tells y: $D_z(x) = \infty$ *(so y won't route to x via z)*



Link Failure Causes "Counting to Infinity" Effect

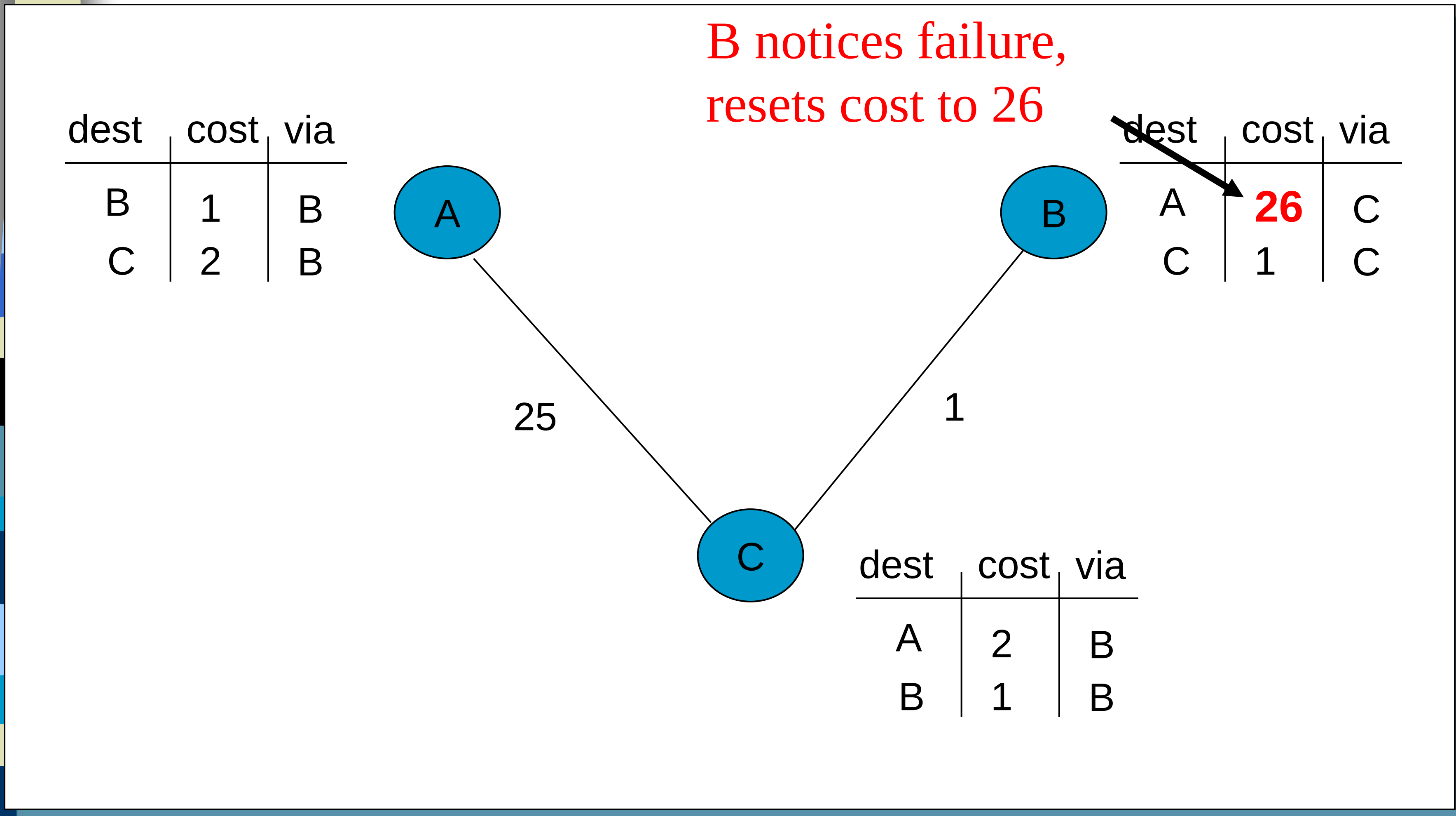
dest	cost	via
B	1	B
C	2	B



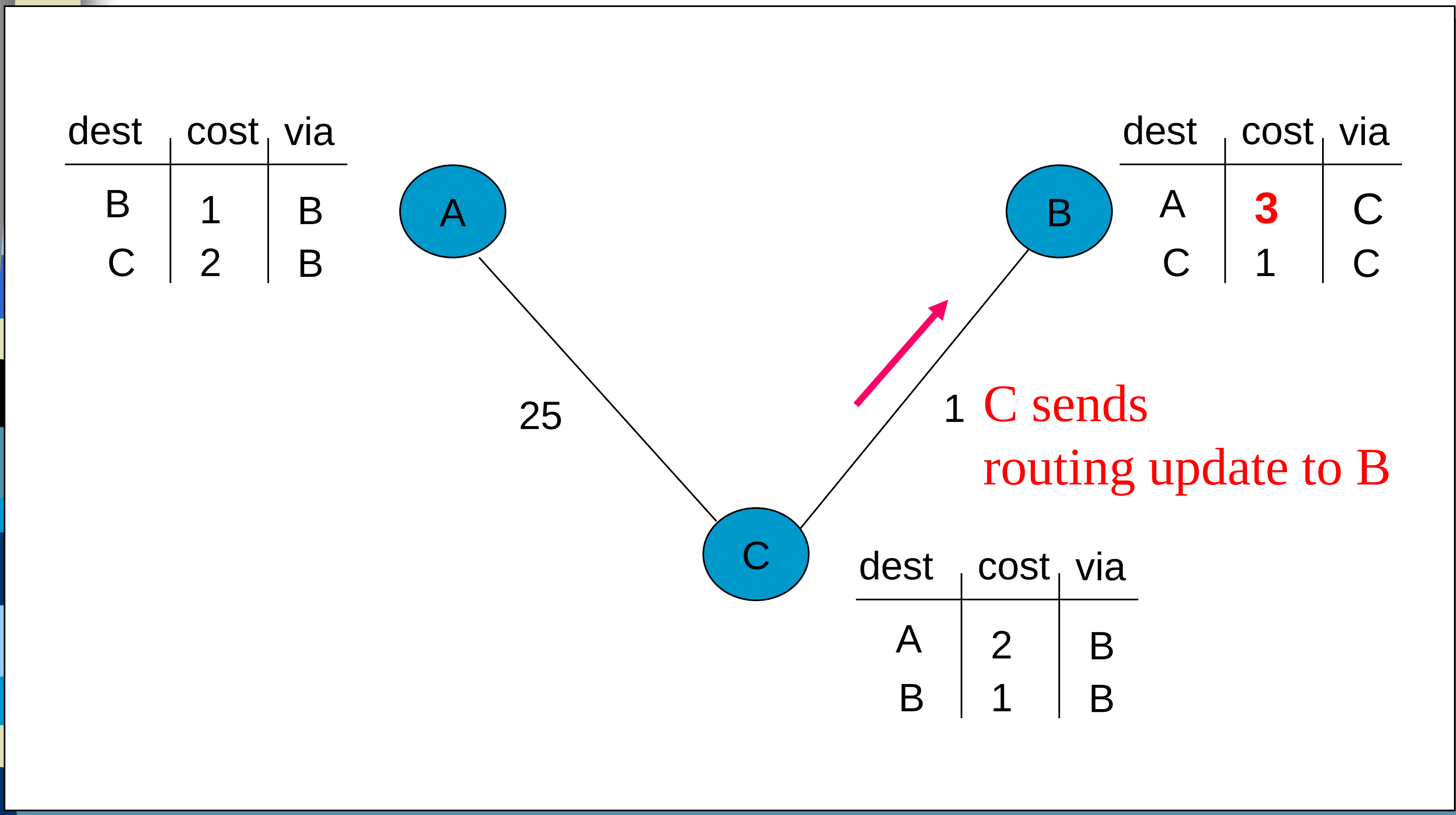
dest	cost	via
A	1	A
C	1	C

dest	cost	via
A	2	B
B	1	B

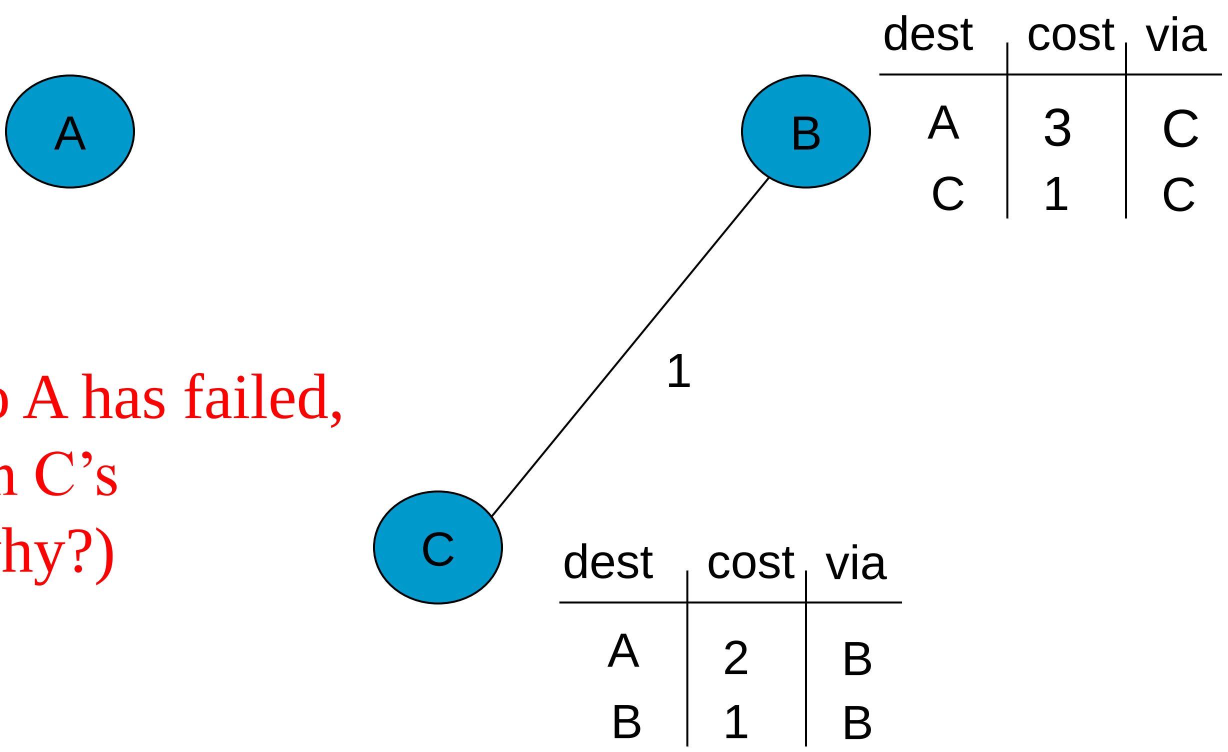
B Notices A-B Link Failure



C Sends Dist. Vector to B

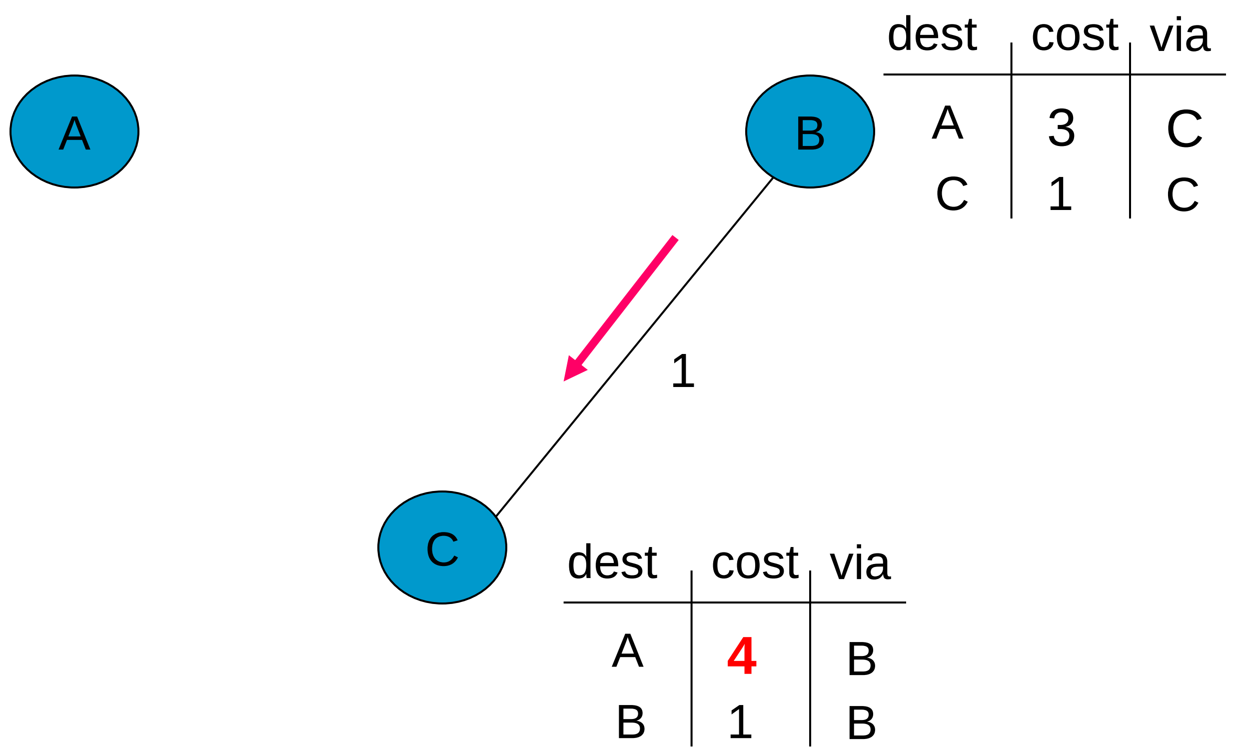


A-C Link Fails

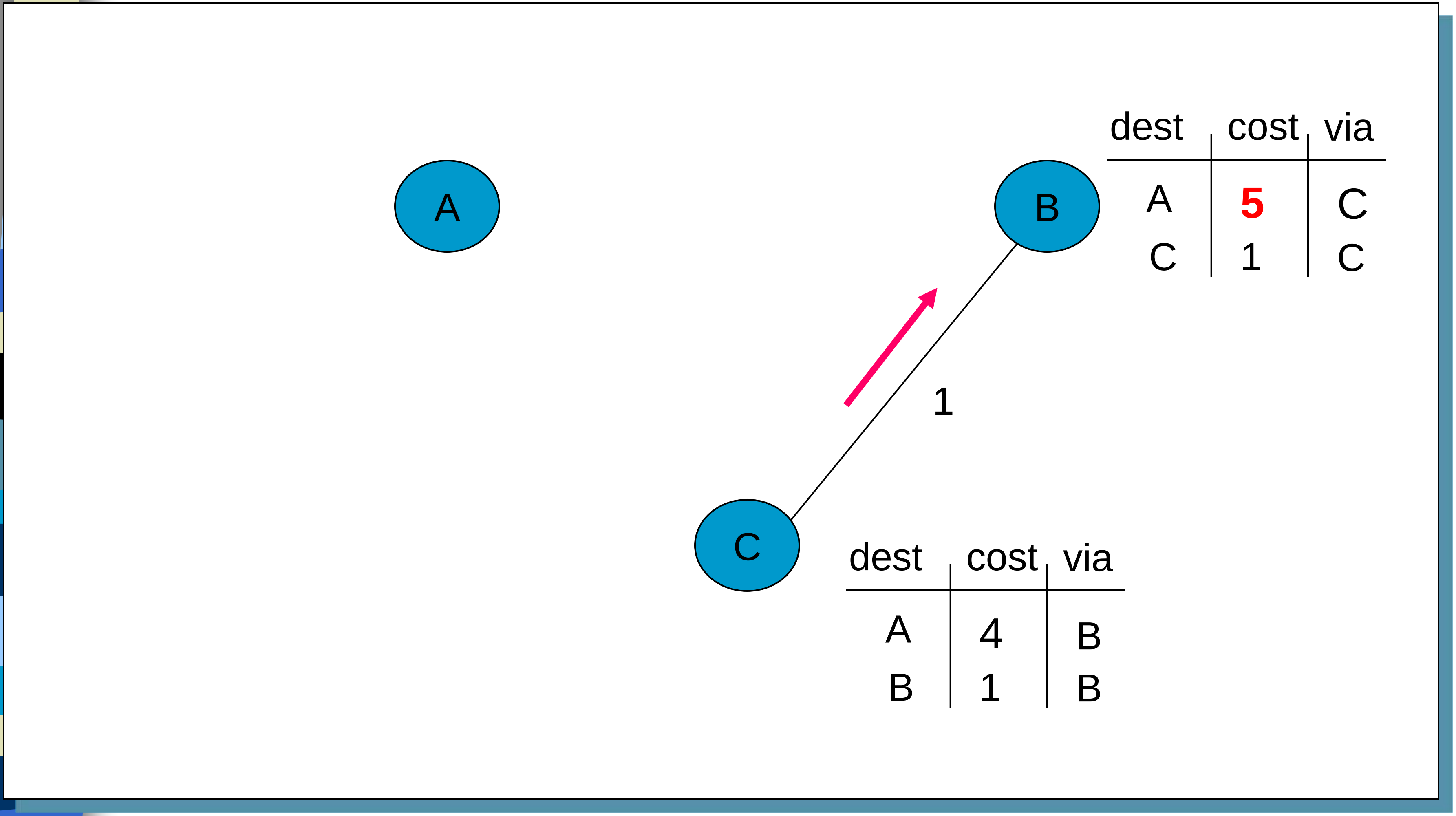


C detects link to A has failed, but no change in C's routing table (why?)

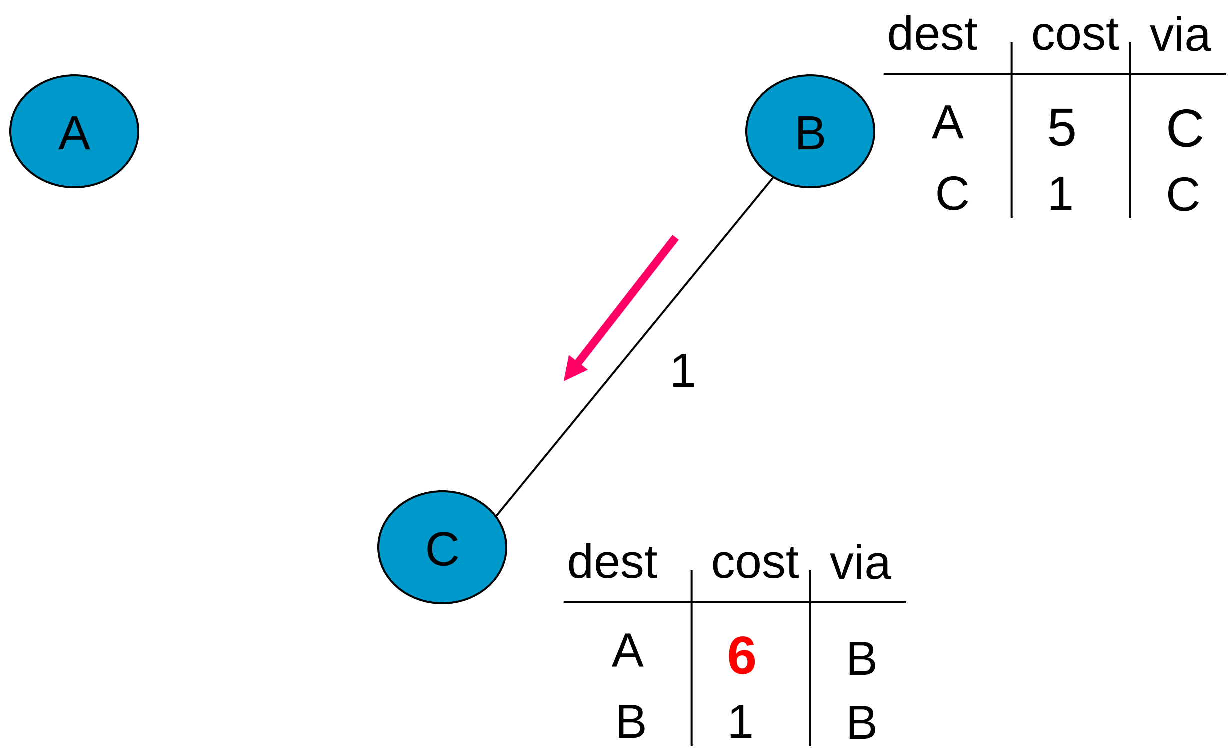
Now, B and C Count to Infinity



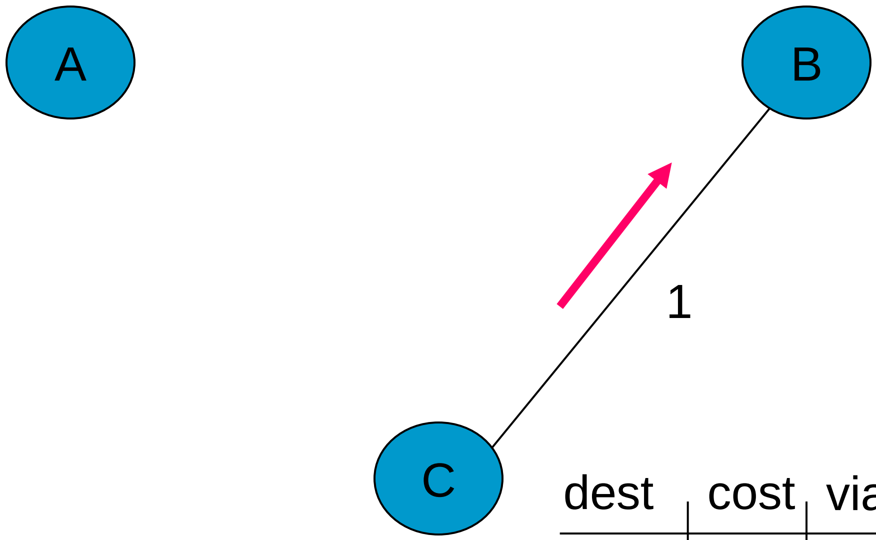
B and C Count to Infinity (2)



B and C Count to Infinity (3)



B and C Count to Infinity (4)



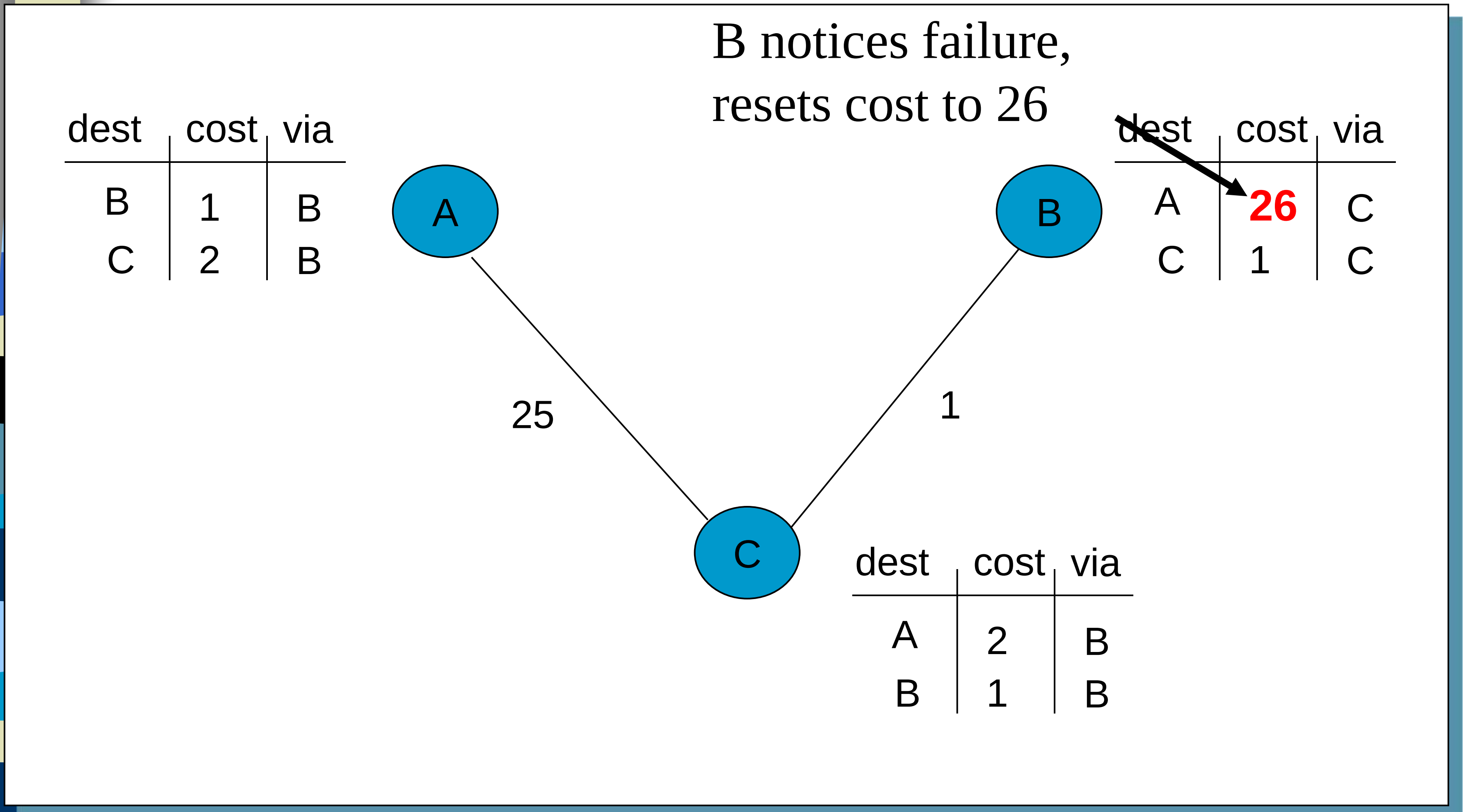
dest	cost	via
A	7	C
C	1	C

dest	cost	via
A	6	B
B	1	B

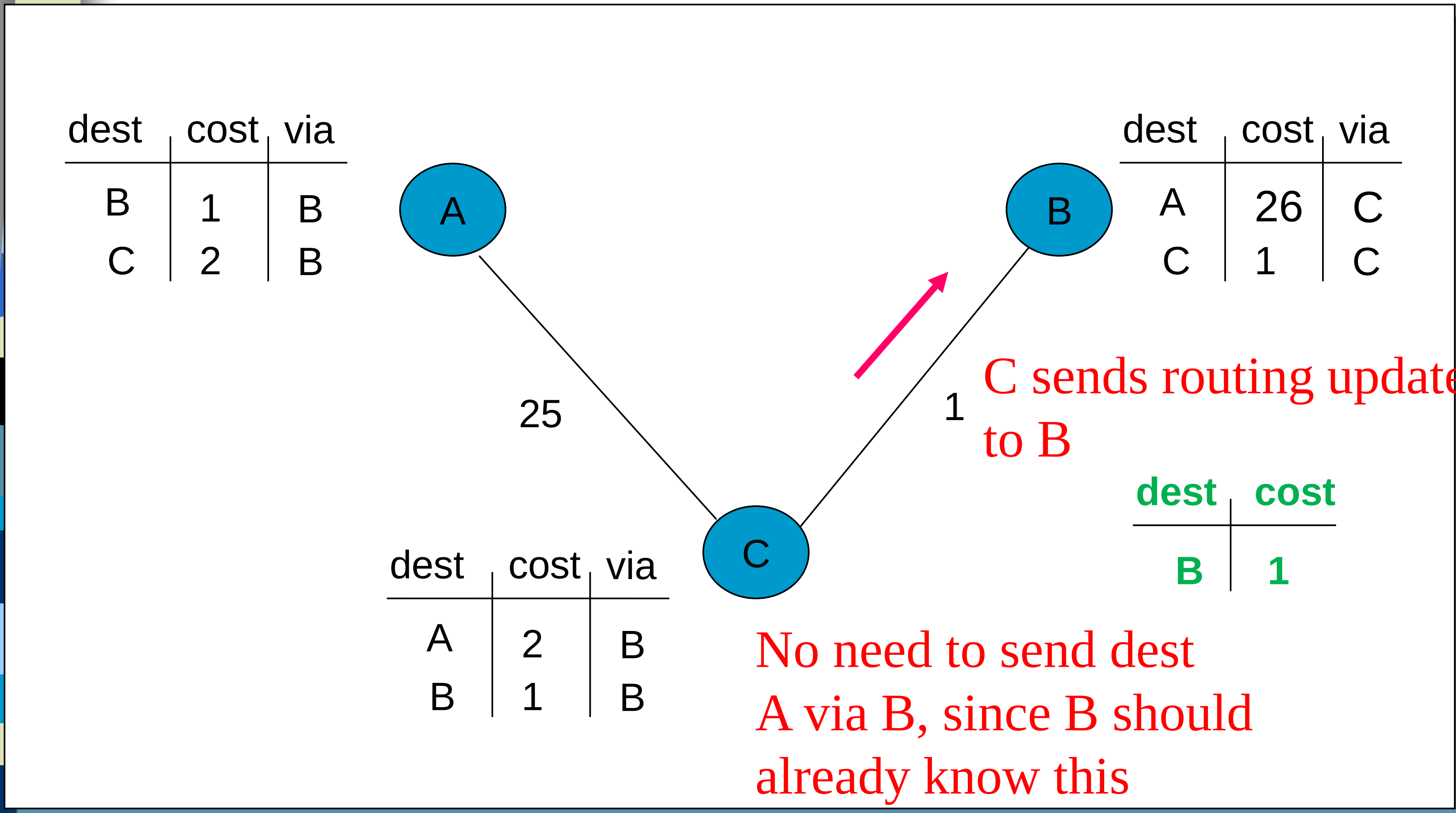
Some "Solutions"

- Split horizon
  - C does not advertise route to B when it sends its distance vector
  - A router should not advertise a network through the interface from which the update came
- Poisoned reverse
  - C advertises route to B with infinite distance in its distance vector

B Notices A-B Link Failure

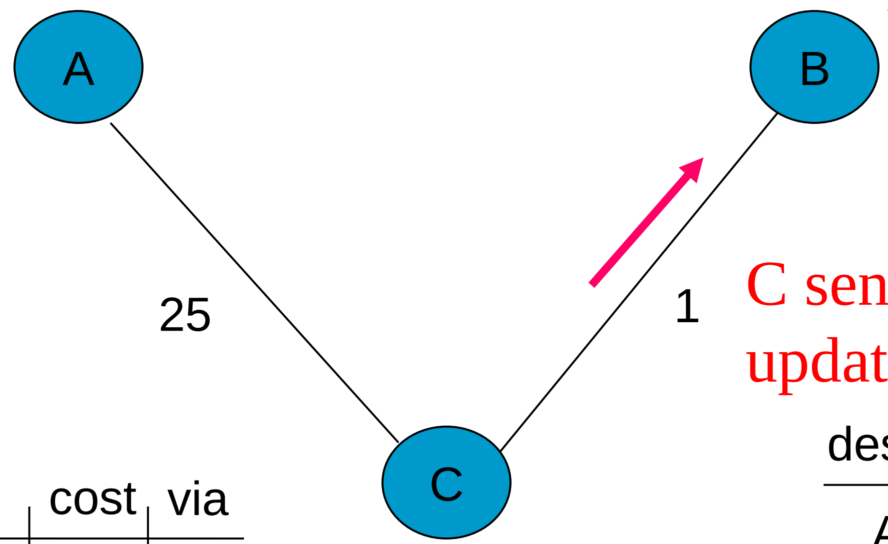


Split Horizon



Split Horizon With Poisoned Reverse

dest	cost	via
B	1	B
C	2	B



dest	cost	via
A	∞	--
C	1	C

dest	cost	via
A	2	B
B	1	B

C sends routing update to B

dest	cost
A	∞
B	1

If lowest cost path is via B, then when updating B send infinite cost

PART II

Implementing Intra-Network Relationships with OSPF

Protocoles et Interconnexions

Course Overview and Introduction

Dario Vieira
Department of Computer Science
EFREI

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

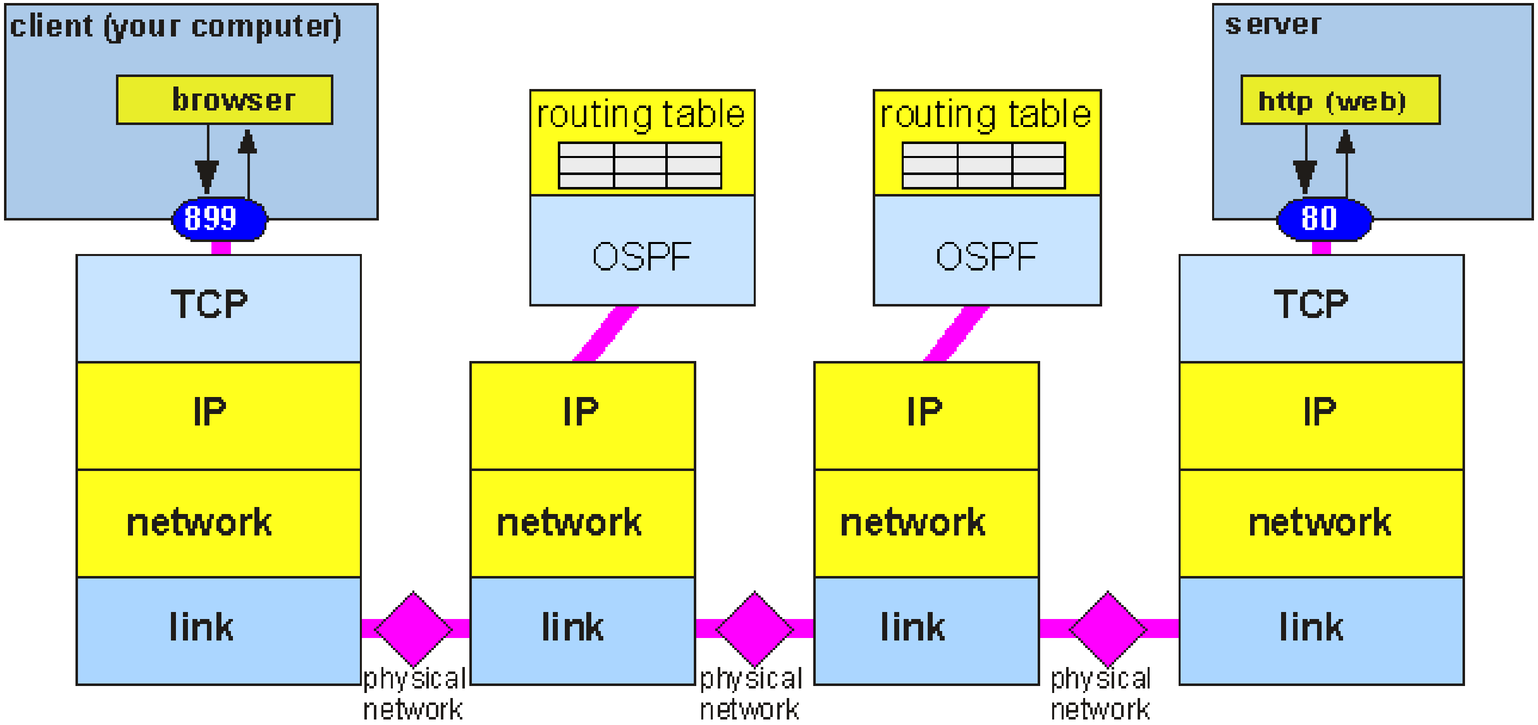
4.6 Routing in the Internet
- RIP
- OSPF
- BGP

4.7 Broadcast and multicast routing

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

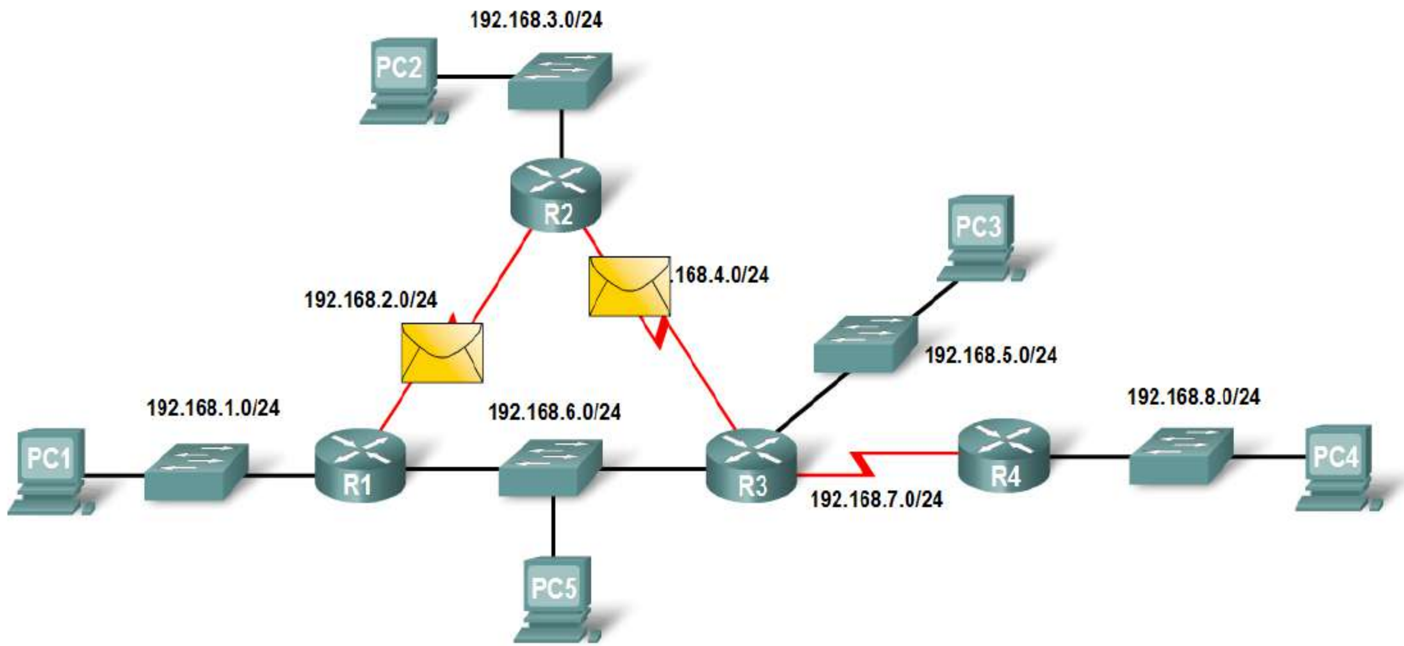
OSPF over IP


client (your computer)
browser
899
TCP
IP
network
link
physical network
routing table
OSPF
IP
network
link
physical network
routing table
OSPF
IP
network
link
physical network
server
http (web)
80
TCP
IP
network
link

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 (Type Of Service)
  - 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

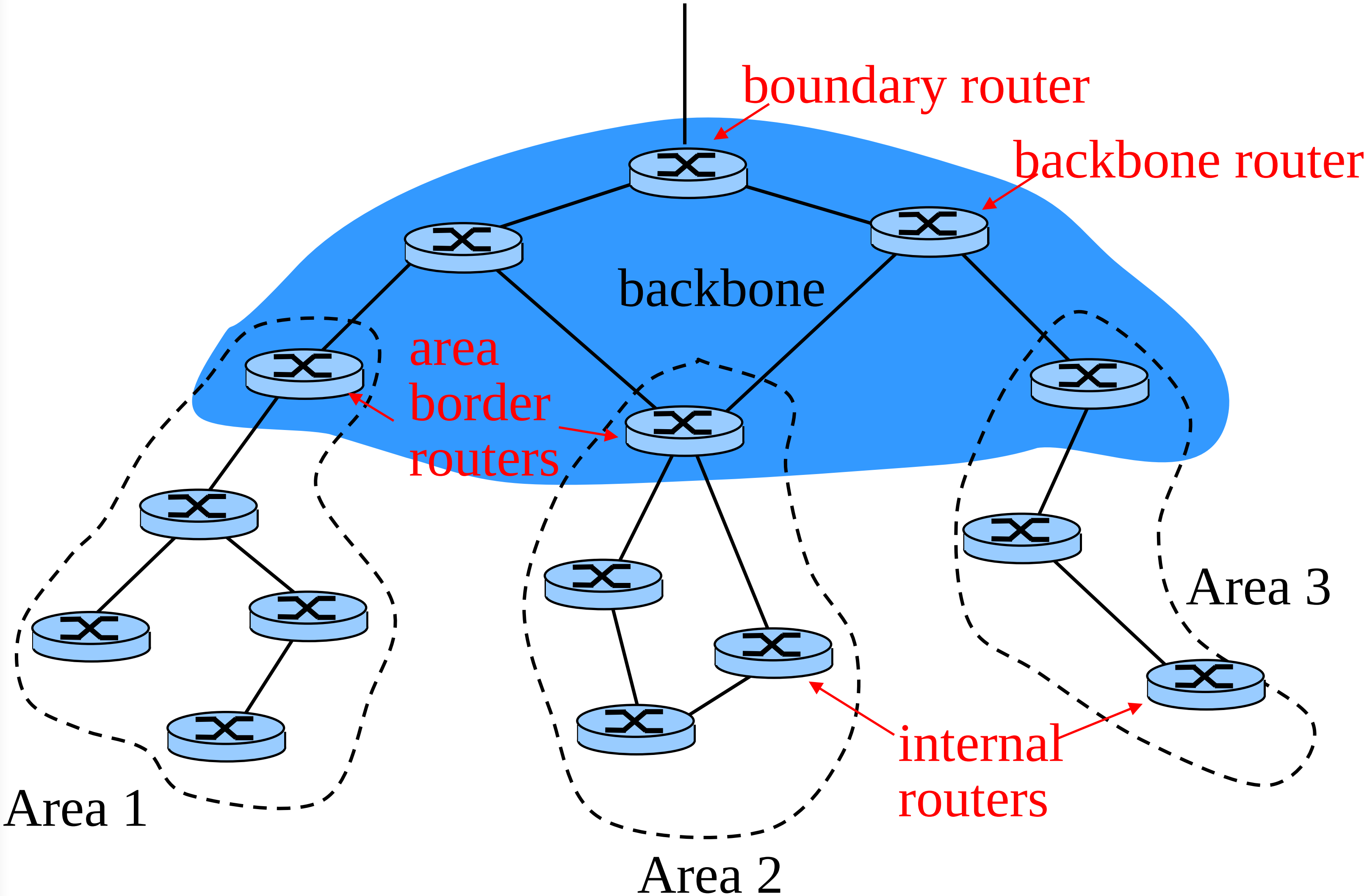
Multiple same-cost paths



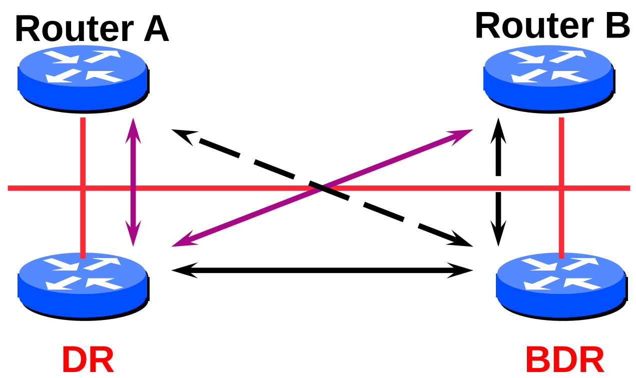

```
R2#show ip route
<output omitted>

R    192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                    [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
```

Hierarchical OSPF



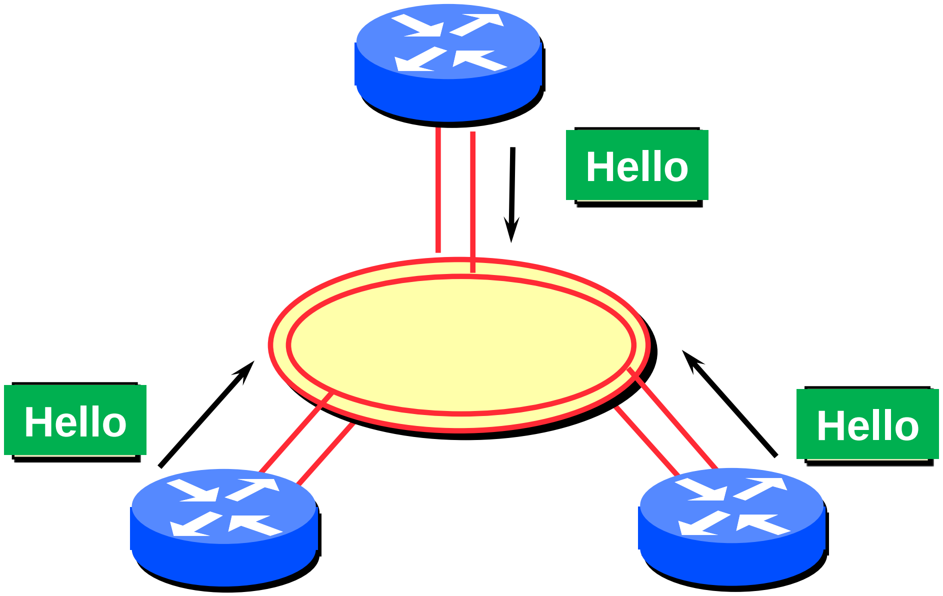
Broadcast Network: Designated Router (DR)



- Adjacencies only formed with DR (**Routeurs Désignés**) and (Backup Designated router) BDR
- All router exchanges routing information with the DR and BDR
- DR updates the database of all its neighbours
- This scales: 2n problem rather than having an n-squared problem

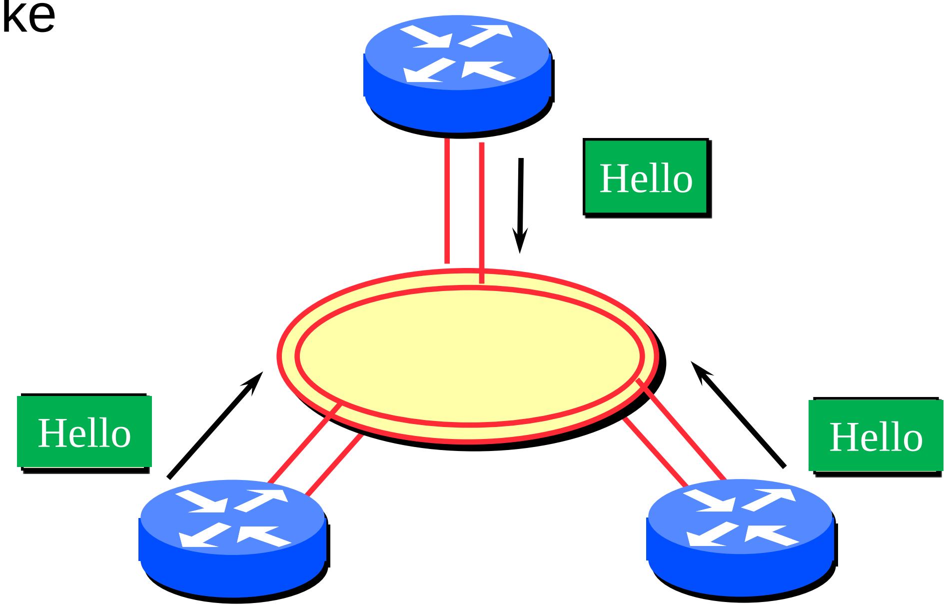
OSPF: How it works

- Hello Protocol
  - Responsible for establishing and maintaining neighbour relationships
  - Elects Designated Router on broadcast networks



The Hello Packet

- Contains information like
  - Router Priority
  - Hello Interval
  - Router dead interval
  - Network mask
  - List of neighbors

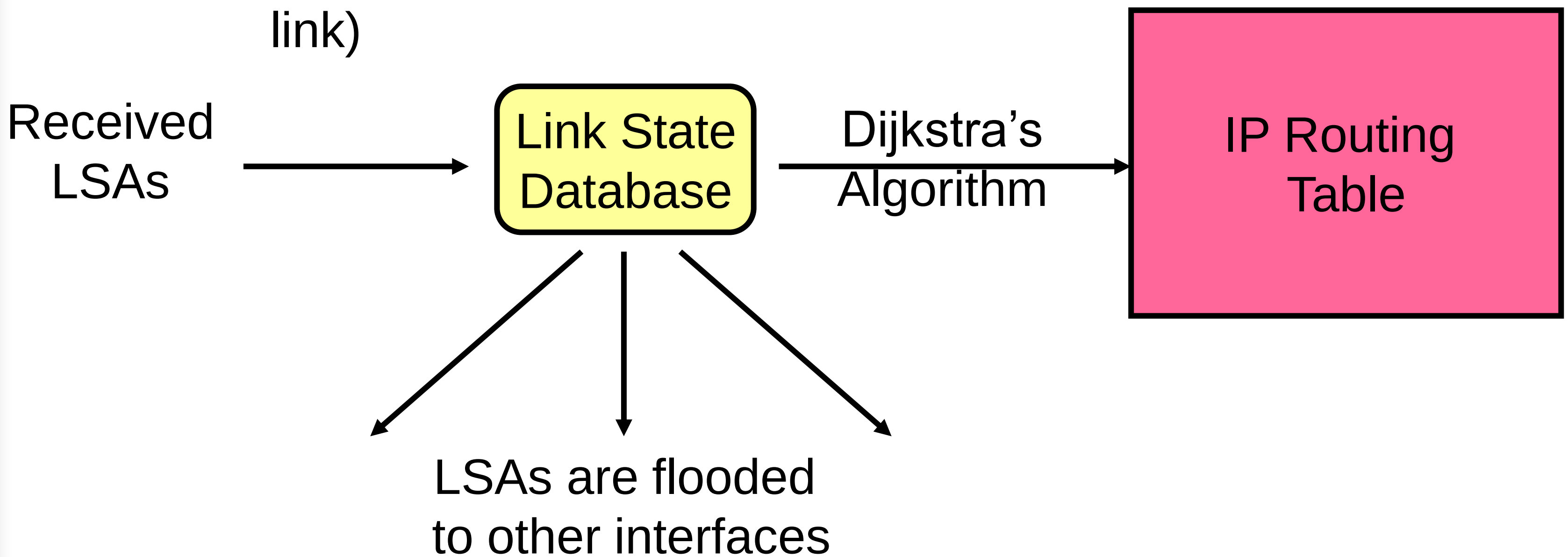


Hello Packets sent periodically on all OSPF enabled interfaces

OSPF: How it works

1. Each router establishes a relationship *("adjacency")* with its neighbors

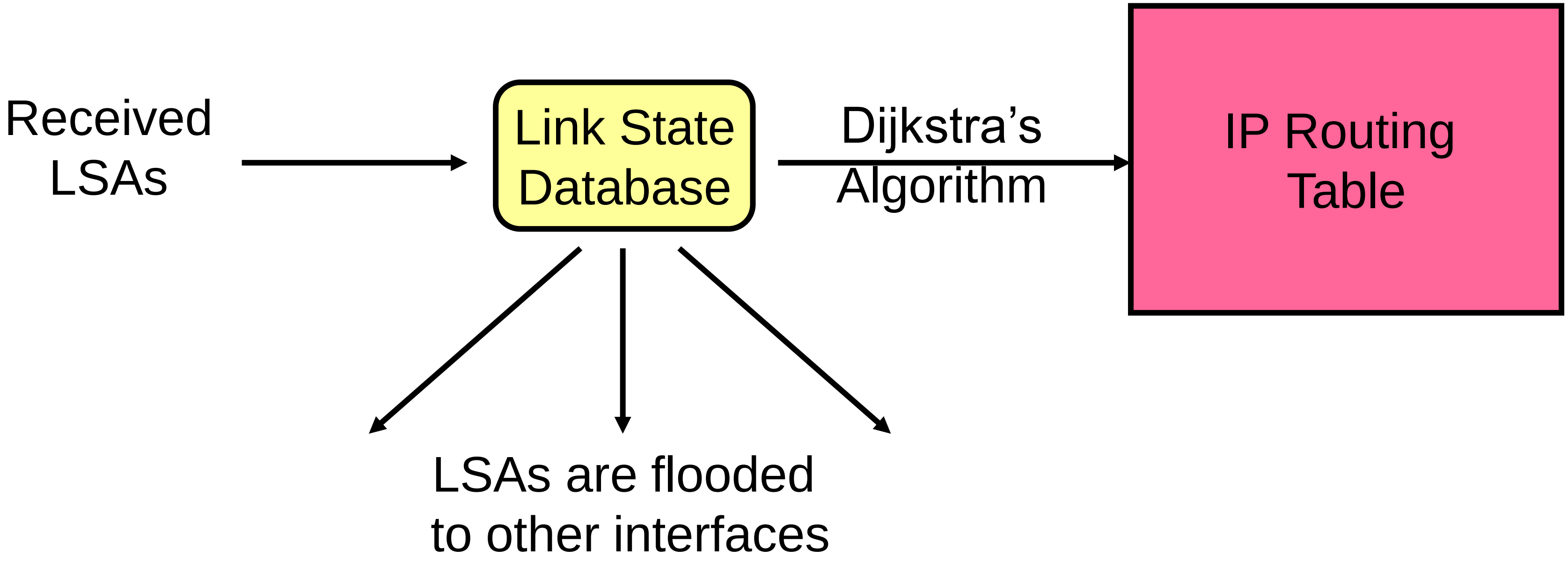
2. Each router constructs a *Link State Packet* (***LSP***) which are distributed to all neighbours
   - Using an *link state advertisements (**LSAs**)*
   - **LSA** = (link id, state of the link, cost, neighbors of the link)



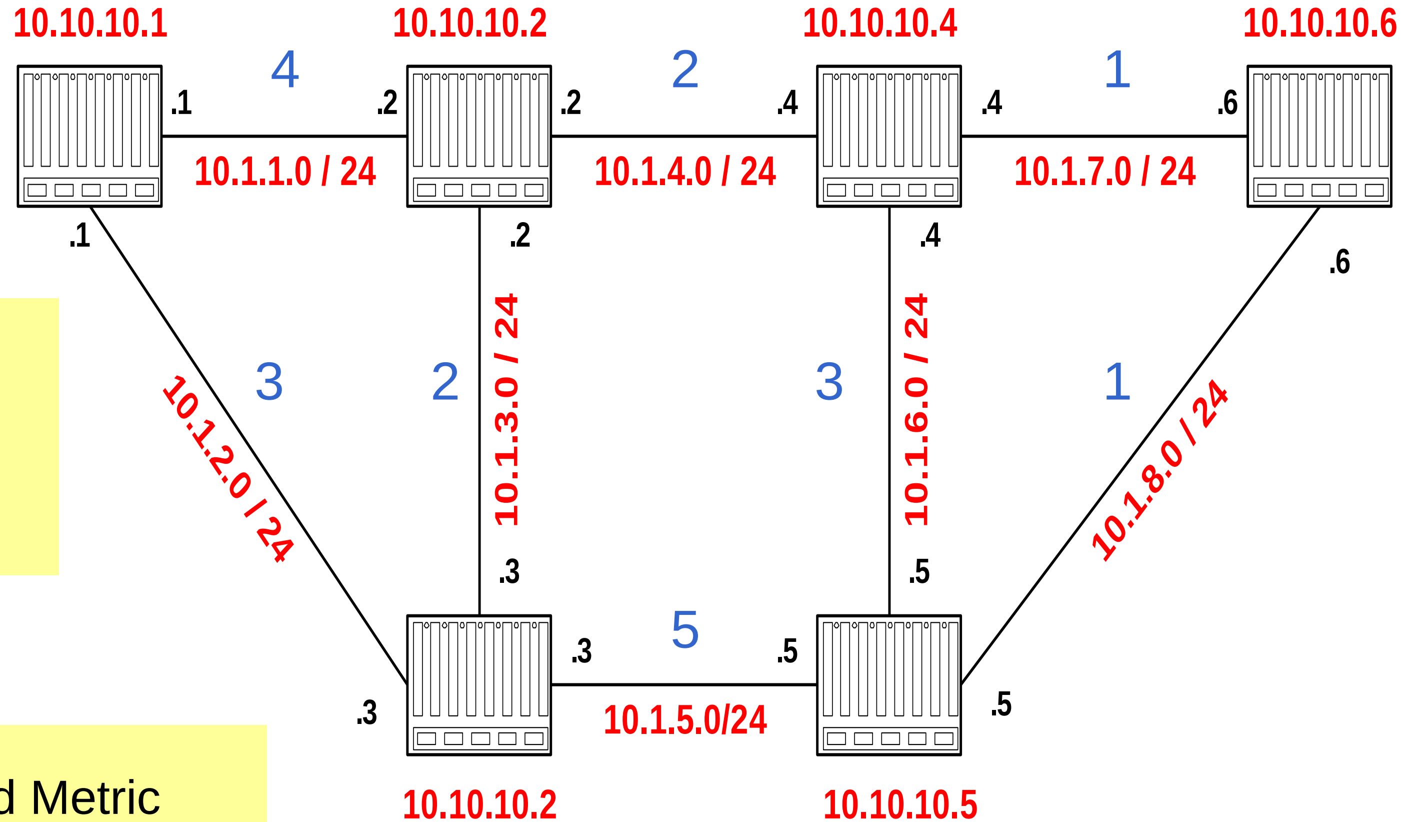
OSPF: How it works

3. Each router maintains a database of all received LSAs (*topological database* or *link state database*), which describes the network has a graph with weighted edges

4. Each router uses its link state database to run a shortest path algorithm (Dijikstra's algorithm) to produce the shortest path to each network



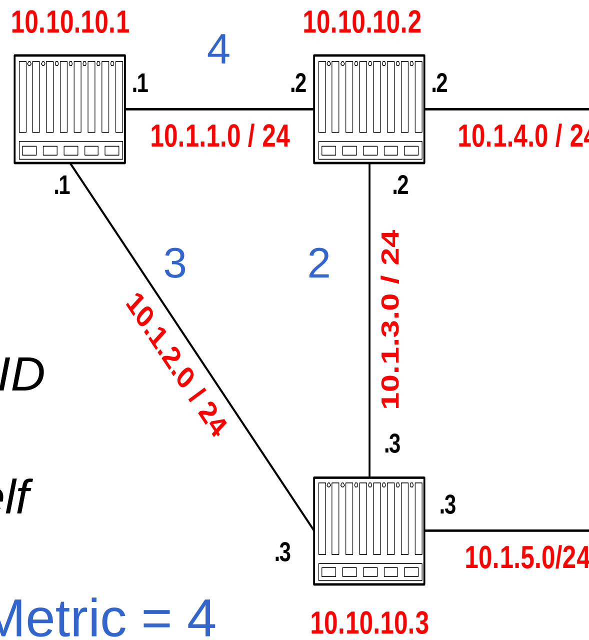
Example Network



Router IDs are selected independent of interface addresses

Link costs are called Metric
- Metric is in the range [0 , 2^{16}]
- Metric can be asymmetric

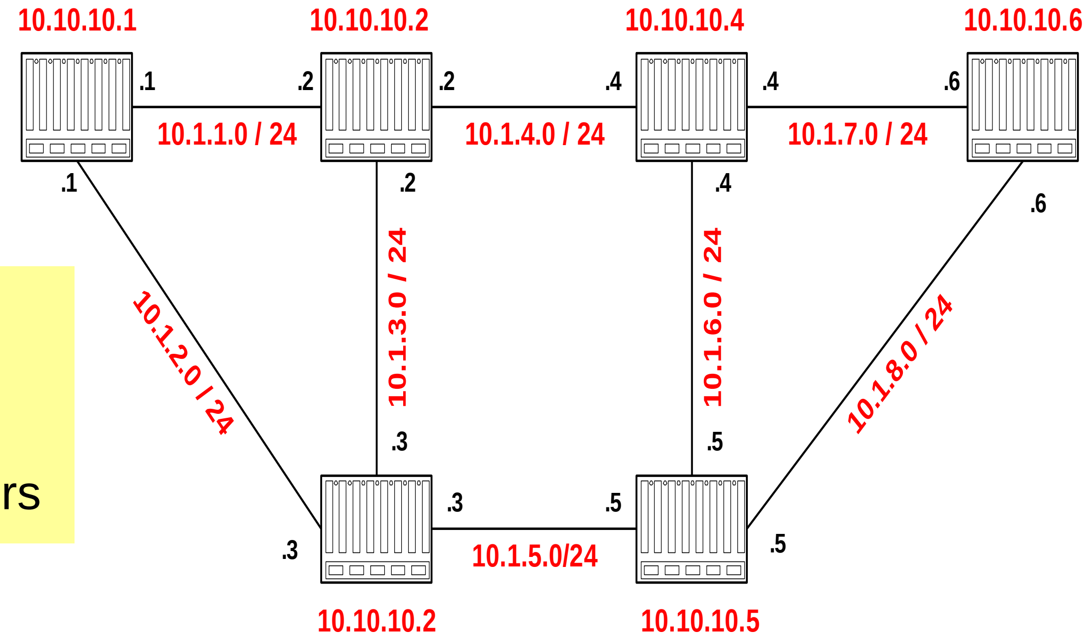
Link State Advertisement (LSA)



- **The LSA of router 10.10.10.1 is as follows:**
- **Link State ID**: 10.10.10.1 *= Router ID*
- **Advertising Router:** 10.10.10.1 *= Router ID*
- **Number of links:** 3 *= 2 links plus router itself*
- **Description of Link 1:** Link ID = 10.1.1.1, Metric = 4
- **Description of Link 2:** Link ID = 10.1.2.1, Metric = 3
- **Description of Link 3:** Link ID = 10.10.10.1, Metric = 0

Each router sends its LSA to all routers in the network (using a method called reliable flooding)

Network and Link State Database



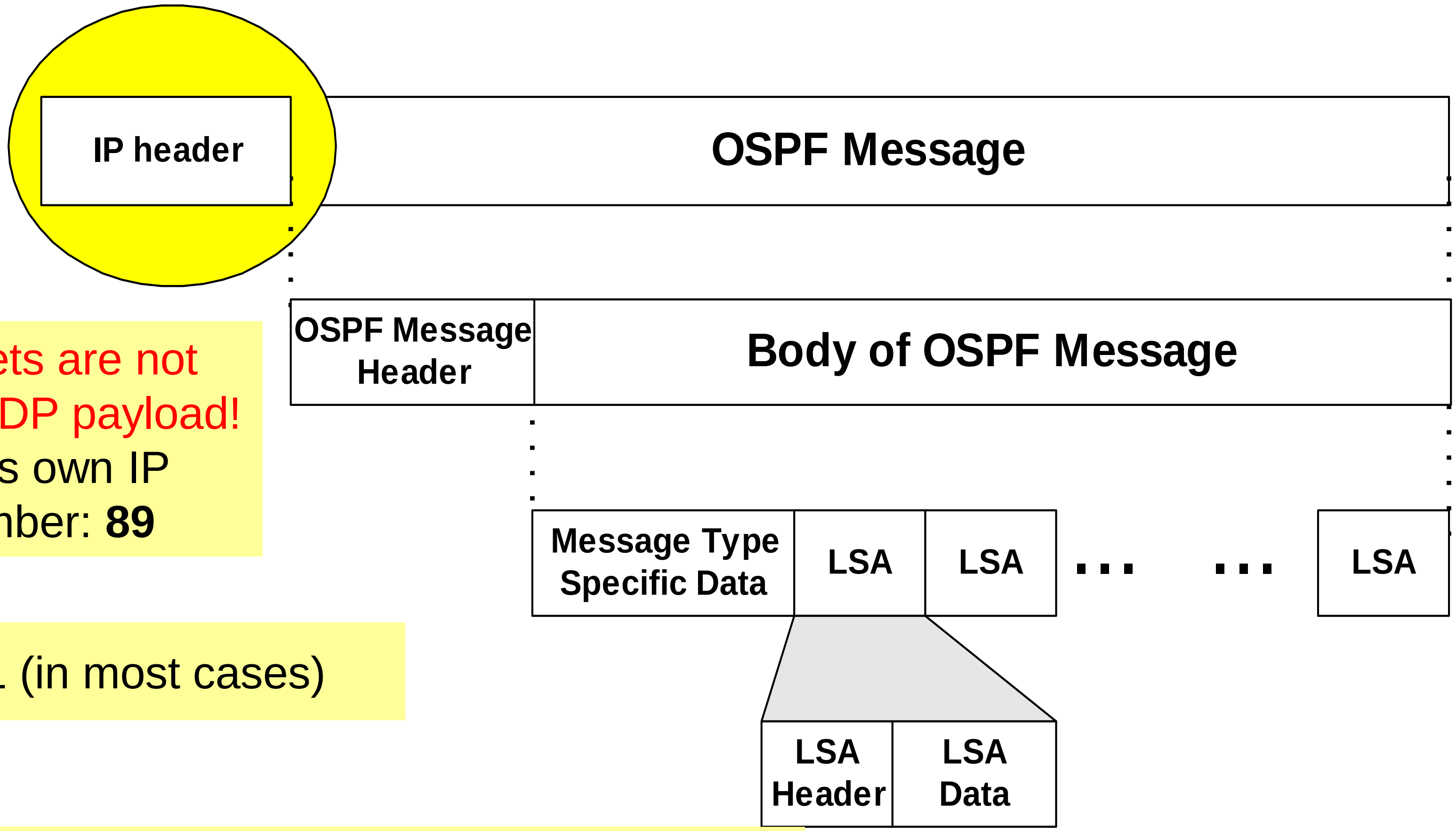
Each router has a database which contains the LSAs from all other routers

LS Type	Link StateID	Adv. Router	Checksum	LS SeqNo	LS Age
Router-LSA	10.1.10.1	10.1.10.1	0x9b47	0x80000006	0
Router-LSA	10.1.10.2	10.1.10.2	0x219e	0x80000007	1618
Router-LSA	10.1.10.3	10.1.10.3	0x6b53	0x80000003	1712
Router-LSA	10.1.10.4	10.1.10.4	0xe39a	0x8000003a	20
Router-LSA	10.1.10.5	10.1.10.5	0xd2a6	0x80000038	18
Router-LSA	10.1.10.6	10.1.10.6	0x05c3	0x80000005	1680

Link State Database

- The collection of all LSAs is called the link-state database
- Each router has an identical link-state database
  - Useful for debugging: Each router has a complete description of the network
- If neighboring routers discover each other for the first time, they will exchange their link-state databases
- The link-state databases are synchronized using reliable flooding

OSPF Packet Format

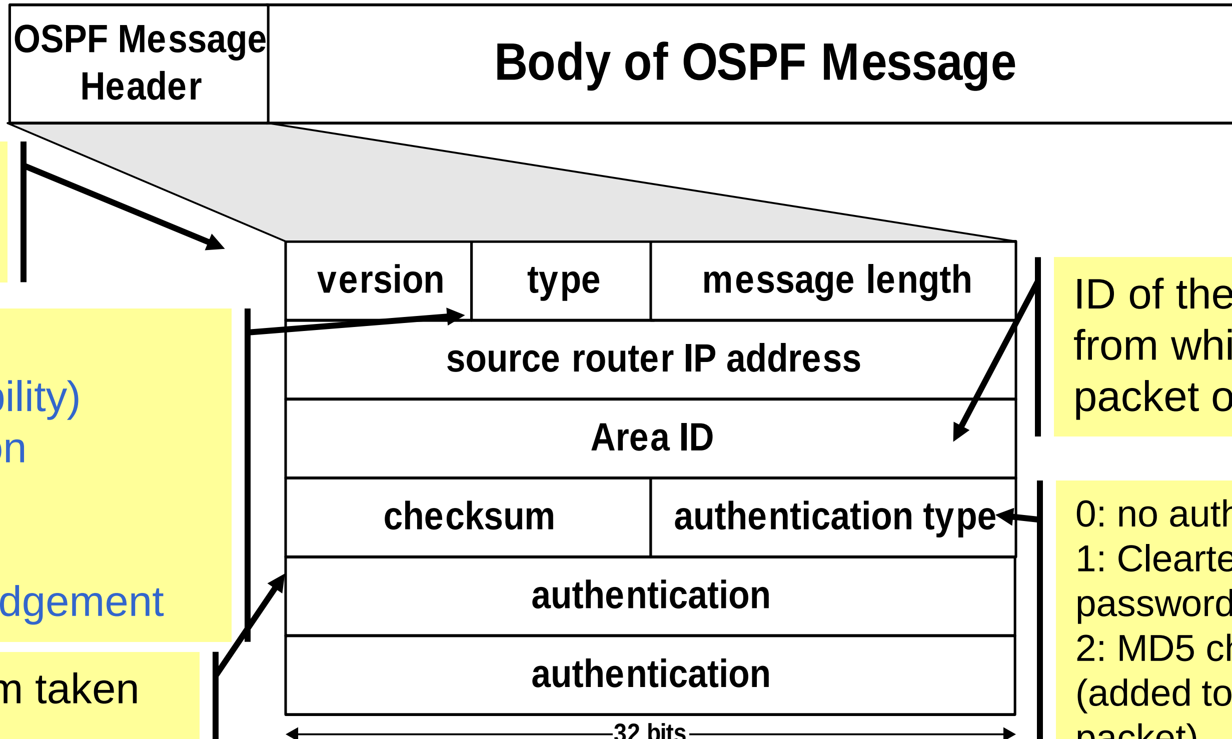


OSPF packets are not carried as UDP payload! OSPF has its own IP protocol number: **89**

TTL: set to 1 (in most cases)

Destination IP: neighbor's IP address or 224.0.0.5 (ALLSPFRouters) or 224.0.0.6 (AllDRouters)

OSPF Packet Format



2: current version is OSPF V2

Message types:
1: Hello (tests reachability)
2: Database description
3: Link Status request
4: Link state update
5: Link state acknowledgement

Standard IP checksum taken over entire packet

ID of the Area from which the packet originated

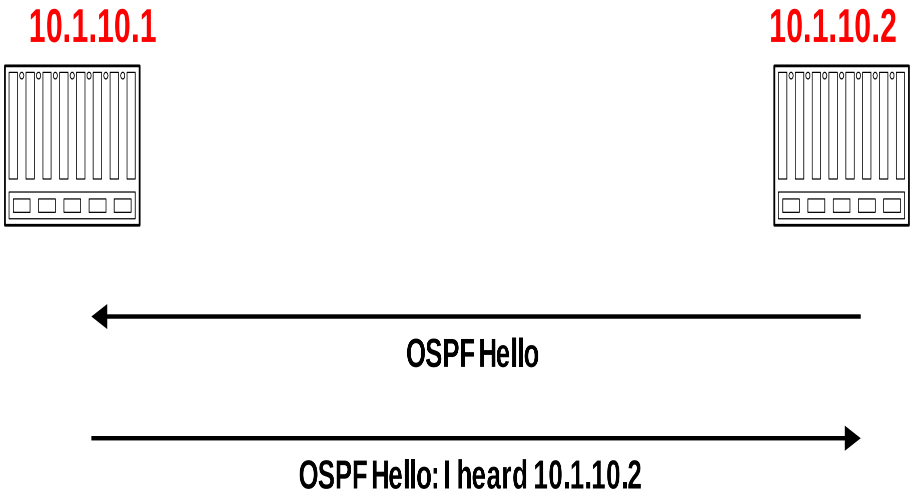
0: no authentication
1: Cleartext password
2: MD5 checksum (added to end packet)

Authentication passwd = 1:	64 cleartext password
Authentication passwd = 2:	0x0000 (16 bits)
	KeyID (8 bits)
	Length of MD5 checksum (8 bits)
	Nondecreasing sequence number (32 bits)

Prevents replay attacks

Discovery of Neighbors

- Routers multicasts OSPF Hello packets on all OSPF-enabled interfaces
- If two routers share a link, they can become neighbors, and establish an adjacency



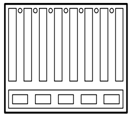
Scenario:
Router 10.1.10.2 restarts

- After becoming a neighbor, routers exchange their link state databases

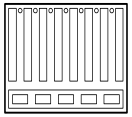
Neighbor discovery and database synchronization

Scenario:
Router 10.1.10.2 restarts

10.1.10.1



10.1.10.2



Discovery of adjacency

OSPF Hello

OSPF Hello: I heard 10.1.10.2

After neighbors are discovered the nodes exchange their databases

Database Description: Sequence = X

Sends empty database description

Sends database description. (description only contains LSA headers)

Database Description: Sequence = X, 5 LSA headers =
Router-LSA, 10.1.10.1, 0x80000006
Router-LSA, 10.1.10.2, 0x80000007
Router-LSA, 10.1.10.3, 0x80000003
Router-LSA, 10.1.10.4, 0x8000003a
Router-LSA, 10.1.10.5, 0x80000038
Router-LSA, 10.1.10.6, 0x80000005

Database description of 10.1.10.2

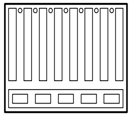
Database Description: Sequence = X+1, 1 LSA header=
Router-LSA, 10.1.10.2, 0x80000005

Acknowledges receipt of description

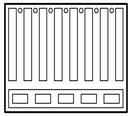
Database Description: Sequence = X+1

Regular LSA exchanges

10.1.10.1



10.1.10.2



Link State Request packets, LSAs =
Router-LSA, 10.1.10.1,
Router-LSA, 10.1.10.2,
Router-LSA, 10.1.10.3,
Router-LSA, 10.1.10.4,
Router-LSA, 10.1.10.5,
Router-LSA, 10.1.10.6,

10.1.10.2 explicitly requests each LSA from 10.1.10.1

10.1.10.1 sends requested LSAs

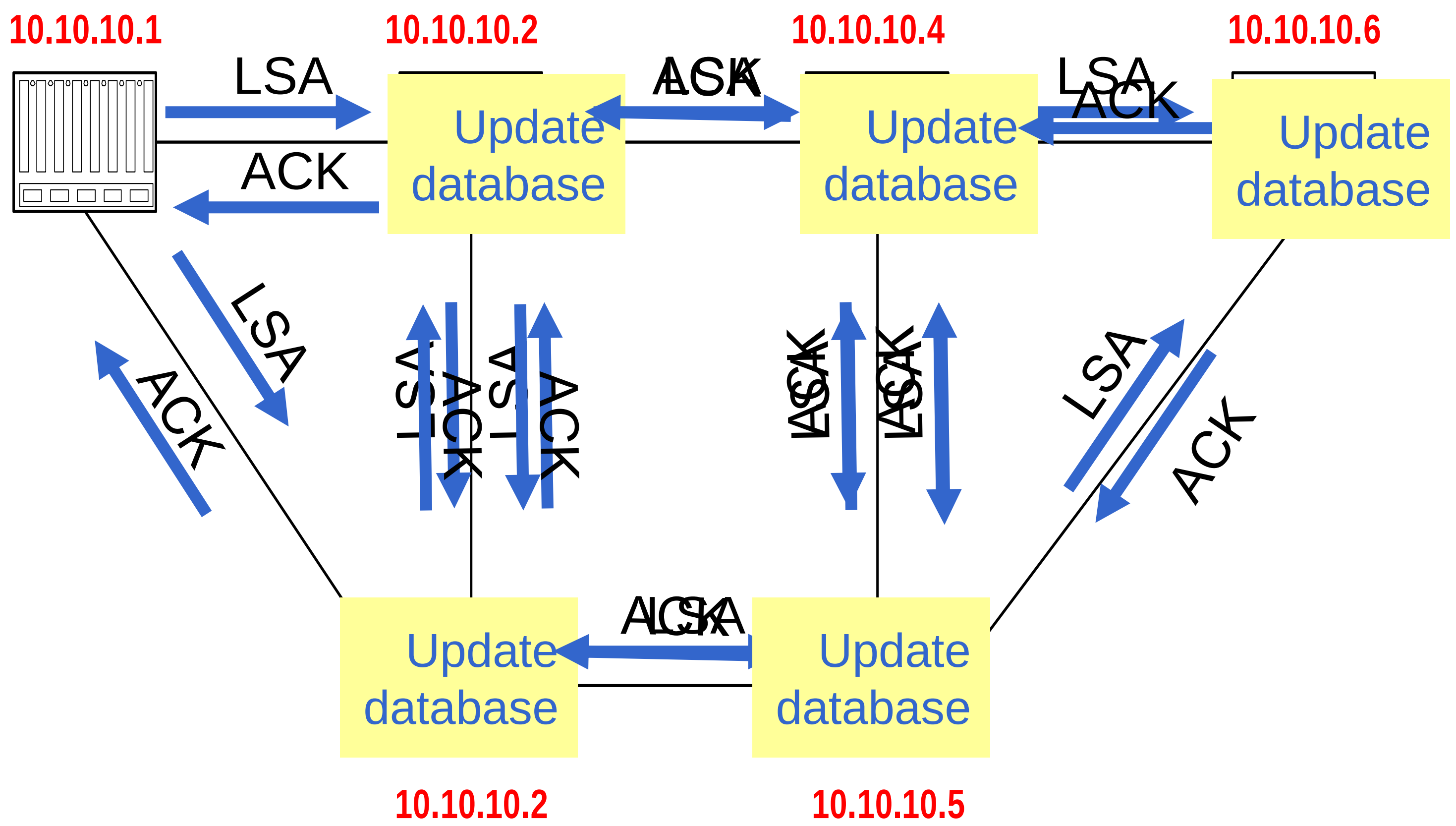
Link State Update Packet, LSAs =
Router-LSA, 10.1.10.1, 0x80000006
Router-LSA, 10.1.10.2, 0x80000007
Router-LSA, 10.1.10.3, 0x80000003
Router-LSA, 10.1.10.4, 0x8000003a
Router-LSA, 10.1.10.5, 0x80000038
Router-LSA, 10.1.10.6, 0x80000005

10.1.10.2 has more recent value for 10.0.1.6 and sends it to 10.1.10.1 (with higher sequence number)

Link State Update Packet, LSA =
Router-LSA, 10.1.1.6, 0x80000006

Routing Data Distribution

- LSA-Updates are distributed to all other routers via **Reliable Flooding**
- **Example:** Flooding of LSA from 10.10.10.1



Dissemination of LSA-Update

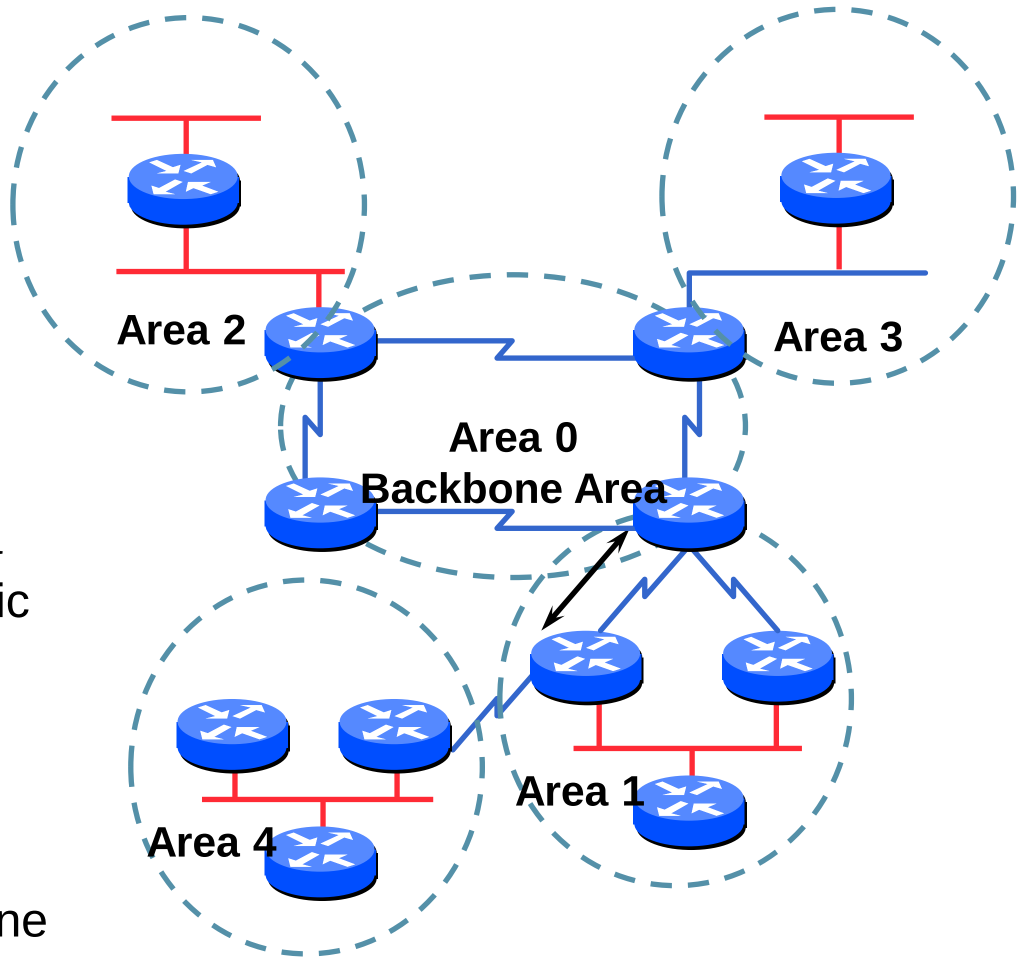
- A router sends and refloods LSA-Updates, whenever the topology or link cost changes (If a received LSA does not contain new information, the router will not flood the packet)

- Exception: Infrequently (every 30 minutes), a router will flood LSAs even if there are not new changes

- Acknowledgements of LSA-updates:
    - explicit ACK, or
    - implicit via reception of an LSA-Update

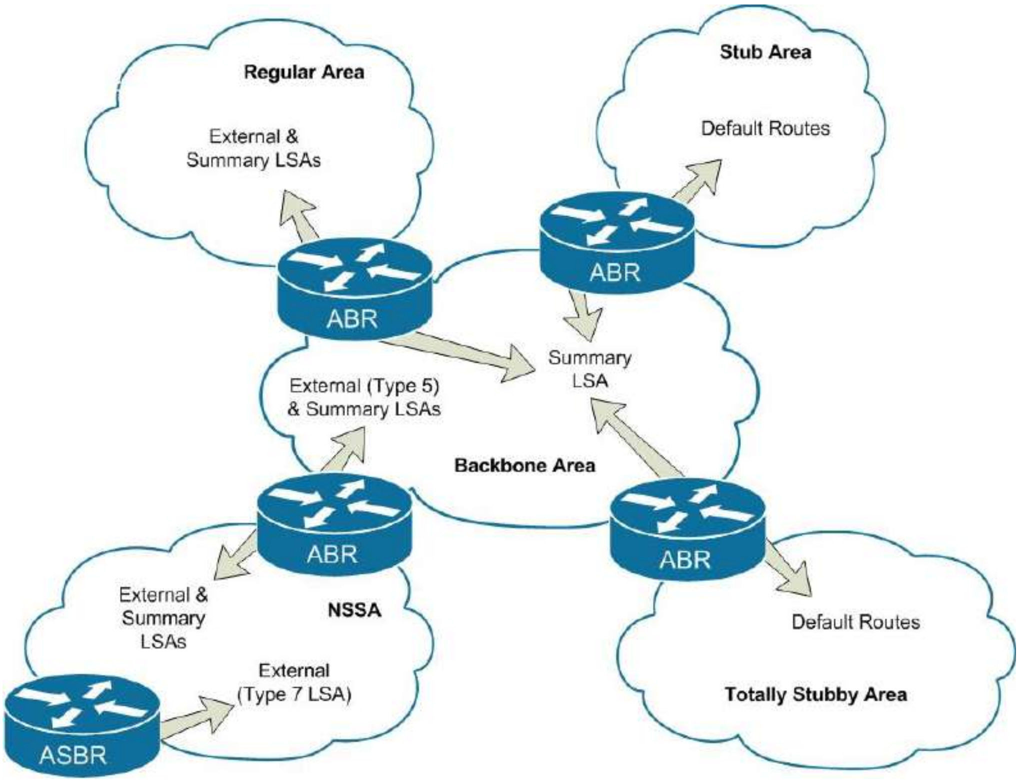
OSPF Areas

- Group of contiguous hosts and networks
- Per area topological database
  - Invisible outside the area
  - Reduction in routing traffic
- Backbone area contiguous
  - All other areas must be connected to the backbone

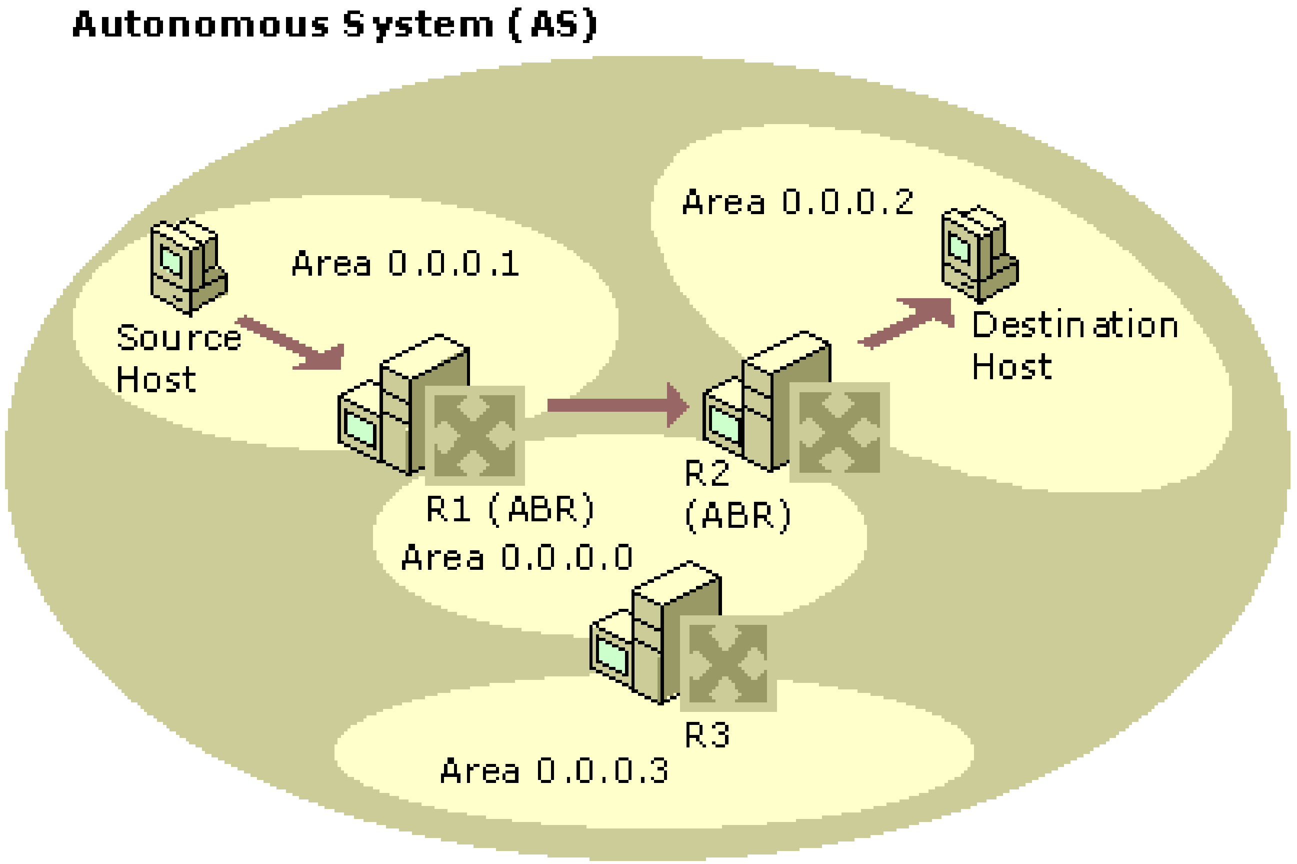


Exemplo

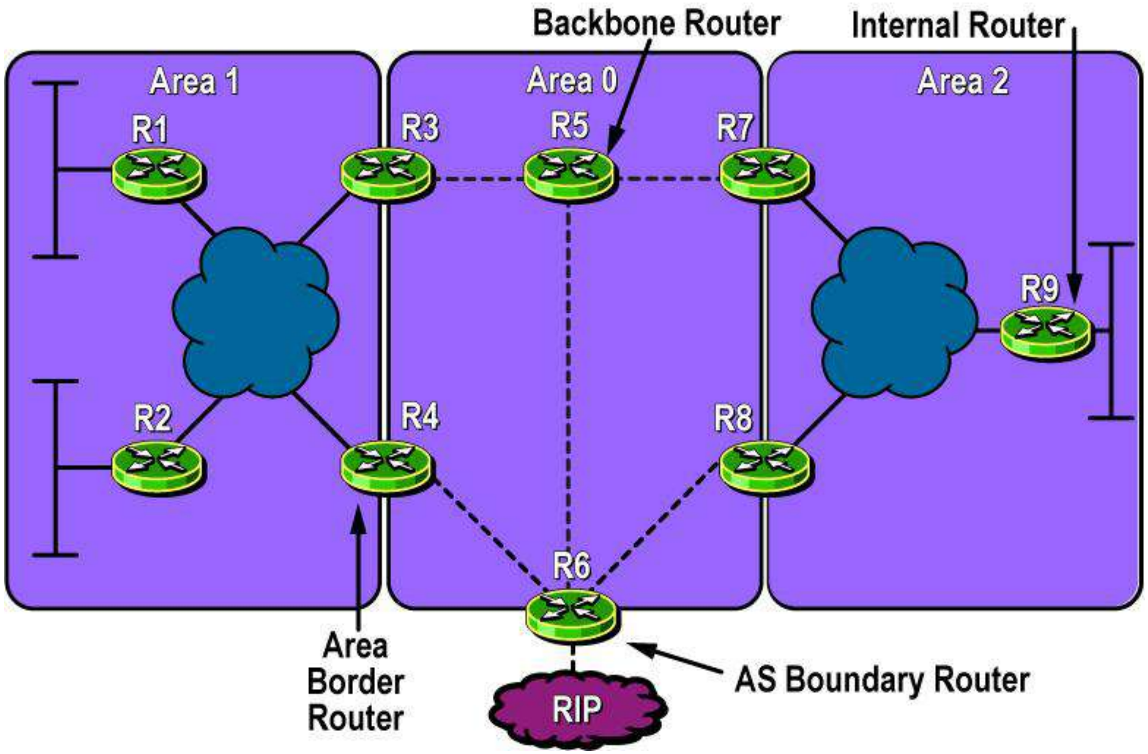
R1(config)# router ospf 10
R1(config-router)# **network 10.0.0.0 0.255.255.255 area 0**
R1(config-router)# **network 192.168.100.0 0.0.3.255 area 0**
R5(config)# router ospf 10
R5(config-router)# **area 2 stub**



Exemplo

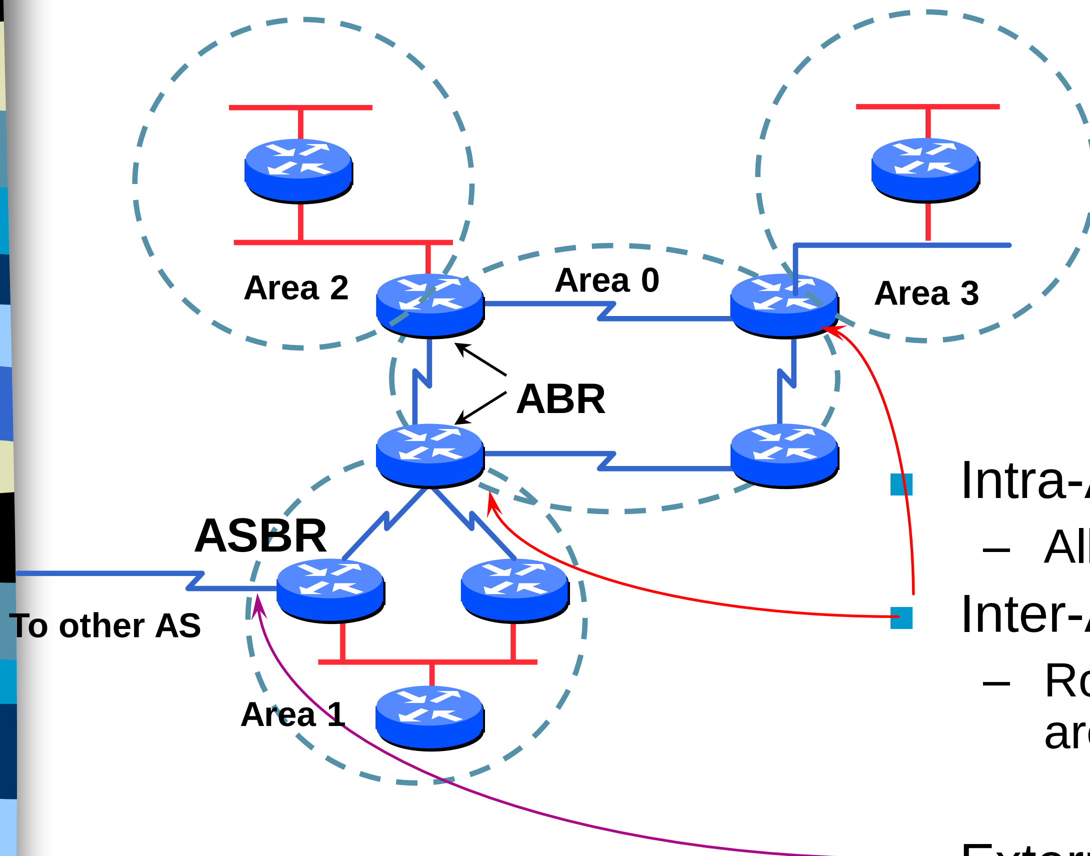

Autonomous System (AS)
Area 0.0.0.1
Source Host
Area 0.0.0.2
Destination Host
R1 (ABR)
R2 (ABR)
Area 0.0.0.0
R3
Area 0.0.0.3

Router Roles



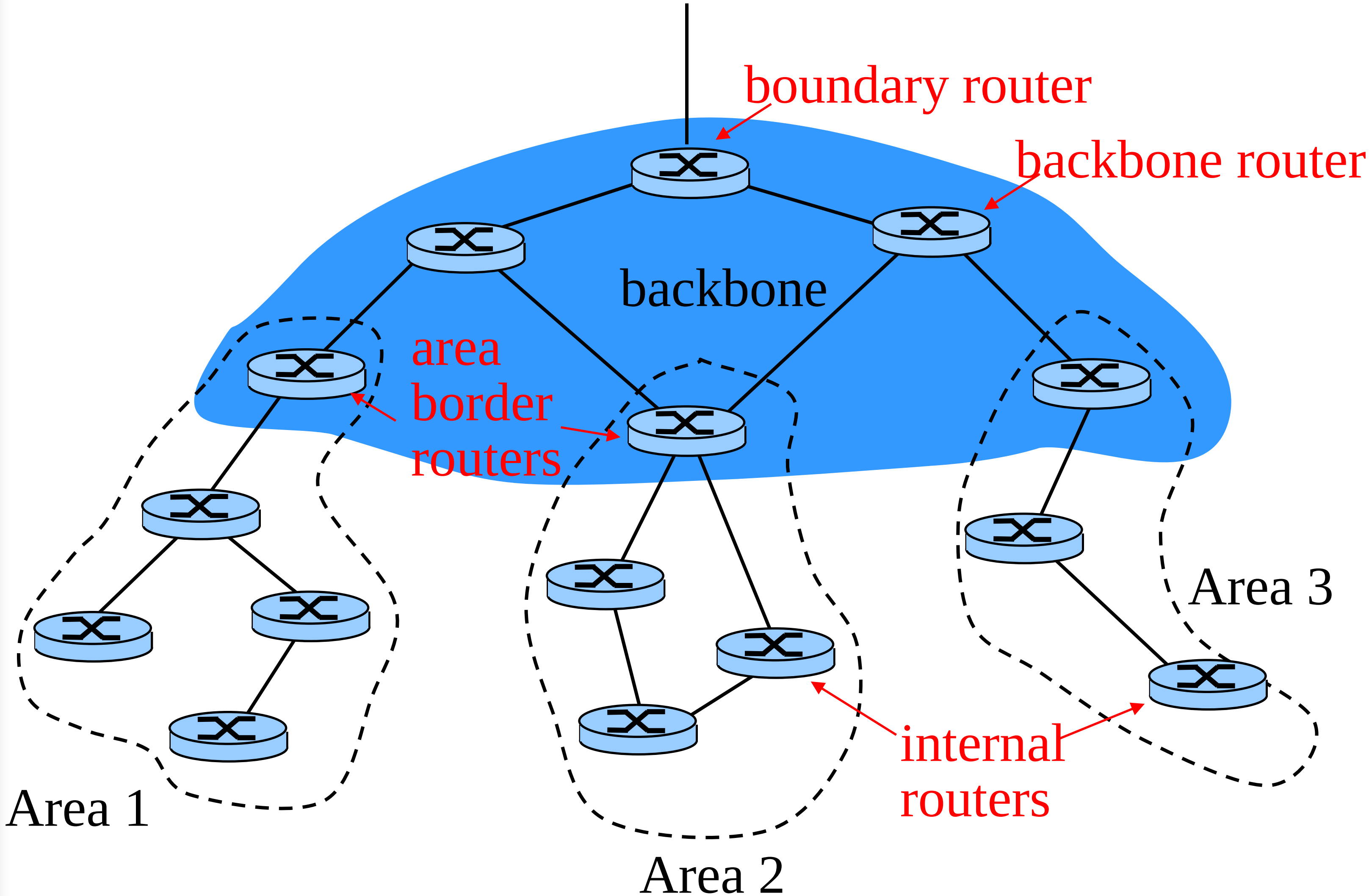
- **Internal:** All interfaces in a single area (routers 1,2,5 and 9 in diagram above)
- **Backbone:** At least one interface assigned to area 0 (routers 3,4, 5, 6, 7, 8, in diagram above)
- **Area Border Router (ABR):** Have interfaces in two or more areas (routers 1,4,5 in diagram above)
- **Autonomous System Boundary Router (ASBR):** Has at least one interface in an OSPF area and at least one interface outside of an OSPF area.

OSPF Route Types



- Intra-Area route
  - All routes inside an area
- Inter-Area route
  - Routes advertised from one area to another area by an ABR
- External route
  - Routes imported into OSPF from another routing protocol by an ASBR

Hierarchical OSPF



PART III

Implementing Inter-Network Relationships with BGP

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