CS 540
Computer Networks II

Sandy Wang
sandy.w@svuca.edu

6. ROUTING PROTOCOLS – RIP & OSPF

Topics

1. Overview
2. LAN Switching
3. IPv4
4. IPv6
5. Tunnels
6. **Routing Protocols -- RIP, RIPng**
7. Routing Protocols -- OSPF
8. IS-IS
9. Midterm Exam
10. BGP
11. MPLS
12. Transport Layer -- TCP/UDP
13. Congestion Control & Quality of Service (QoS)
14. Access Control List (ACL)
15. Final Exam

Reference Books

- **Routing TCP/IP Volume I, 2nd Edition** by Jeff Doyle and Jennifer Carroll

  ISBN: 1-57870-089-2
- **Routing TCP/IP Volume II** by Jeff Doyle and Jennifer DeHaven
  ISBN: 1-57870-089-2
- **Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide, Academic Edition** by Wendel Odom -- July 10, 2013**.**
  ISBN-13: 978-1587144882
- **The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference** by Charles M. Kozierok – October 1, 2005.
  ISBN-13: 978-1593270476
- **CCNA Routing and Switching 200-120 Network Simulator.** By Wendell Odom, Sean Wilkins. Published by Pearson IT Certification.
- http://class.svuca.edu/~sandy/class/CS540/

Topics:

- RIP
- OSPF

Introduction

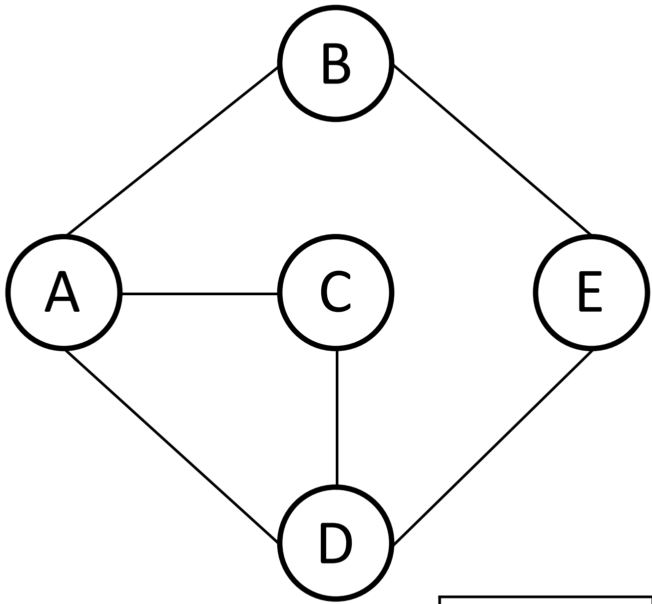
IPv4 original address space

/xx

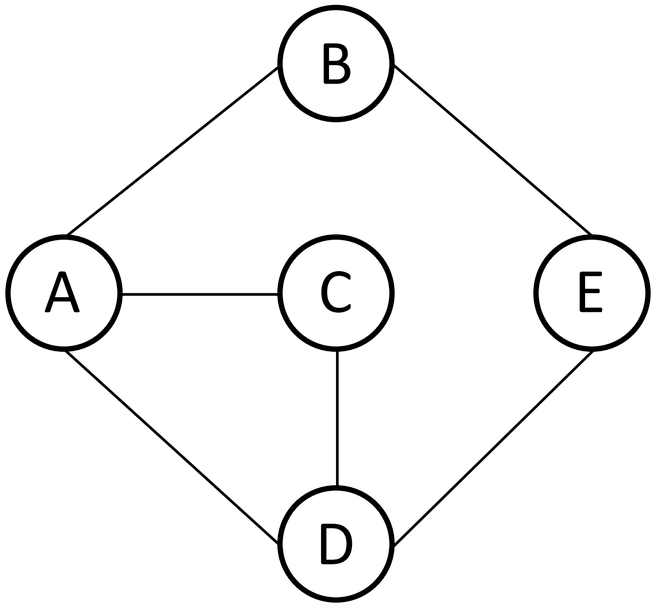
	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector Routing Protocols		Link State Routing Protocols		Path Vector
Classful	**RIP**	**IGRP**			**EGP**
Classless	**RIPv2**	**EIGRP**	**OSPFv2**	**IS-IS**	**BGPv4**
IPv6	**RIPng**	**EIGRP for IPv6**	**OSPFv3**	**IS-IS for IPv6**	**BGPv4 for IPv6**

In this chapter, you will learn to:

- Describe the background and basic features of OSPF.
- Identify and apply the basic OSPF configuration commands.
- Describe, modify and calculate the metric used by OSPF.
- Describe the Designated Router/Backup Designated Router (DR/BDR) election proess in multiaccess networks.
- Employ the `default-information originate` command to configure and propagate a default route in OSPF.



	A	B	C	D	E
A	0	1	1	1	∞
B	1	0	∞	∞	1
C	1	∞	0	1	∞
D	1	∞	1	0	1
E	∞	1	∞	1	0



A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	∞	?

B: Routing Table

	Cost	Nexthop
A	1	A
B	0	--
C	∞	**?**
D	∞	**?**
E	**1**	E

C: Routing Table

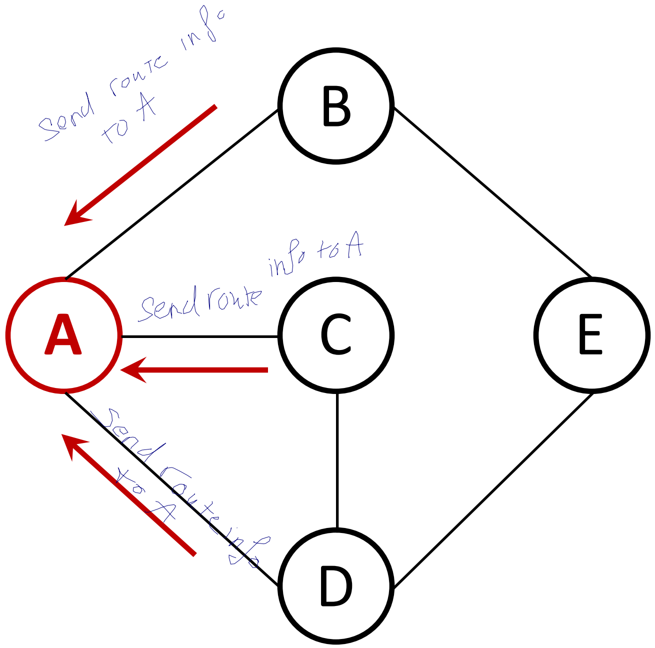
	Cost	Nexthop
A	1	A
B	∞	?
C	0	**--**
D	1	**D**
E	∞	?

D: Routing Table

	Cost	Nexthop
A	1	A
B	∞	?
C	1	**C**
D	**0**	**--**
E	**1**	E

E: Routing Table

	Cost	Nexthop
A	∞	?
B	1	B
C	∞	**?**
D	1	**D**
E	**0**	--



A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**∞** 1+1	? B 1 D

B → A

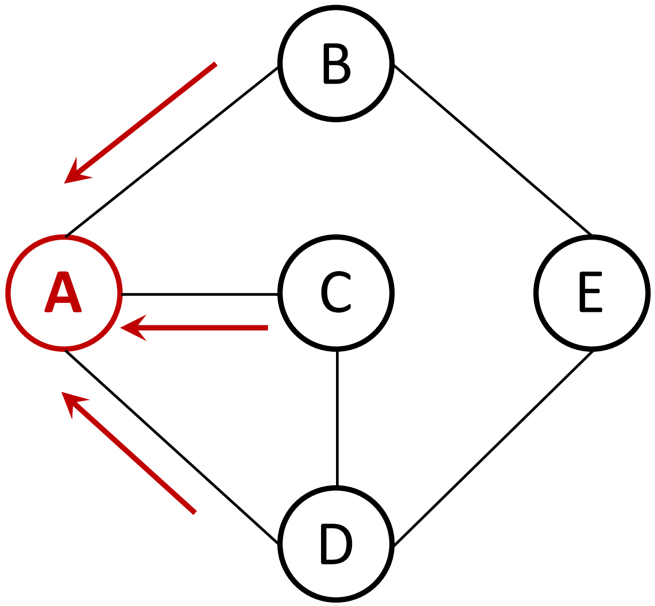
	Cost	Nexthop
A	1	A
B	0	--
C	**∞**	**?**
D	**∞**	**?**
E	**1**	E

C → A

	Cost	Nexthop
A	1	A
B	**∞**	?
C	0	**--**
D	1	**D**
E	**∞**	?

D → A

	Cost	Nexthop
A	1	A
B	**∞**	?
C	1	**C**
D	**0**	**--**
E	**1**	E



A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**∞**	?

\+

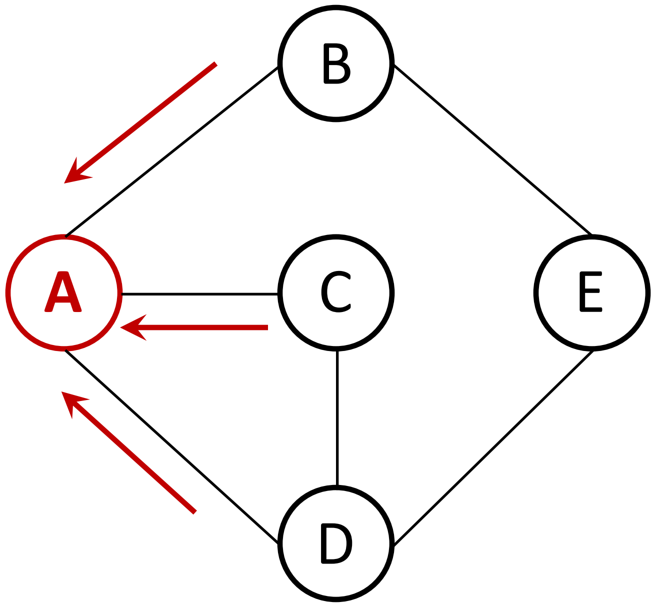
B → A

	Cost	Nexthop
A	1	A
B	0	--
C	**∞**	**?**
D	**∞**	**?**
E	**1**	E

=

A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B



A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B

+

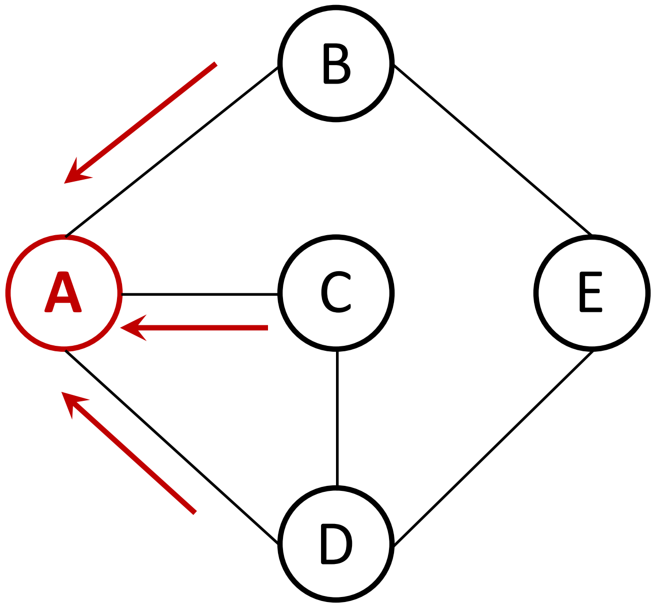
C → A

	Cost	Nexthop
A	1	A
B	**∞**	?
C	0	**--**
D	1	**D**
E	**∞**	?

=

A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B



A: Routing Table

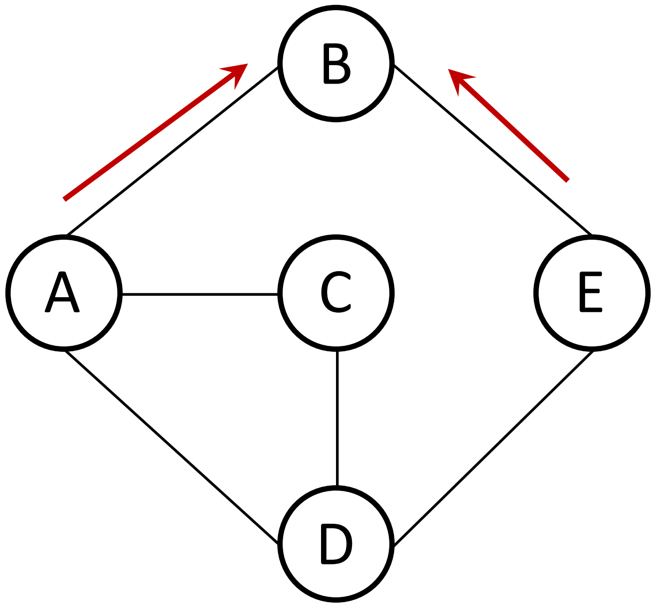
	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B

D → A

	Cost	Nexthop
A	1	A
B	**∞**	?
C	1	**C**
D	**0**	**--**
E	**1**	E

A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B, D



B: Routing Table

	Cost	Nexthop
A	1	A
B	0	--
C	**∞**	**?**
D	**∞**	**?**
E	**1**	E

B: Routing Table

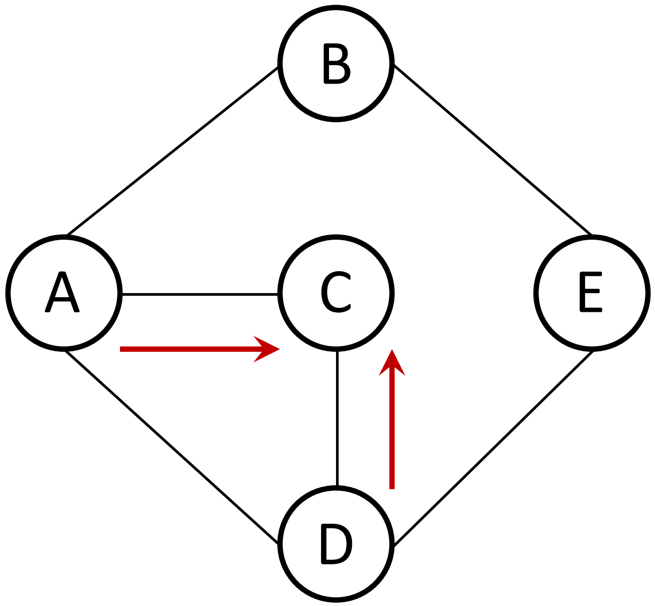
	Cost	Nexthop
A	1	A
B	0	--
C	**2**	**A**
D	**2**	**A, E**
E	**1**	E

A → B

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**∞**	?

E → B

	Cost	Nexthop
A	**∞**	?
B	1	B
C	**∞**	**?**
D	1	**D**
E	**0**	--



C: Routing Table

	Cost	Nexthop
A	1	A
B	**∞**	?
C	0	**--**
D	1	**D**
E	**∞**	?

C: Routing Table

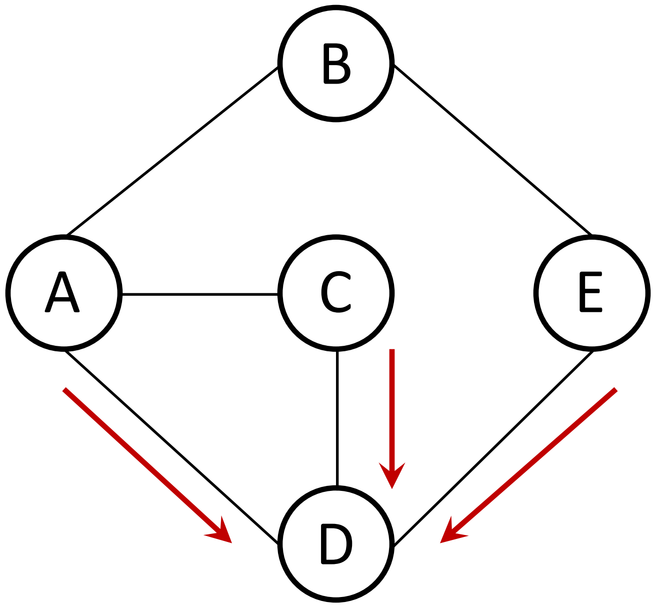
	Cost	Nexthop
A	1	A
B	**2**	A
C	0	**--**
D	1	**D**
E	**2**	D

A → C

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**∞**	?

D → C

	Cost	Nexthop
A	1	A
B	**∞**	?
C	1	**C**
D	**0**	**--**
E	**1**	E



D: Routing Table

	Cost	Nexthop
A	1	A
B	**∞**	?
C	1	**C**
D	**0**	**--**
E	**1**	E

D: Routing Table

	Cost	Nexthop
A	1	A
B	**2**	A, E
C	1	**C**
D	**0**	**--**
E	**1**	E

A → D

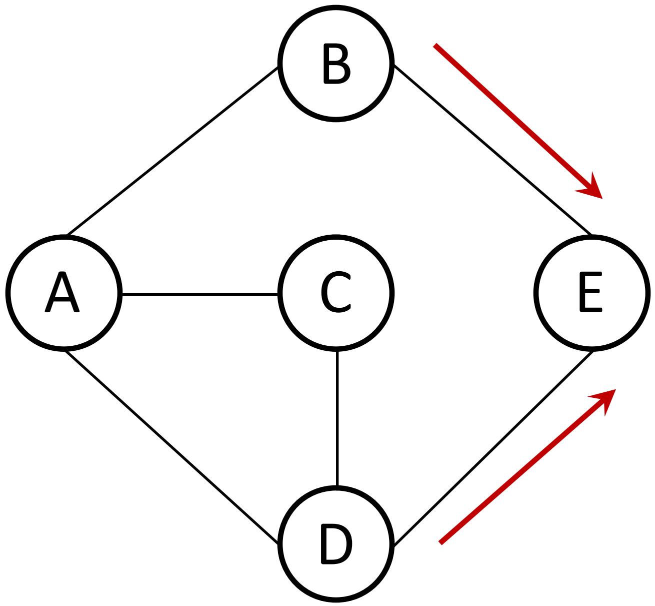
	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**∞**	?

C → D

	Cost	Nexthop
A	1	A
B	**∞**	?
C	0	**--**
D	1	**D**
E	**∞**	?

E → D

	Cost	Nexthop
A	**∞**	?
B	1	B
C	**∞**	**?**
D	1	**D**
E	**0**	--



E: Routing Table

	Cost	Nexthop
A	**∞**	?
B	1	B
C	**∞**	**?**
D	1	**D**
E	**0**	--

E: Routing Table

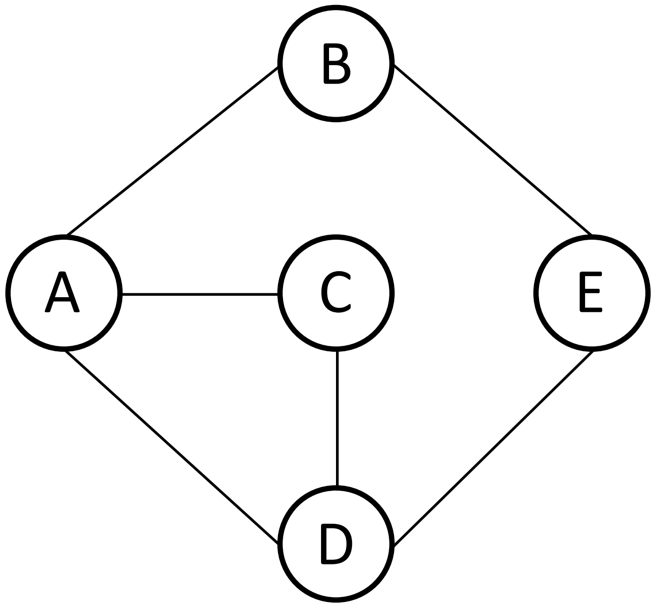
	Cost	Nexthop
A	**2**	B, D
B	1	B
C	**2**	**D**
D	1	**D**
E	**0**	--

B → E

	Cost	Nexthop
A	1	A
B	0	--
C	**∞**	**?**
D	**∞**	**?**
E	**1**	E

D → E

	Cost	Nexthop
A	1	A
B	**∞**	?
C	1	**C**
D	**∞**	**?**
E	**1**	E



A: Routing Table

	Cost	Nexthop
A	0	--
B	1	B
C	1	**C**
D	1	**D**
E	**2**	B, D

B: Routing Table

	Cost	Nexthop
A	1	A
B	0	--
C	**2**	**A**
D	**2**	**A, E**
E	**1**	E

C: Routing Table

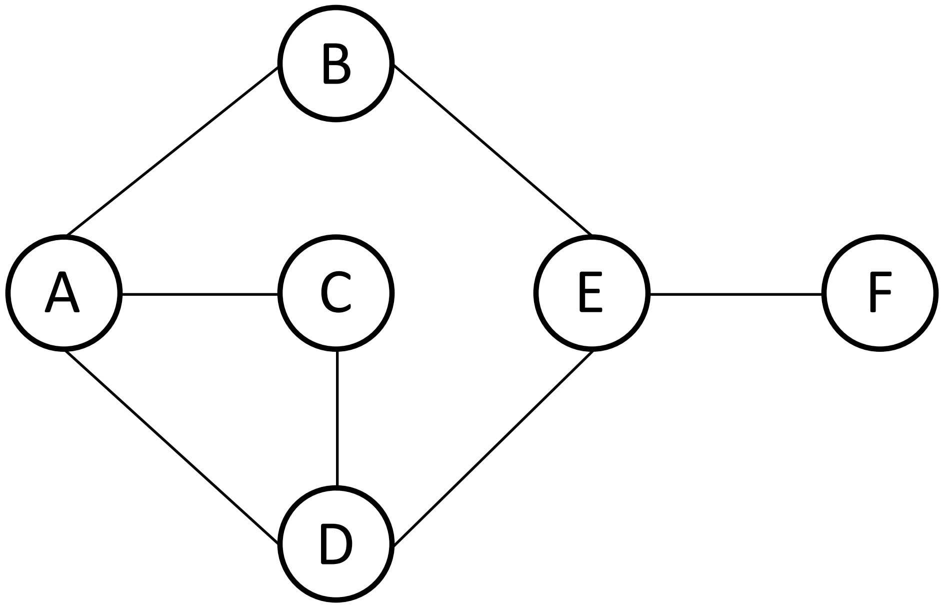
	Cost	Nexthop
A	1	A
B	**2**	A
C	0	**--**
D	1	**D**
E	**2**	D

D: Routing Table

	Cost	Nexthop
A	1	A
B	**2**	A, E
C	1	**C**
D	**0**	**--**
E	**1**	E

E: Routing Table

	Cost	Nexthop
A	**2**	B, D
B	1	B
C	**2**	**D**
D	1	**D**
E	**0**	--



1st Update

F: Routing Table

	Cost	Nexthop
A	∞	?
B	∞	?
C	∞	**?**
D	∞	**?**
E	**1**	E
F	**0**	--

+

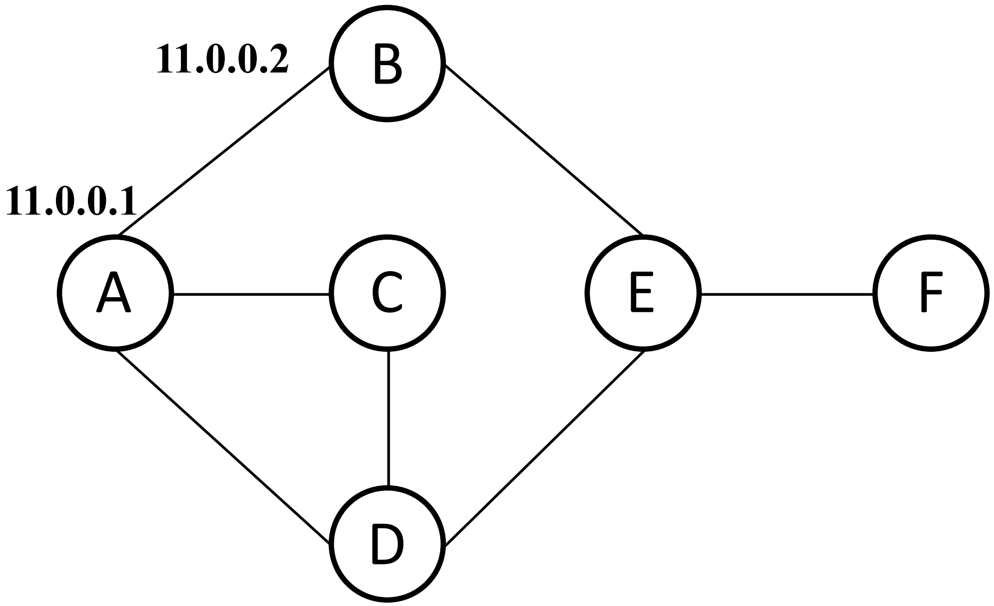
E → F

	Cost	Nexthop
A	∞	?
B	1	B
C	∞	**?**
D	1	**D**
E	**0**	--
F	**1**	F

→

F: Routing Table

	Cost	Nexthop
A	∞	?
B	**2**	E
C	∞	**?**
D	**2**	**E**
E	**1**	E
F	**0**	--



2nd Update

F: Routing Table

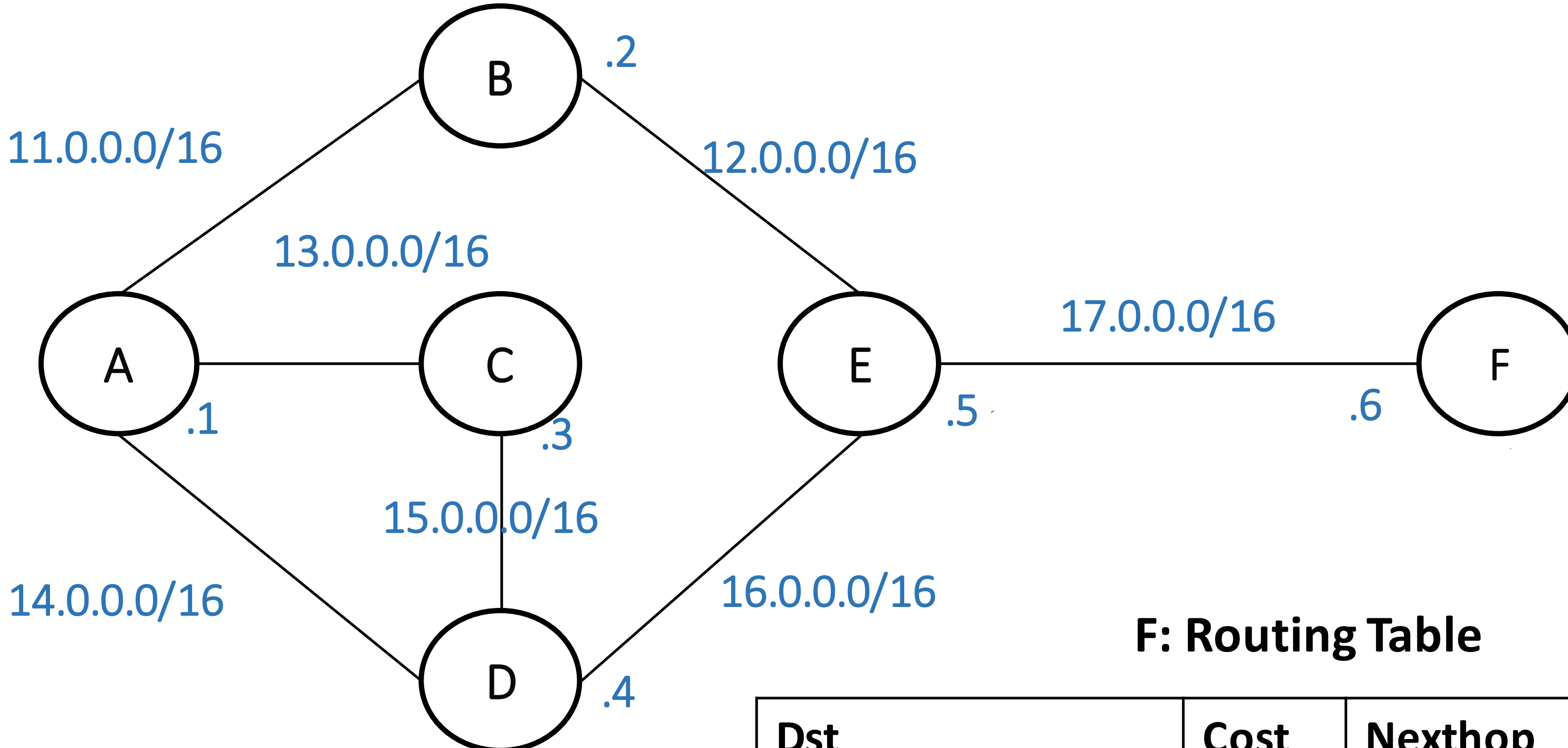
	Cost	Nexthop
A	**∞**	?
B	**2**	E
C	**∞**	**?**
D	**2**	**E**
E	**1**	E
F	**0**	--

E → F

	Cost	Nexthop
A	**2**	B, D
B	1	B
C	**2**	**D**
D	1	**D**
E	**0**	--
F	**1**	F

F: Routing Table

	Cost	Nexthop
A	**3**	E
B	**2**	E
C	**3**	**E**
D	**2**	**E**
E	**1**	E
F	**0**	--



F: Routing Table

Dst	Cost	Nexthop
11.0.0.0/16	**3**	17.0.0.5
12.0.0.0/16	**2**	17.0.0.5
13.0.0.0/16	**3**	17.0.0.5
14.0.0.0/16	**3**	17.0.0.5
15.0.0.0/16	**3**	17.0.0.5
16.0.0.0/16	**2**	17.0.0.5
17.0.0.0/16	**1**	connected

Characteristics of Distance Vector Routing

- **Periodic Updates:** Updates to the routing tables are sent at the end of a certain time period. A typical value is 90 seconds.
- **Triggered Updates:** If a metric changes on a link, a router immediately sends out an update without waiting for the end of the update period.
- **Full Routing Table Update**: Most distance vector routing protocol send their neighbors the entire routing table (not only entries which change).
- **Route invalidation timers:** Routing table entries are invalid if they are not refreshed. A typical value is to invalidate an entry if no update is received after 3-6 update periods.

The Count-to-Infinity Problem

A's Routing Table		
to	via (next hop)	cost
C	B	2

B's Routing Table		
to	via (next hop)	cost
C	C	1

now link B-C goes down

takes precedence

A: to	A: via	A: cost	A → B	B → A	B: to	B: via	B: cost
C	B	2			C	-	oo
			C 2 →	← C oo			
C	-	oo			C	A	3
			C oo →	← C 3			
C	B	4			C	-	oo
			C 4 →	← C oo			

Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a “next-hop-view”
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B

- How can the Count-to-Infinity problem be solved?

Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a "next-hop-view"
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B
- How can the Count-to-Infinity problem be solved?
- **Solution:** Never advertise the cost to a neighbor if this neighbor is the next hop on the current path **(Split Horizon)**
  - Example: A would not send the first routing update to B, since B is the next hop on A's current route to C
  - Split Horizon does not solve count-to-infinity in all cases!

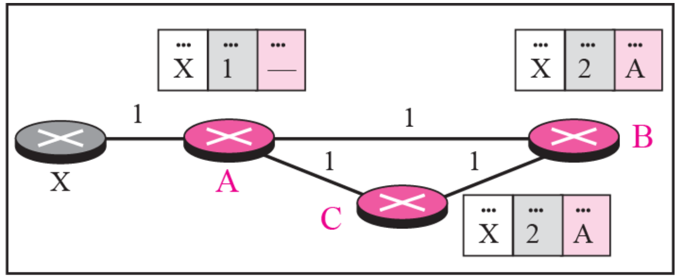
Split Horizon and Poison Reverse

- One drawback of Split Horizon
  - Normally, the DV protocol uses a timer and if there is no news about a route, the node deletes the route from its table
  - In the previous e.g., node B cannot guess that this is due to split horizon or because A has not received any news about C recently
- Poison Reverse
  - Node A can still advertise the value for C, but is the source of information is B, it can replace the distance with infinity as a warning

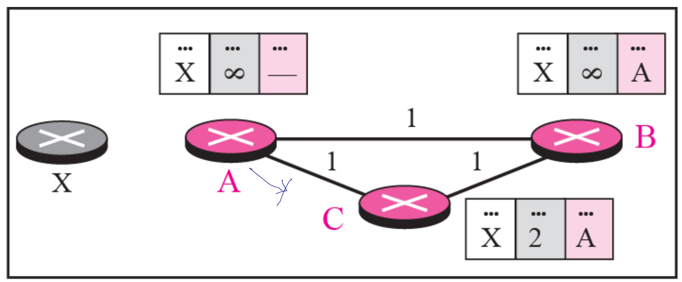
Three-node instability

If the instability is btw three nodes, stability cannot be guaranteed

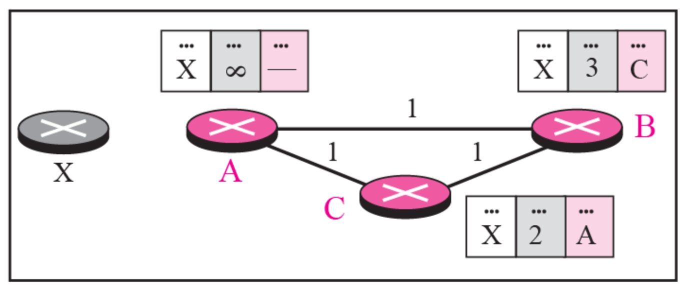
Before failure



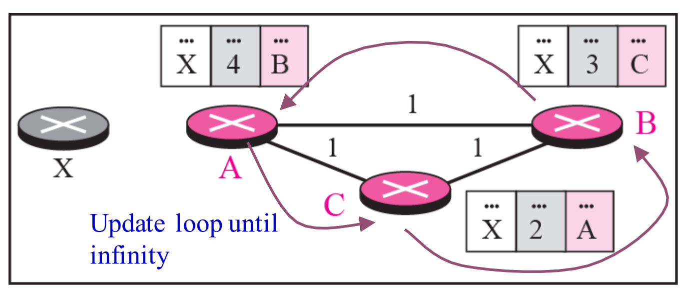
After A sends the route to B and C, but the packet to C is lost



After C sends the route to B



After B sends the route to A



RIP - Routing Information Protocol

- A simple intra-domain protocol
- Straightforward implementation of Distance Vector Routing
- Each router advertises its distance vector every 30 seconds (or whenever its routing table changes) to all of its neighbors
- RIP always uses 1 as link metric
- Maximum hop count is 15, with "16" equal to "∞"
- Routes are timeout (set to 16) after 3 minutes if they are not updated

RIP - History

• Late 1960s :	Distance Vector protocols were used in the ARPANET
• Mid-1970s:	XNS (Xerox Network system) routing protocol is the precursor of RIP in IP (and Novell's IPX RIP and Apple's routing protocol)
• 1982	Release of **routed** for BSD Unix
• 1988	RIPv1 (RFC 1058) - classful routing
• 1993	RIPv2 (RFC 1388) - adds subnet masks with each route entry - allows classless routing
• 1998	Current version of RIPv2 (RFC 2453)

RIP Packet Format

MAC Hdr	IP Hdr	UDP Hdr	RIP Hdr	RIP type-specific Data

IPv4

- MAC Header
  - Src – MAC of sending interface
  - Dst – 0100:5E00:0009 ?
  - Type – 0x0800 frame has an IPv4 packet
- IP Header
  - Src – IP addr of sending interface
  - Dst – 224.0.0.9 multicast
  - Protocol – UDP 17
- UDP Header
  - Port 520

IPv6

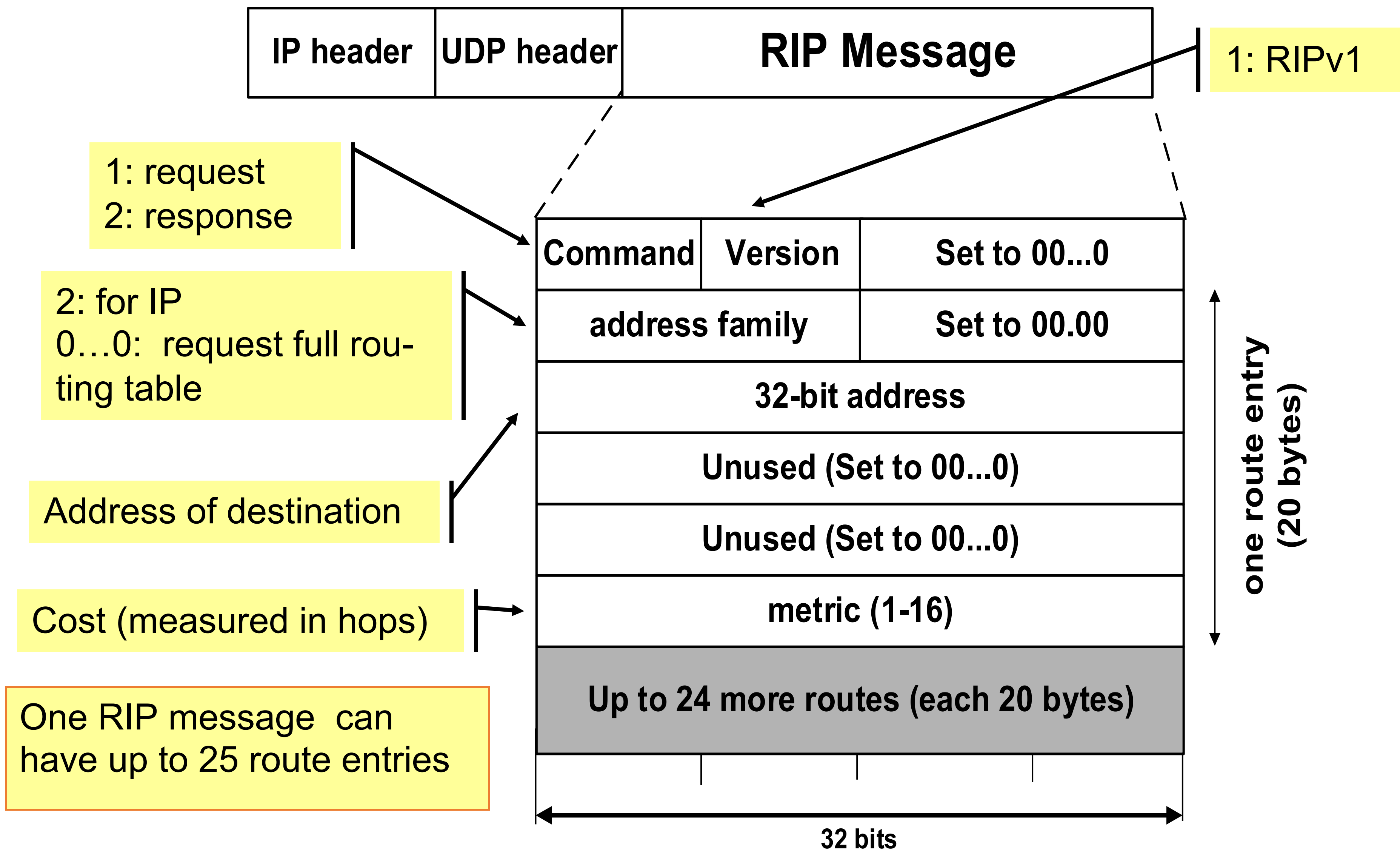
- MAC Header
  - Dst – 3333:0000:0009
  - Type – 0x86DD
- IPv6 Header
  - Src – IPv6 addr of sending interface
  - Dst – FF02::9
  - Protocol – UDP 17
- UDP Header
  - Port 521

RIP Messages

- This is the operation of RIP in **`routed`**. Dedicated port for RIP is UDP port 520.

- Two types of messages:
  - **Request messages**
    - used to ask neighboring nodes for an update
  - **Response messages**
    - contains an update

RIPv1 Packet Format



RIPv2

- RIPv2 is an extends RIPv1:
  - Subnet masks are carried in the route information
  - Authentication of routing messages
  - Route information carries next-hop address
  - Exploites IP multicasting

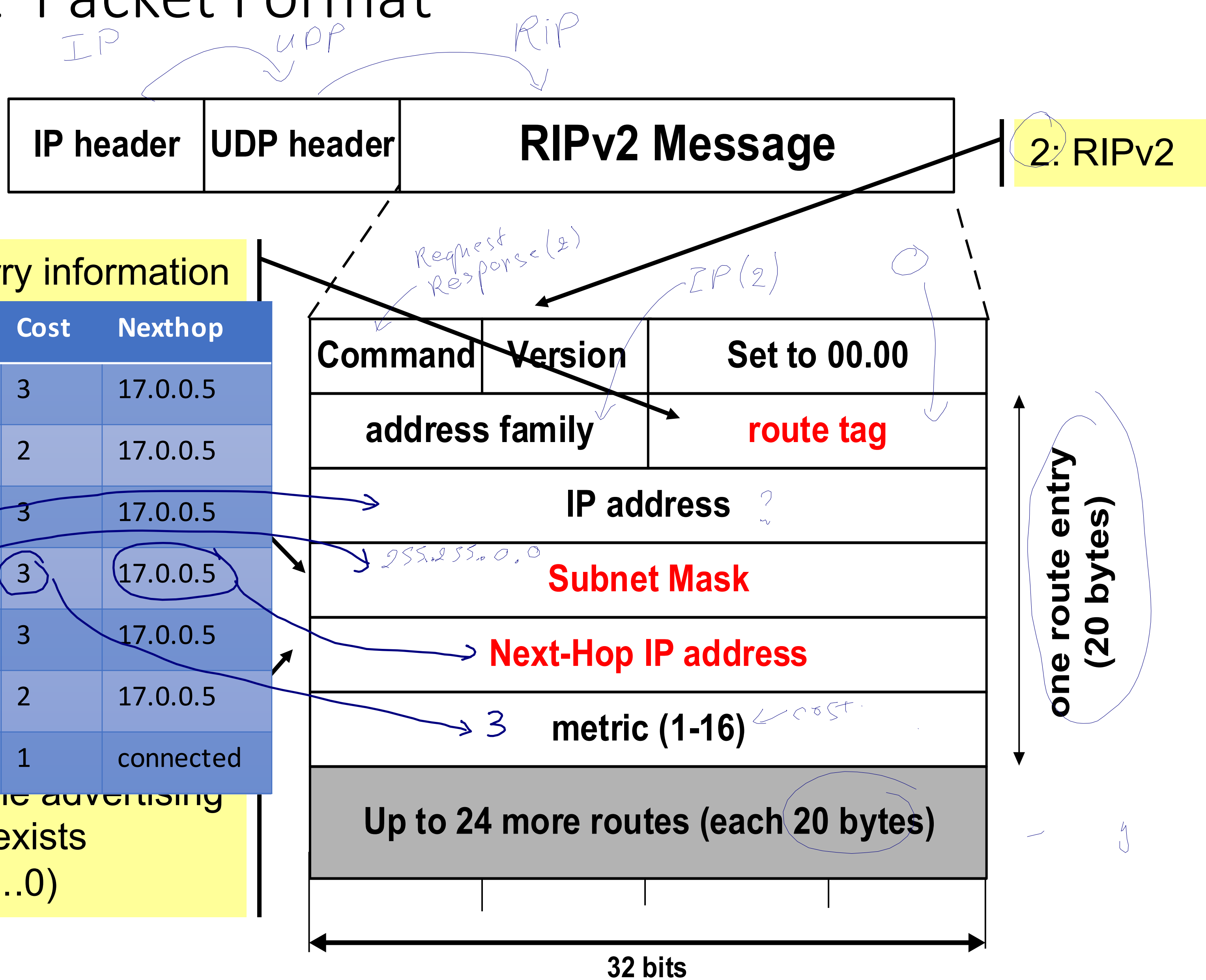
- Extensions of RIPv2 are carried in unused fields of RIPv1 messages

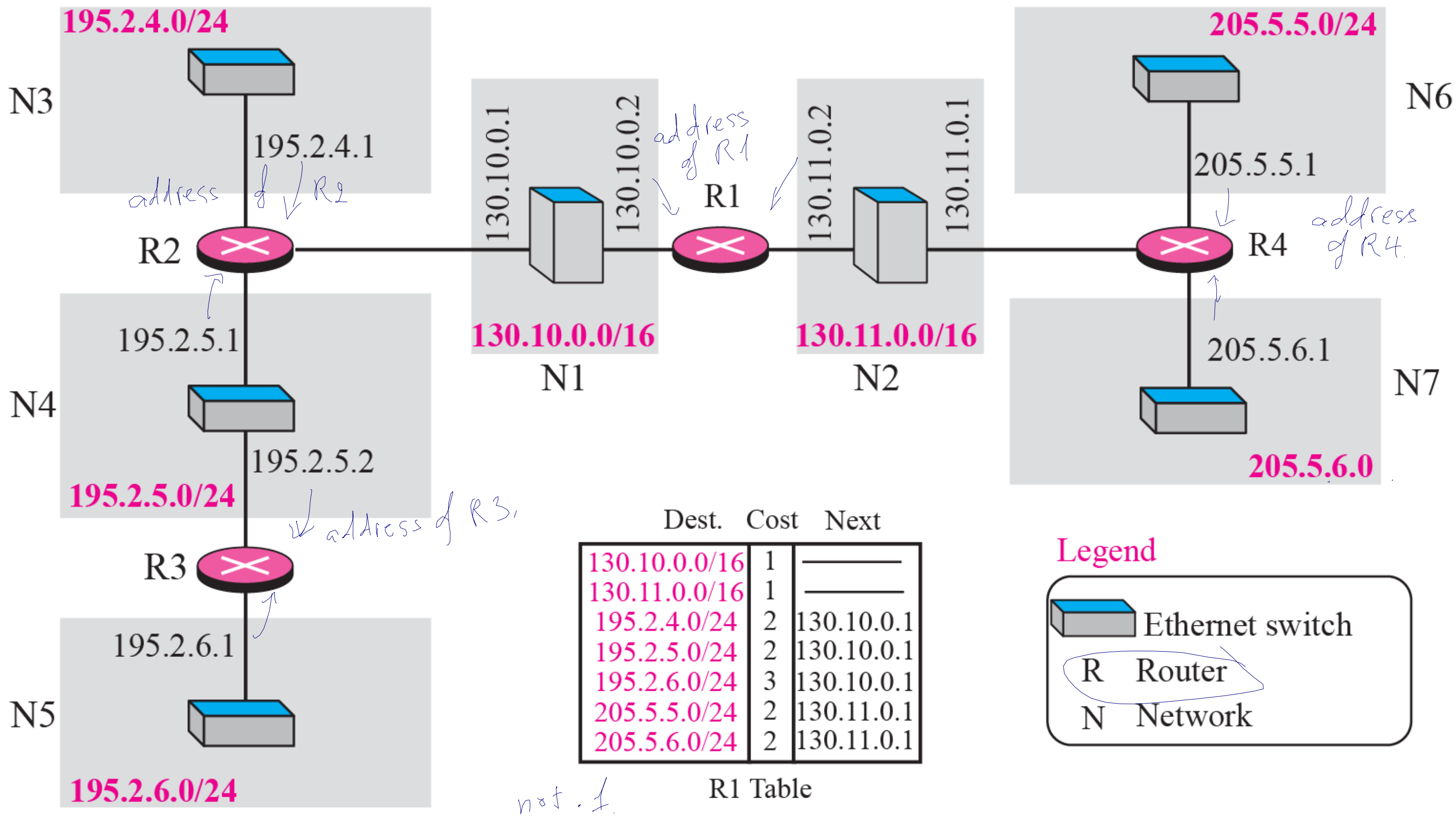
RIPv2 Packet Format

Used to carry information

Dst	Cost	Nexthop
11.0.0.0/16	3	17.0.0.5
12.0.0.0/16	2	17.0.0.5
13.0.0.0/16	3	17.0.0.5
14.0.0.0/16	3	17.0.0.5
15.0.0.0/16	3	17.0.0.5
16.0.0.0/16	2	17.0.0.5
17.0.0.0/16	1	connected

Io… a… subnet than the advertising router, if one exists (otherwise 0….0)





Dest.	Cost	Next
130.10.0.0/16	1	——
130.11.0.0/16	1	——
195.2.4.0/24	2	130.10.0.1
195.2.5.0/24	2	130.10.0.1
195.2.6.0/24	3	130.10.0.1
205.5.5.0/24	2	130.11.0.1
205.5.6.0/24	2	130.11.0.1

R1 Table

not .1

Dest.	Cost	Next
130.10.0.0/16	1	——
130.11.0.0/16	2	130.10.0.2
195.2.4.0/24	1	——
195.2.5.0/24	1	——
195.2.6.0/24	2	195.2.5.2
205.5.5.0/24	3	130.10.0.2
205.5.6.0/24	3	130.10.0.2

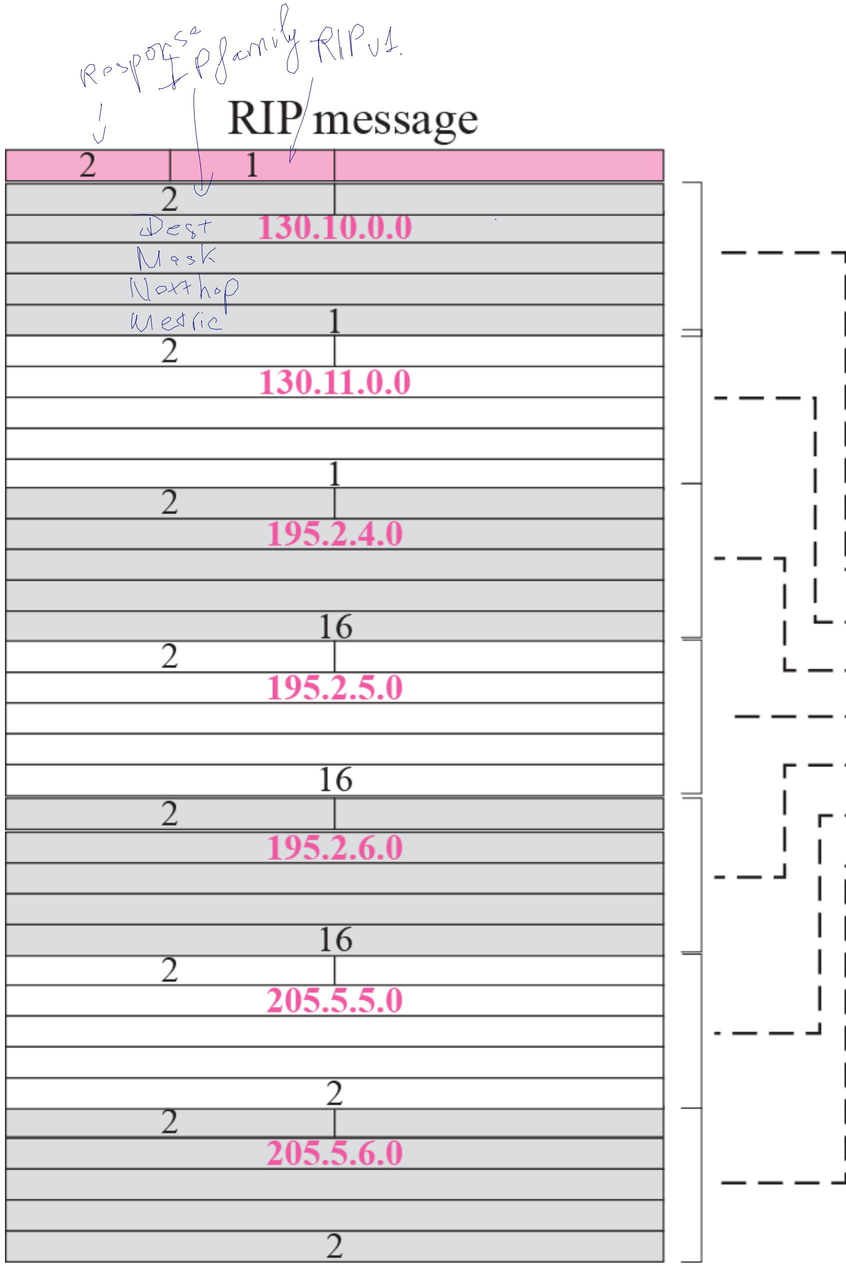
R2 Table

Dest.	Cost	Next
130.10.0.0/16	2	195.2.5.1
130.11.0.0/16	3	195.2.5.1
195.2.4.0/24	2	195.2.5.1
195.2.5.0/24	1	——
195.2.6.0/24	1	——
205.5.5.0/24	4	195.2.5.1
205.5.6.0/24	4	195.2.5.1

R3 Table

Dest.	Cost	Next
130.10.0.0/16	2	130.11.0.2
130.11.0.0/16	1	——
195.2.4.0/24	3	130.11.0.2
195.2.5.0/24	3	130.11.0.2
195.2.6.0/24	4	130.11.0.2
205.5.5.0/24	1	——
205.5.6.0/24	1	——

R4 Table



Dest.	Cost
130.10.0.0	1
130.11.0.0	1
195.2.4.0	16
195.2.5.0	16
195.2.6.0	16
205.5.5.0	2
205.5.6.0	2

Information extracted
from message

Routing with RIP

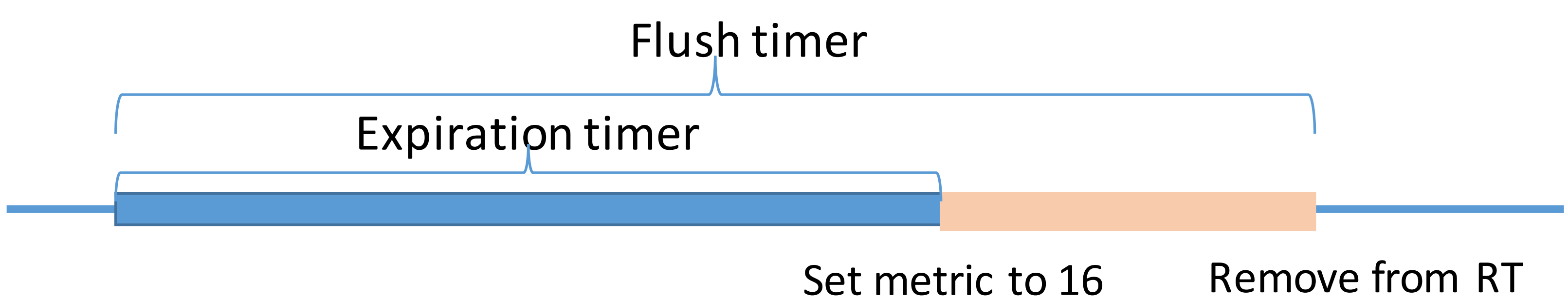
- **Initialization:** Send a **request packet** (command = 1, address family=0..0) on all interfaces:
    - RIPv1 uses broadcast (255.255.255.255) if possible,
    - RIPv2 uses multicast address 224.0.0.9, if possible

  requesting routing tables from neighboring routers
- **Request received**: Routers that receive above request send their entire routing table
- **Response received**: Update the routing table

- **Regular routing updates**: Every 30 seconds, send all or part of the routing tables to every neighbor in an response message
- **Triggered Updates:** Whenever the metric for a route change, send entire routing table.

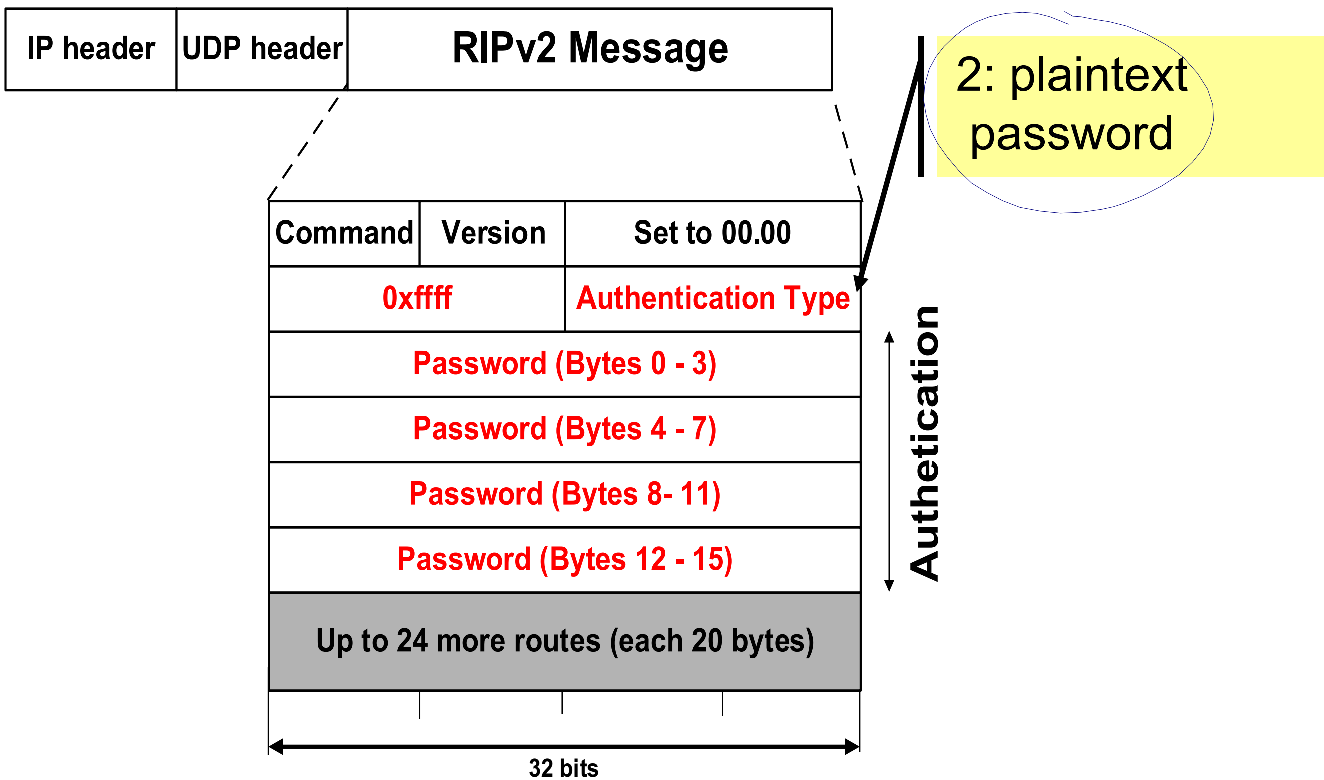
RIP Timers

- **Periodic Update:** 30 seconds with random jitter to avoid table synchronization.
- **Expiration Timer** (Timeout, invalidation timer): 180 seconds
  - Reset to initial value when an update received
- **Garbage collection/Flush Timer**: 240 seconds. Routes will be removed from routing table when the timer expired.
- **Holddown Timer**: 180 seconds. Set when an update with a hop count higher than the metric recorded in the route table.



RIP Security

- Issue: Sending bogus routing updates to a router
- RIPv1: No protection
- RIPv2: Simple authentication scheme

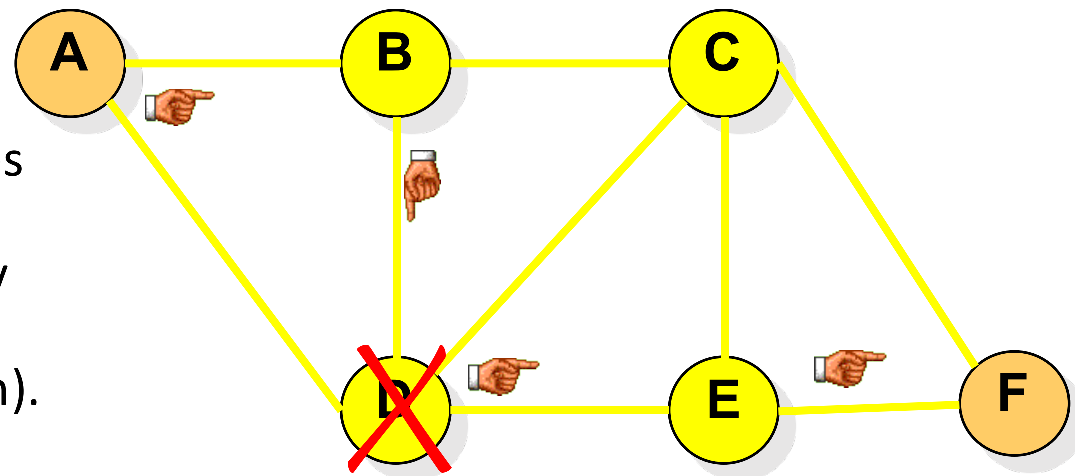


RIP Problems

- RIP takes a long time to stabilize
  - Even for a small network, it takes several minutes until the routing tables have settled after a change
- RIP has all the problems of distance vector algorithms, e.g., count-to-Infinity
  - RIP uses split horizon to avoid count-to-infinity
- The maximum path in RIP is 15 hops

Distance Vector vs. Link State Routing

- With distance vector routing, each node has information only about the next hop:
  - Node A: to reach F go to B
  - Node B: to reach F go to D
  - Node D: to reach F go to E
  - Node E: go directly to F



- Distance vector routing makes poor routing decisions if directions are not completely correct
  (e.g., because a node is down).

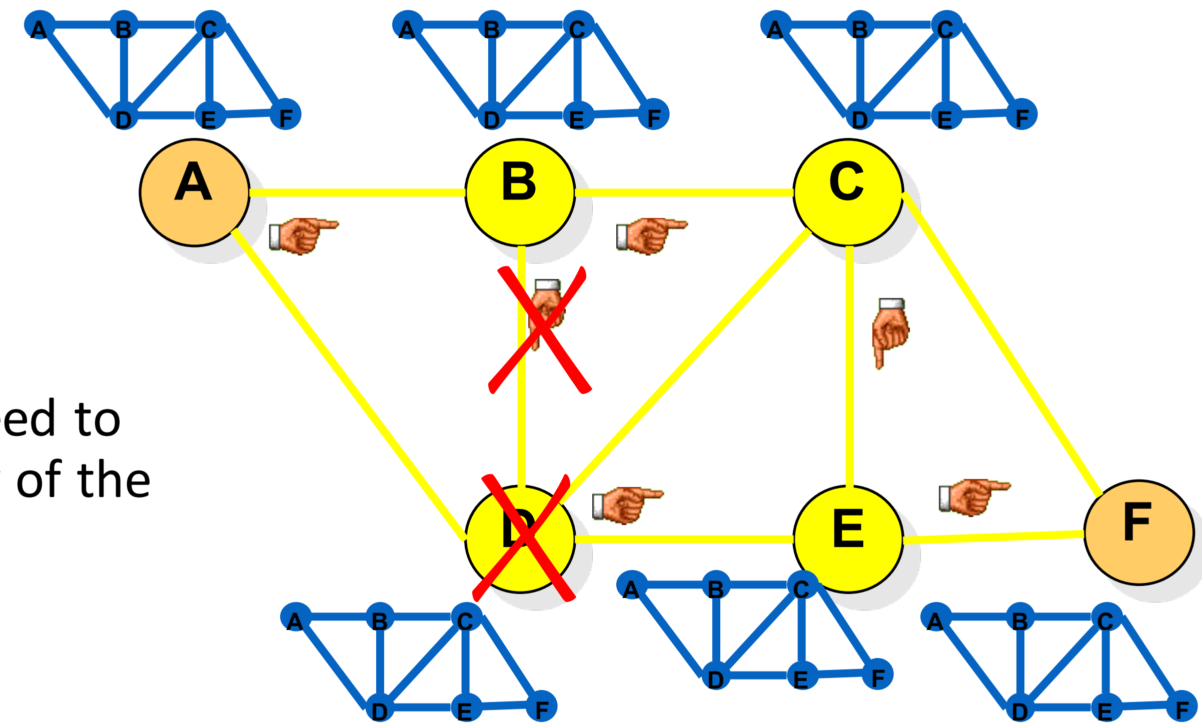
- If parts of the directions incorrect, the routing may be incorrect until the routing algorithms has re-converged.

Distance Vector vs. Link State Routing

- In link state routing, each node has a complete map of the topology

- If a node fails, each node can calculate the new route

- Difficulty: All nodes need to have a consistent view of the network



Link State Routing: Properties

- Each node requires complete topology information
- Link state information must be flooded to all nodes
- Guaranteed to converge

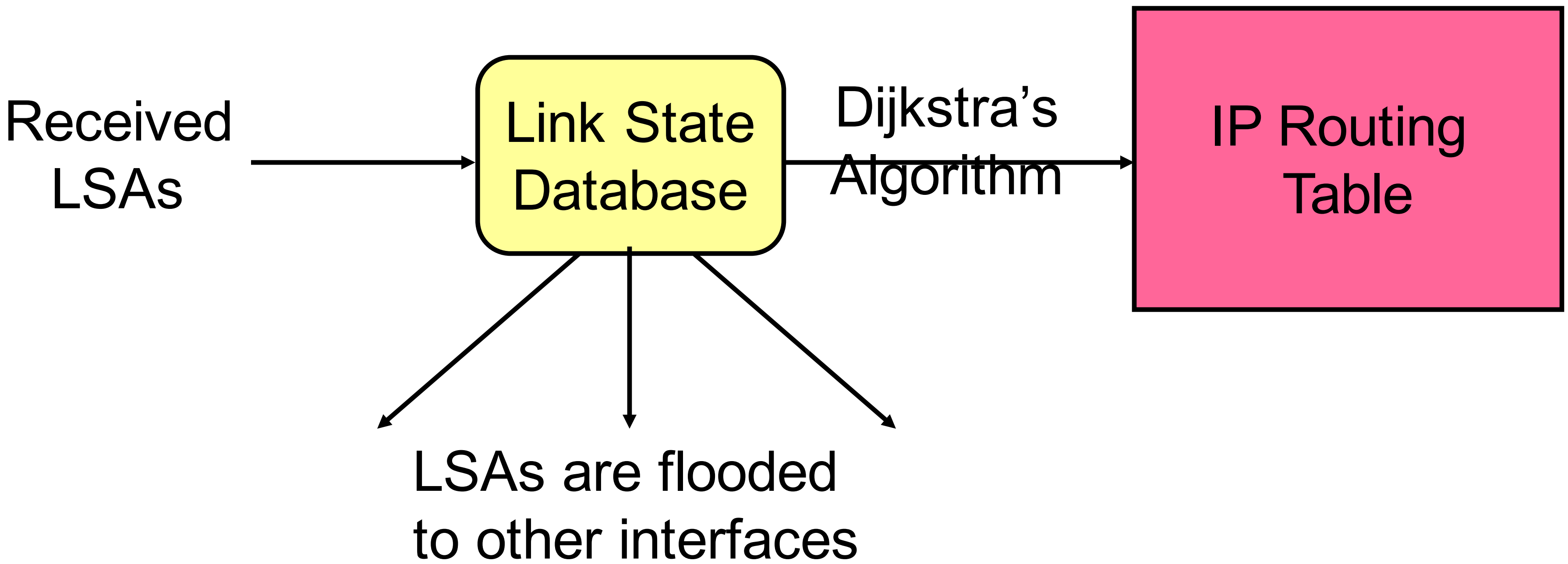
Link State Routing: Basic princples

1. Each router establishes a relationship *("adjacency")* with its neighbors
2. Each router generates *link state advertisements (LSAs)* which are distributed to all routers

   LSA = (link id, state of the link, cost, neighbors of the link)

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

Operation of a Link State Routing protocol



Dijkstra's Shortest Path Algorithm for a Graph

Input: Graph (N,E) with

N the set of nodes and $E \subseteq N \times N$ the set of edges

d_{vw} link cost (d_{vw} = infinity if $(v,w) \notin E$, $d_{vv} = 0$)

s source node.

Output: D_n cost of the least-cost path from node s to node n

$M = \{s\};$

for each $n \notin M$

$D_n = d_{sn};$

while ($M \neq$ all nodes) do

Find $w \notin M$ for which $D_w = \min\{D_j \ ; j \notin M\};$

Add w to M;

for each $n \notin M$

$D_n = \min_w [\ D_n, D_w + d_{wn}\ ];$

Update route;

enddo

Introduction to OSPF Concepts

Introducing OSPF and Link State Concepts

- **Advantages of OSPF**
- **Brief History**
- **Terminology**
- **Link State Concepts**

Introducing the OSPF Routing Protocol

- **Metric based on Cost (Bandwidth)**
- **Hello Protocol**
- **Steps to OSPF Operation**
- **DR/BDR**
- **OSPF Network Types**

Brief History

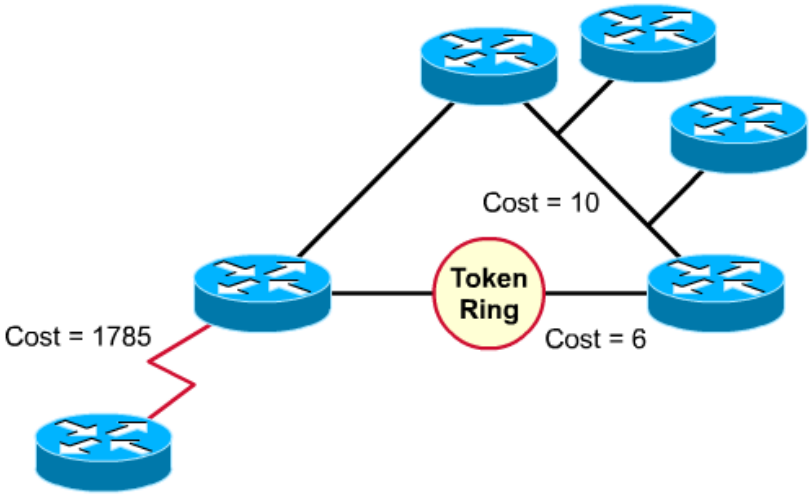
- 1987 – Initial development of OSPF began by IETF OSPF working group
- 1989 – OSPFv1 published in RFC 1131. OSPFv1 was experimental and was never deployed.
- 1991 – OSPFv2 introduced in RFC 1247 by John Moy. It is classless by design.
- 1998 – OSPFv2 specification updated in RFC 2328 which remains the current RFC for OSPF
- 1999 – OSPFv3 for IPv6 published in RFC 2740
- 2008 – OSPFv3 updated in RFC 5340

Features of OSPF

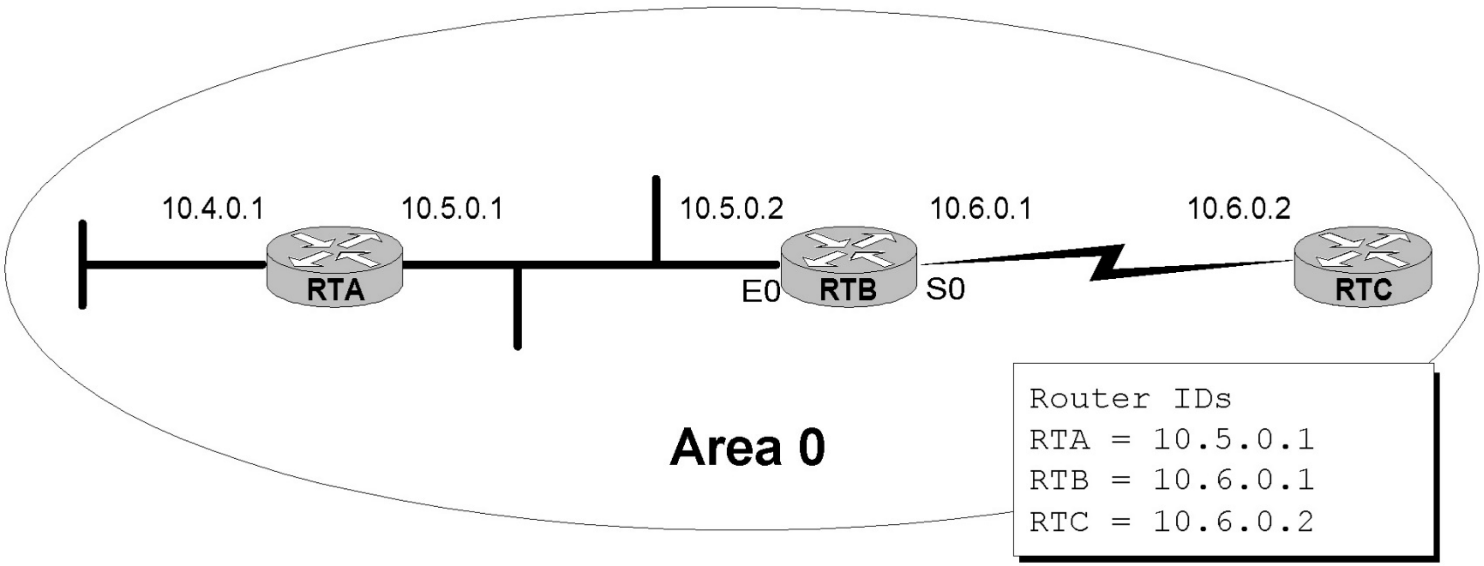
- Classless
- Efficient – no periodic updates. Use the SPF algorithm to choose the best path
- Fast convergence
- Scalable – Hierarchical
- Secure – Support MD5 authentication

Terminology

- **Link**: Interface on a router
- **Link state**: Description of an interface and of its relationship to its neighboring routers, including:
  - IP address/mask of the interface,
  - The type of network it is connected to
  - The routers connected to that network
  - The metric (cost) of that link
- The collection of all the link-states would form a **link-state database**.



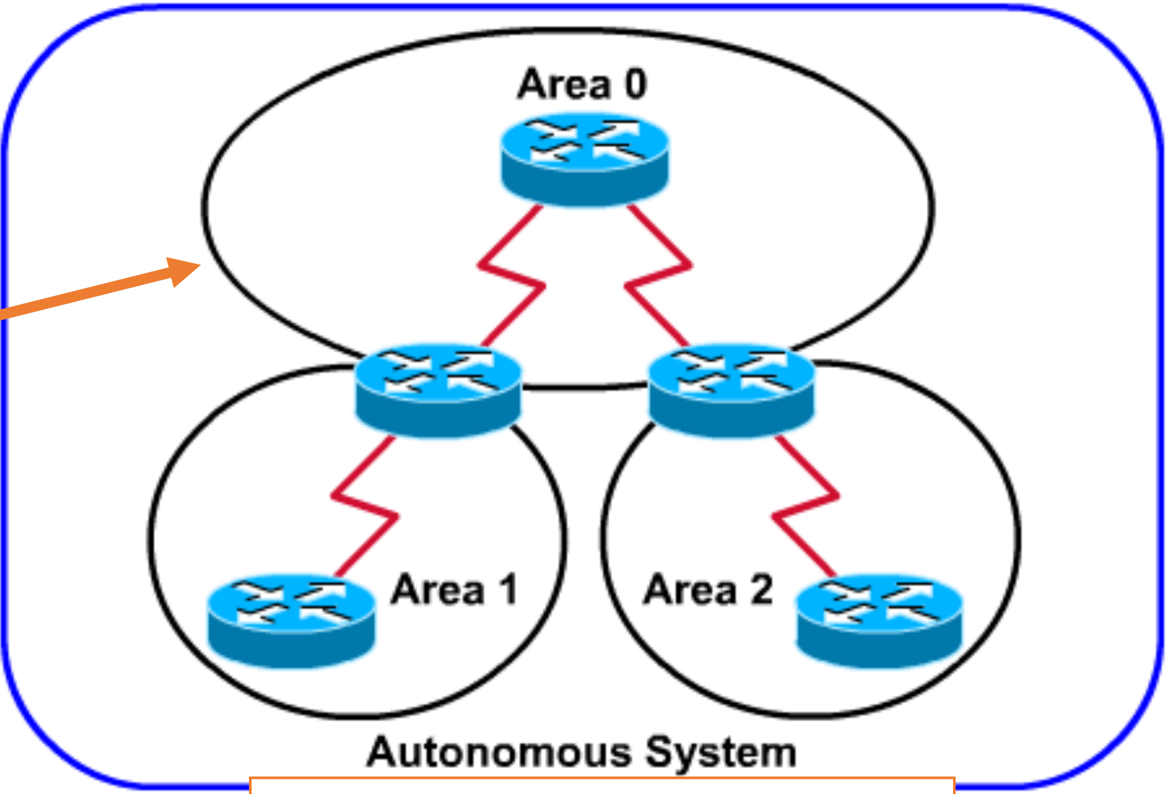
Terminology



- **Router ID** – Used to identify the routers in the OSPF network
  - IP address configured with the OSPF `router-id` command (extra)
  - **Highest loopback address** (configuration coming)
  - **Highest active IP address** (any IP address)
- Loopback address has the advantage of never going down, thus diminishing the possibility of having to re-establish adjacencies. (more in a moment)

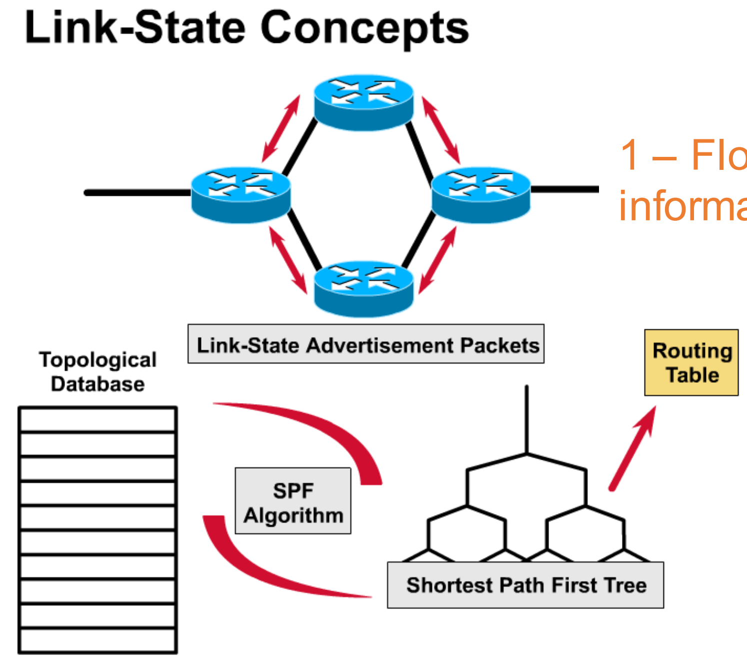
Terminology

Single Area OSPF uses only one area, usually Area 0



Or "OSPF Routing Domain"

Link State

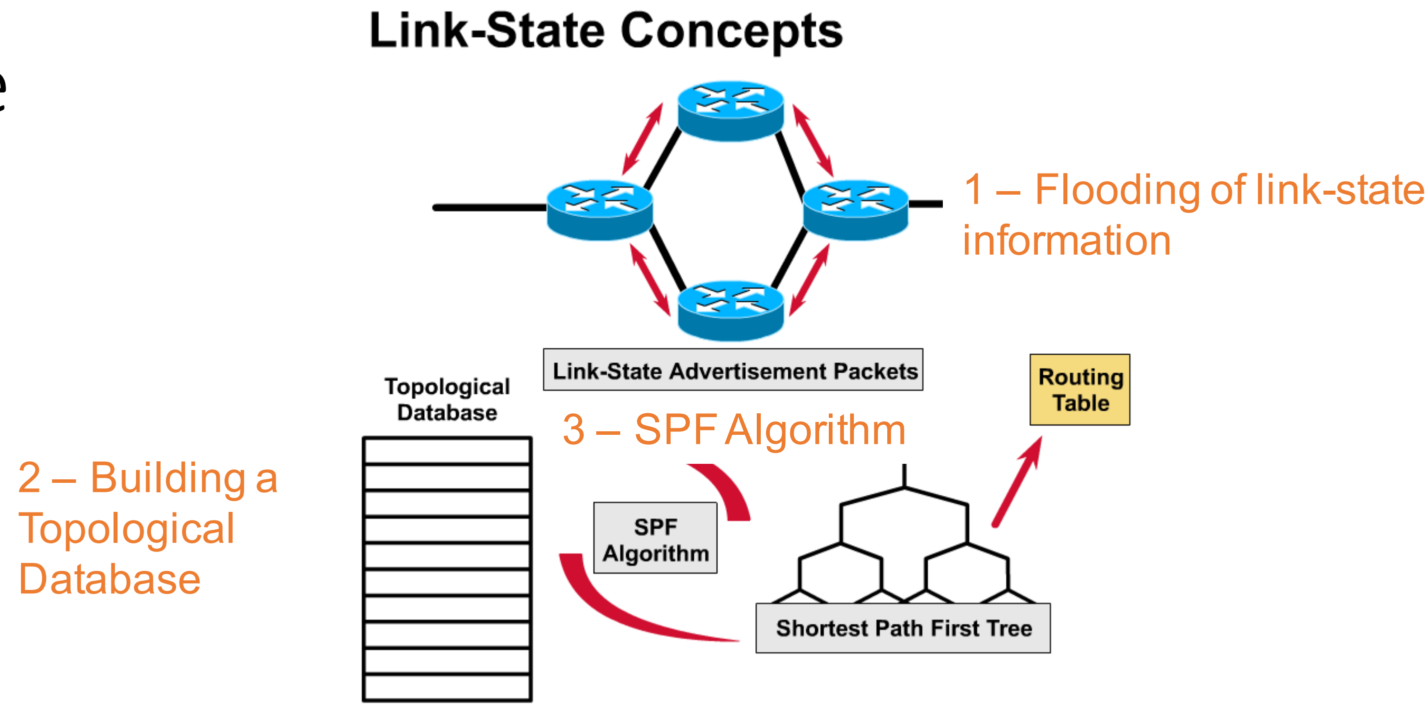


1 – Flooding of link-state information

1 – Flooding of link-state information

- The first thing that happens is that each node, router, on the network announces its own piece of link-state information to other all other routers on the network. This includes who their neighboring routers are and the cost of the link between them.
- Example: “Hi, I’m RouterA, and I can reach RouterB via a T1 link and I can reach RouterC via an Ethernet link.”
- Each router sends these announcements to all of the routers in the network.

Link State



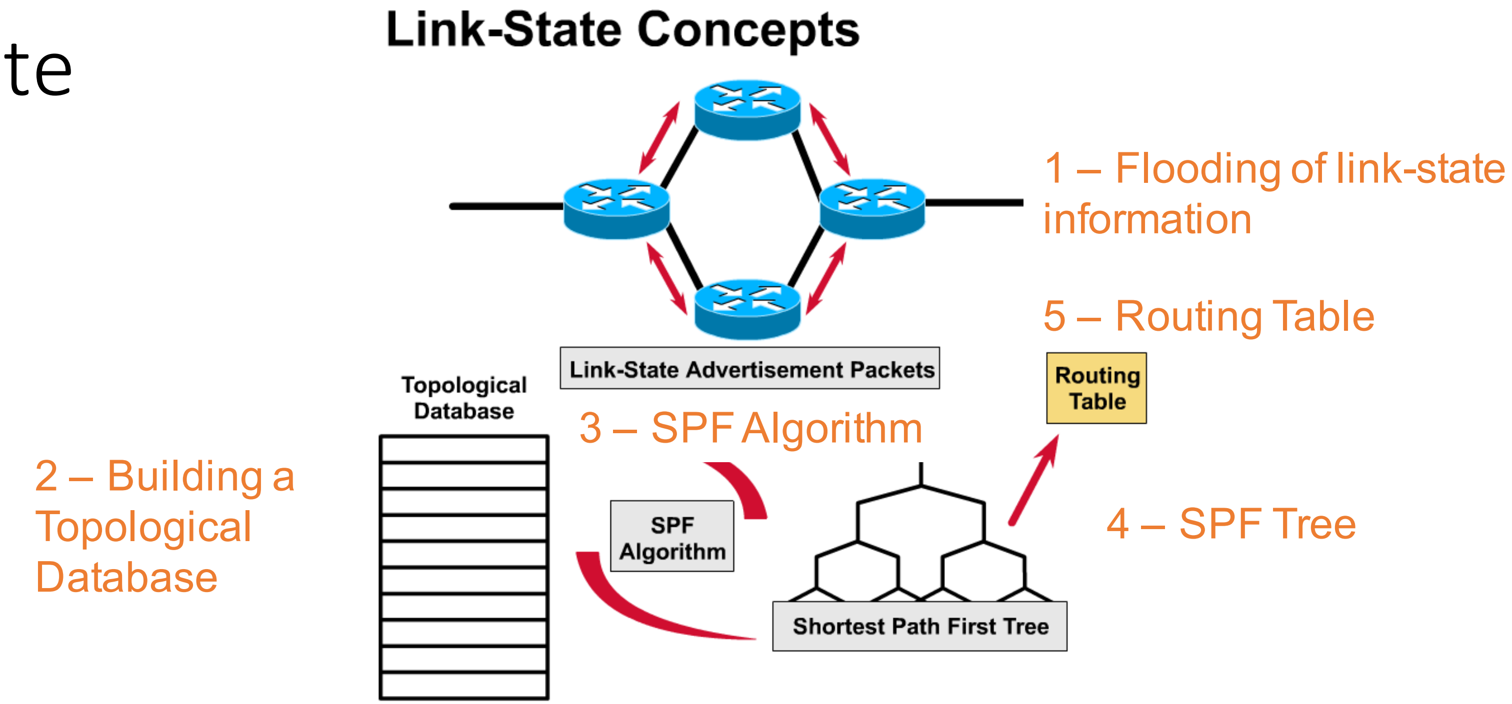
2. Building a Topological Database

- Each router collects all of this link-state information from other routers and puts it into a topological database.

3. Shortest-Path First (SPF), Dijkstra's Algorithm

- Using this information, the routers can recreate a topology graph of the network.
- Believe it or not, this is actually a very simple algorithm and I highly suggest you look at it some time, or even better, take a class on algorithms. (Radia Perlman's book, *Interconnections*, has a very nice example of how to build this graph – she is one of the contributors to the SPF and Spanning-Tree algorithms.)

Link State



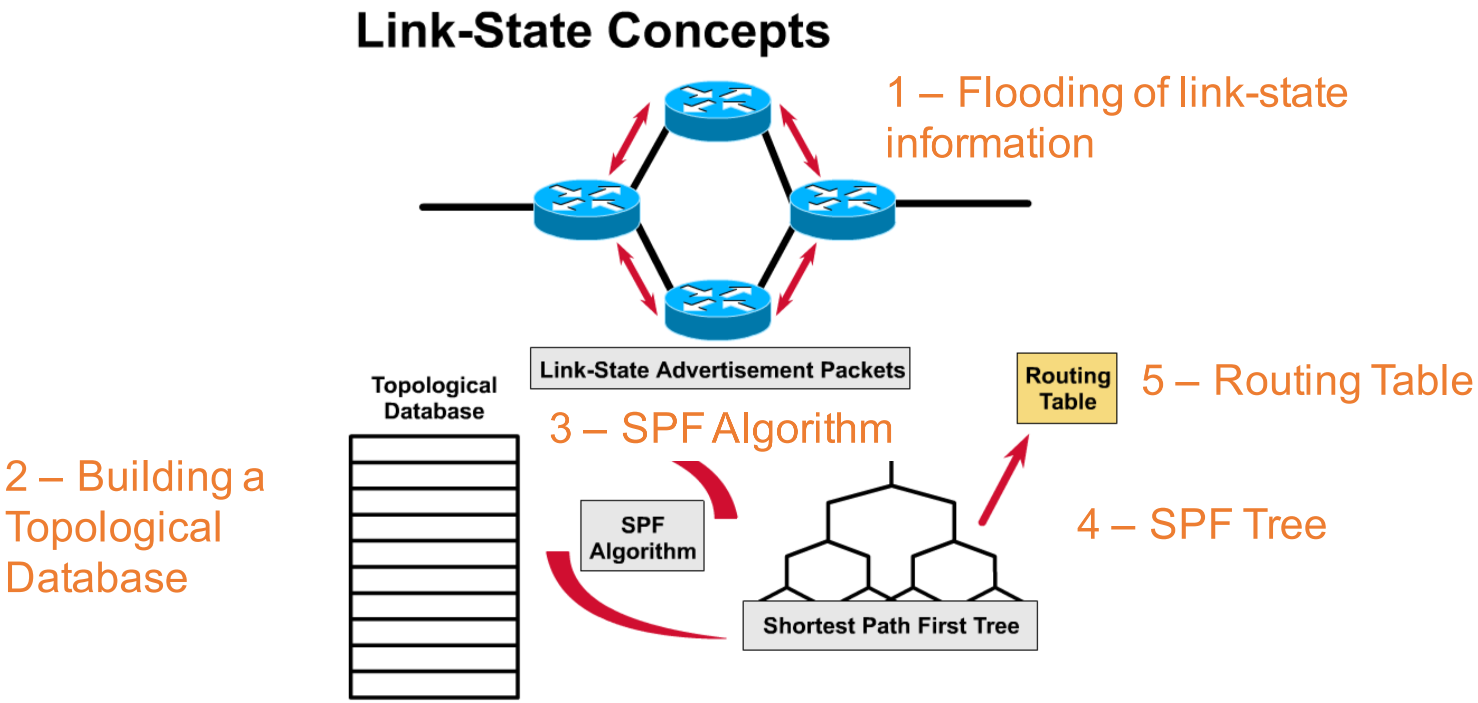
4. Shortest Path First Tree

- This algorithm creates an SPF tree, with the router making itself the root of the tree and the other routers and links to those routers, the various branches.

5. Routing Table

- Using this information, the router creates a routing table.

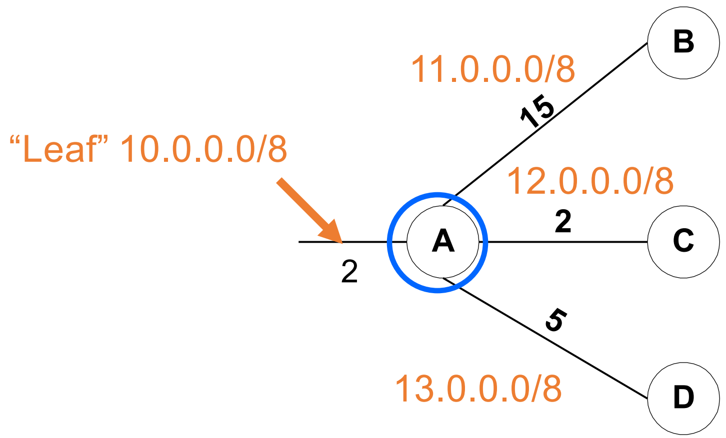
Link State Concepts



- How does the SPF algorithm create an SPF Tree?
- Let’s take a look!
- This is *extra* Information.

Extra: Simplified Link State Example

- *In order to keep it simple, we will take some liberties with the actual process and algorithm, but you will get the basic idea!*
- You are **RouterA** and you have exchanged "Hellos" with:
  - RouterB on your network 11.0.0.0/8 with a cost of 15,
  - RouterC on your network 12.0.0.0/8 with a cost of 2
  - RouterD on your network 13.0.0.0/8 with a cost of 5
  - Have a "leaf" network 10.0.0.0/8 with a cost of 2
- This is your link-state information, which you will flood to all other routers.
- All other routers will also flood their link state information. (OSPF: only within the area)



Extra: Simplified Link State Example

RouterA's Topological Data Base (Link State Database)

All other routers flood their own link state information to all other routers.

RouterA gets all of this information and stores it in its LSD (Link State Database).

Using the link state information from each router, RouterC runs Dijkstra algorithm to create a SPT. (next)

RouterB:

- **Connected to RouterA on network 11.0.0.0/8, cost of 15**
- **Connected to RouterE on network 15.0.0.0/8, cost of 2**
- **Has a "leaf" network 14.0.0.0/8, cost of 15**

RouterC:

- **Connected to RouterA on network 12.0.0.0/8, cost of 2**
- **Connected to RouterD on network 16.0.0.0/8, cost of 2**
- **Has a "leaf" network 17.0.0.0/8, cost of 2**

RouterD:

- **Connected to RouterA on network 13.0.0.0/8, cost of 5**
- **Connected to RouterC on network 16.0.0.0/8, cost of 2**
- **Connected to RouterE on network 18.0.0.0/8, cost of 2**
- **Has a "leaf" network 19.0.0.0/8, cost of 2**

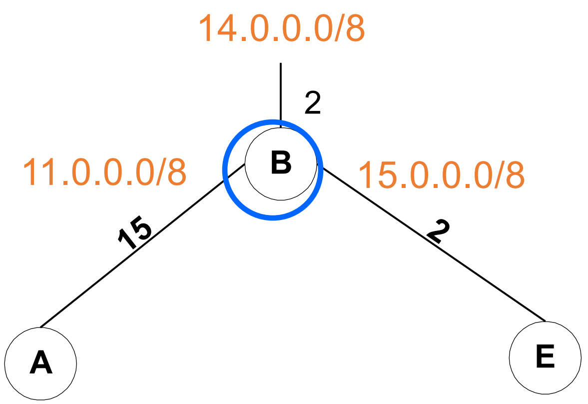
RouterE:

- **Connected to RouterB on network 15.0.0.0/8, cost of 2**
- **Connected to RouterD on network 18.0.0.0/8, cost of 10**
- **Has a "leaf" network 20.0.0.0/8, cost of 2**

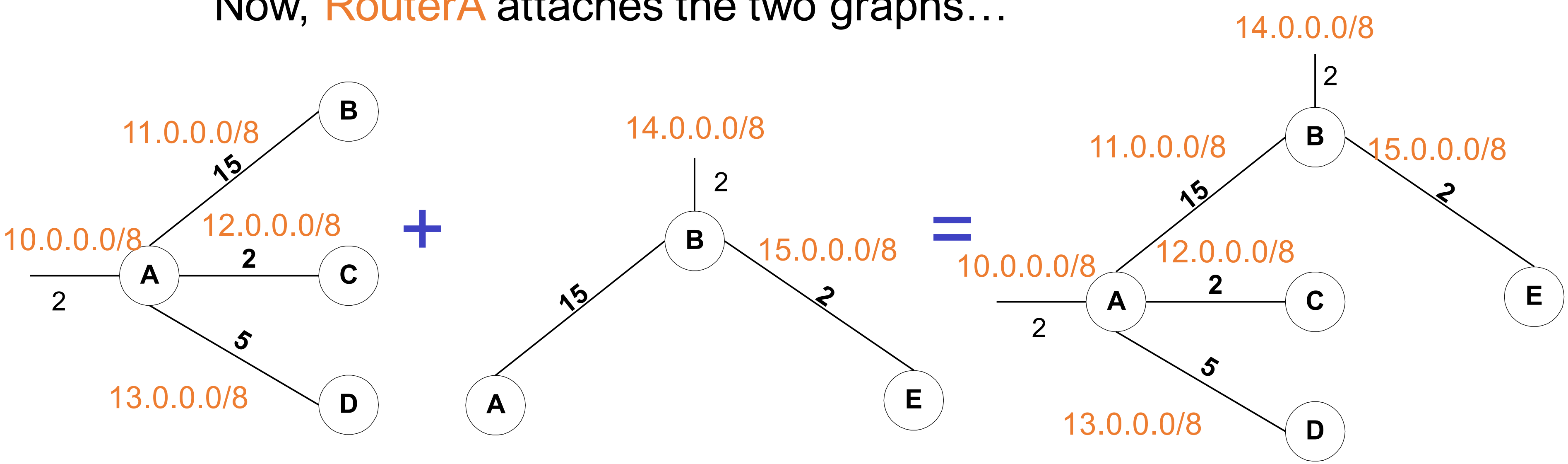
Link State information from RouterB

We now get the following link-state information from **RouterB**:

- Connected to RouterA on network 11.0.0.0/8, cost of 15
- Connected to RouterE on network 15.0.0.0/8, cost of 2
- Have a "leaf" network 14.0.0.0/8, cost of 15



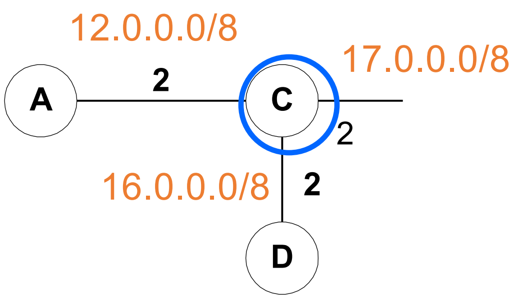
Now, RouterA attaches the two graphs…



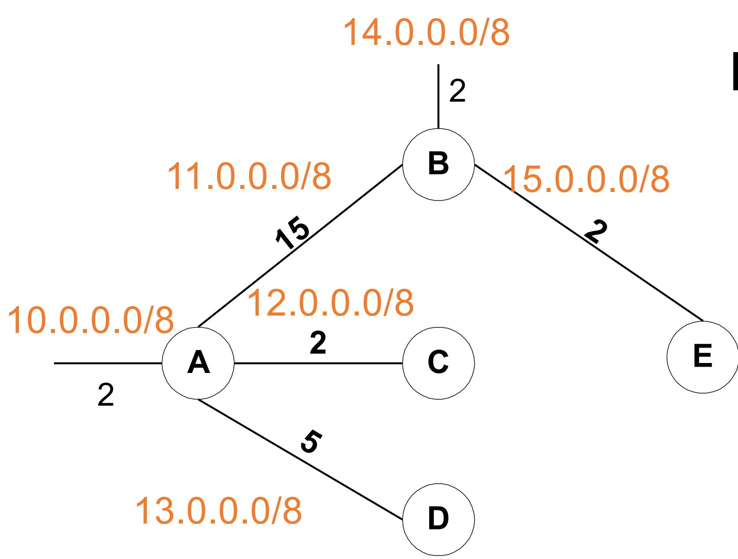
Link State information from RouterC

We now get the following link-state information from **RouterC**:

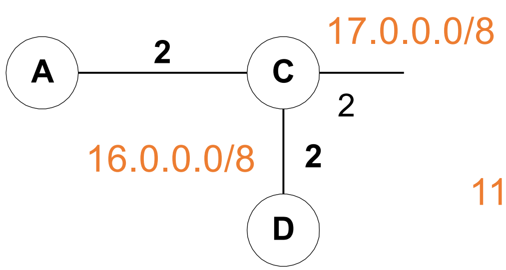
- Connected to RouterA on network 12.0.0.0/8, cost of 2
- Connected to RouterD on network 16.0.0.0/8, cost of 2
- Have a "leaf" network 17.0.0.0/8, cost of 2



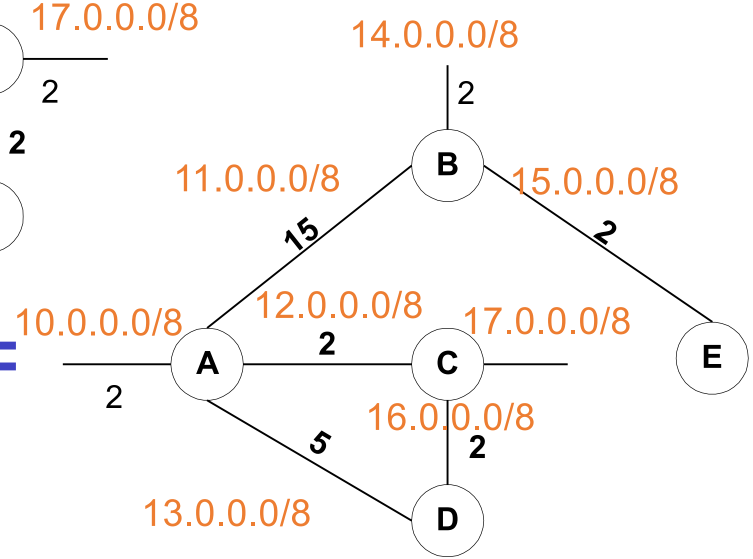
Now, RouterA attaches the two graphs…



+



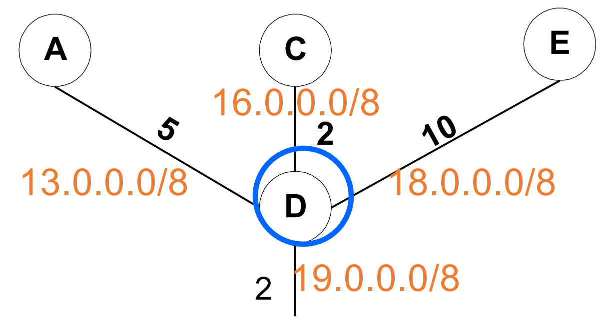
=



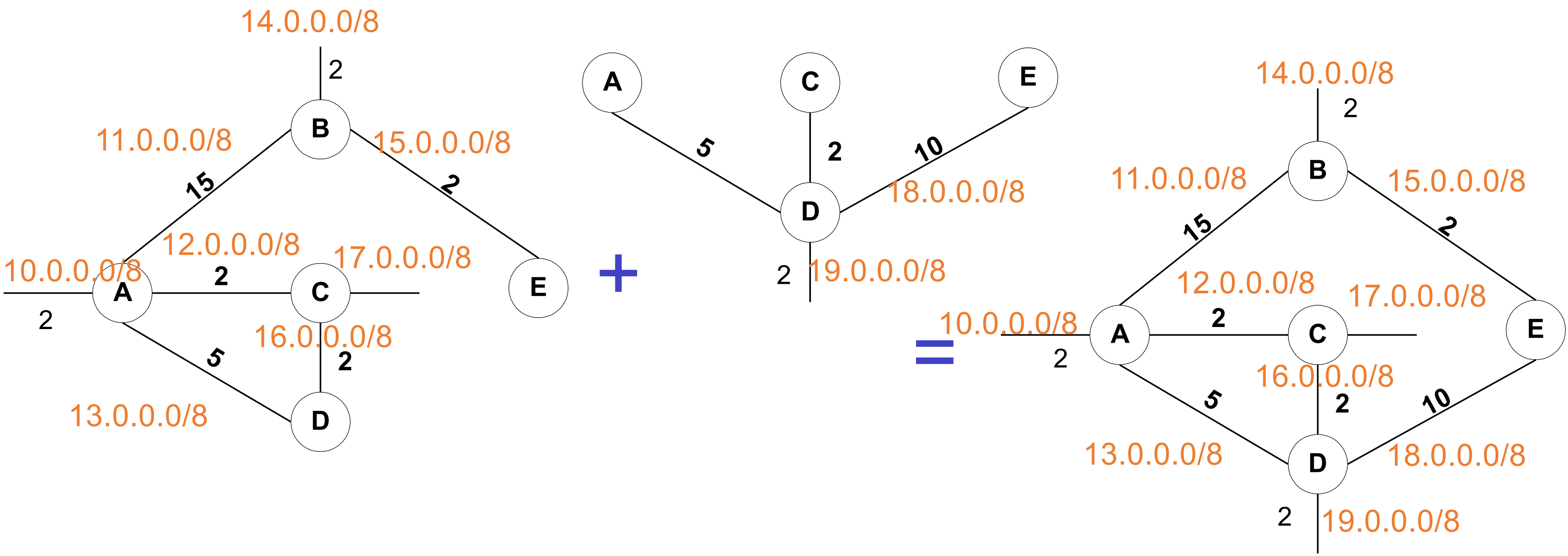
Link State information from RouterD

We now get the following link-state information from **RouterD**:

- Connected to RouterA on network 13.0.0.0/8, cost of 5
- Connected to RouterC on network 16.0.0.0/8, cost of 2
- Connected to RouterE on network 18.0.0.0/8, cost of 2
- Have a "leaf" network 19.0.0.0/8, cost of 2



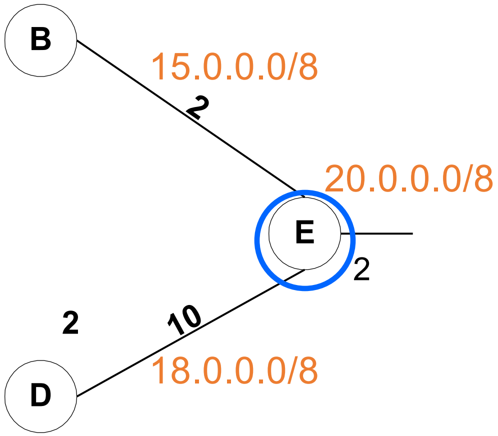
Now, RouterA attaches the two graphs…



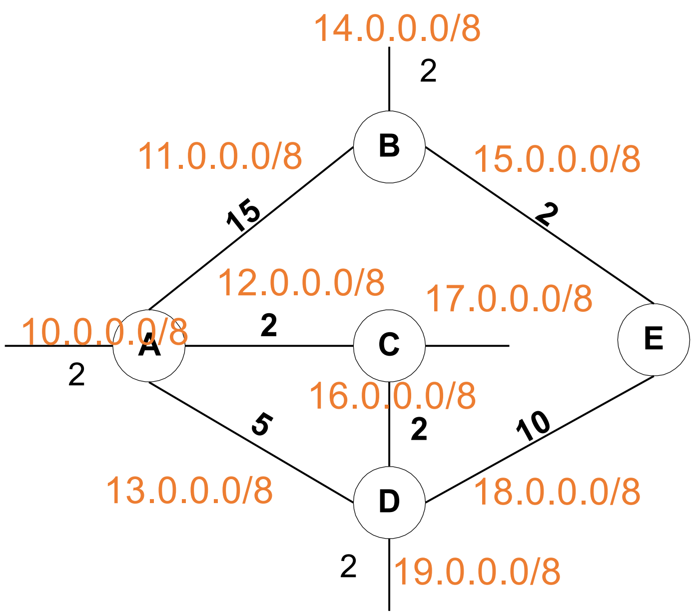
Link State information from RouterE

We now get the following link-state information from **RouterE**:

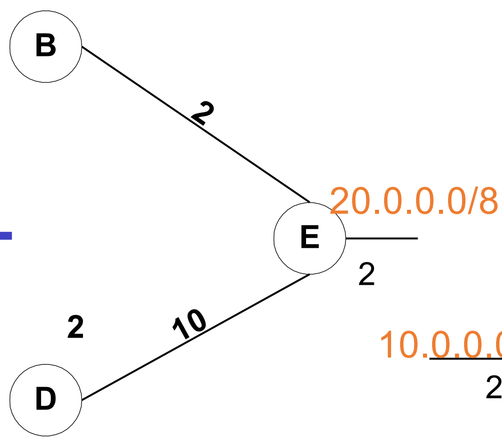
- Connected to RouterB on network 15.0.0.0/8, cost of 2
- Connected to RouterD on network 18.0.0.0/8, cost of 10
- Have a "leaf" network 20.0.0.0/8, cost of 2

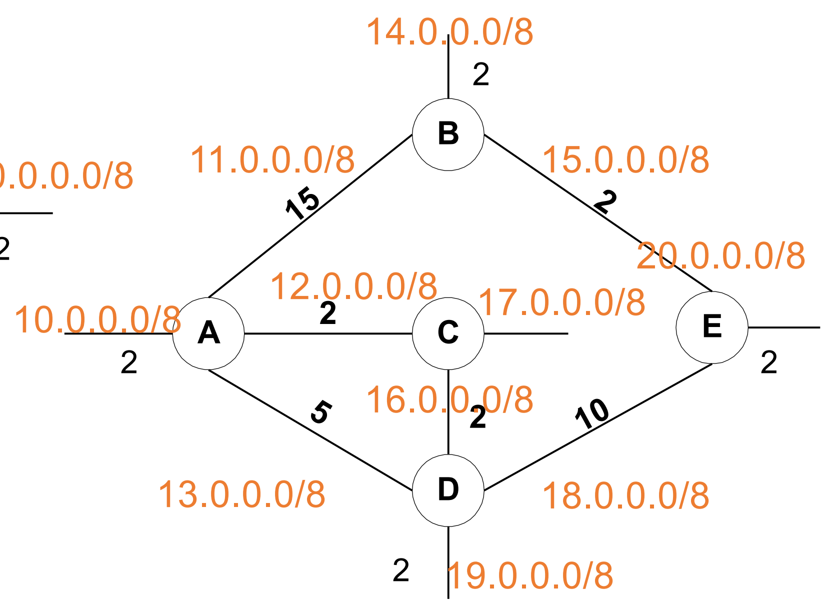


Now, RouterA attaches the two graphs…



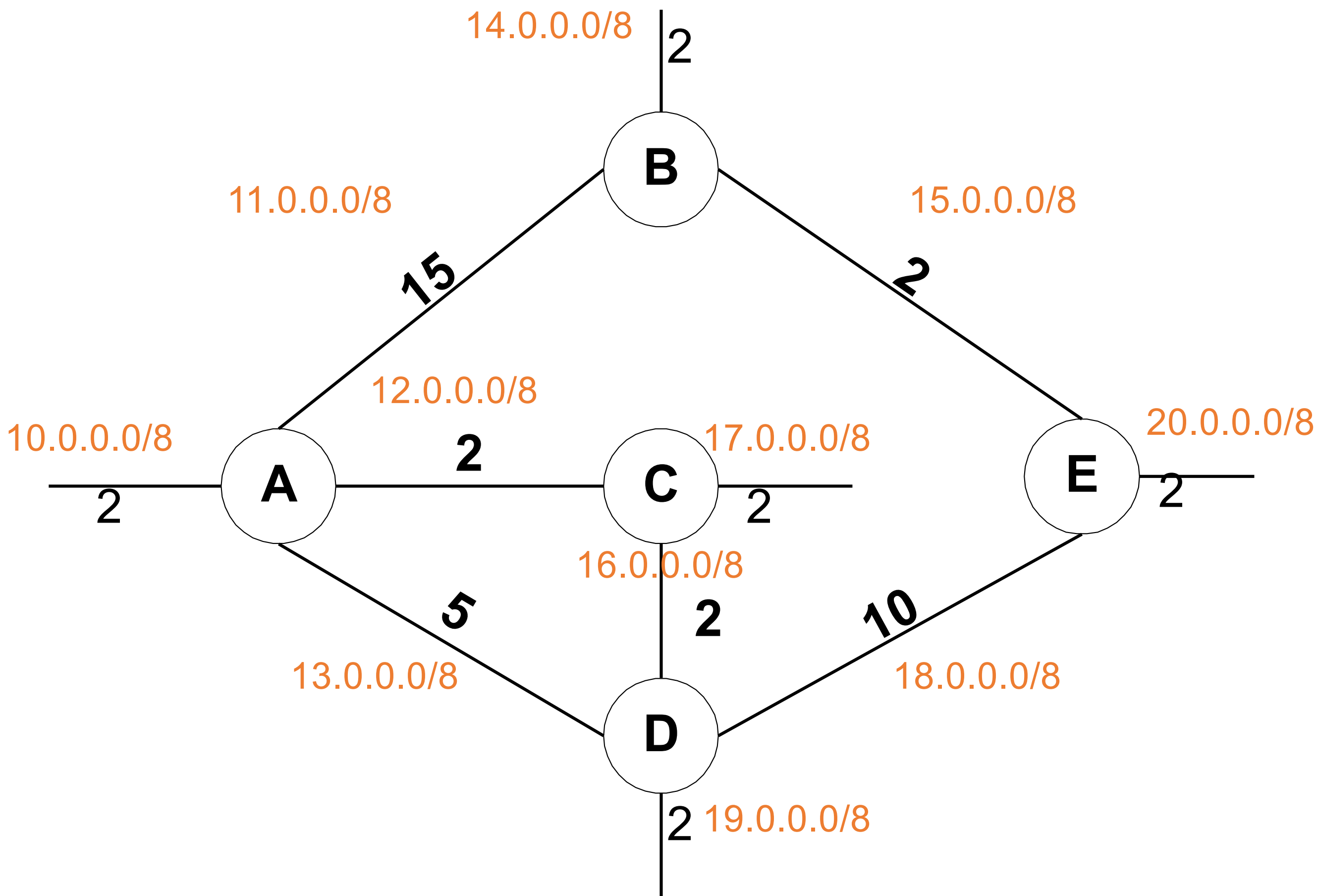
+





Topology

- Using the topological information we listed, RouterA has now built a complete topology of the network.
- The next step is for the link-state algorithm to find the best path to each node and leaf network.



Extra: Simplified Link State Example

RouterA's Topological Data Base (Link State Database)

RouterB:

- **Connected to RouterA on network 11.0.0.0/8, cost of 15**
- **Connected to RouterE on network 15.0.0.0/8, cost of 2**
- **Has a "leaf" network 14.0.0.0/8, cost of 15**

RouterC:

- **Connected to RouterA on network 12.0.0.0/8, cost of 2**
- **Connected to RouterD on network 16.0.0.0/8, cost of 2**
- **Has a "leaf" network 17.0.0.0/8, cost of 2**

RouterD:

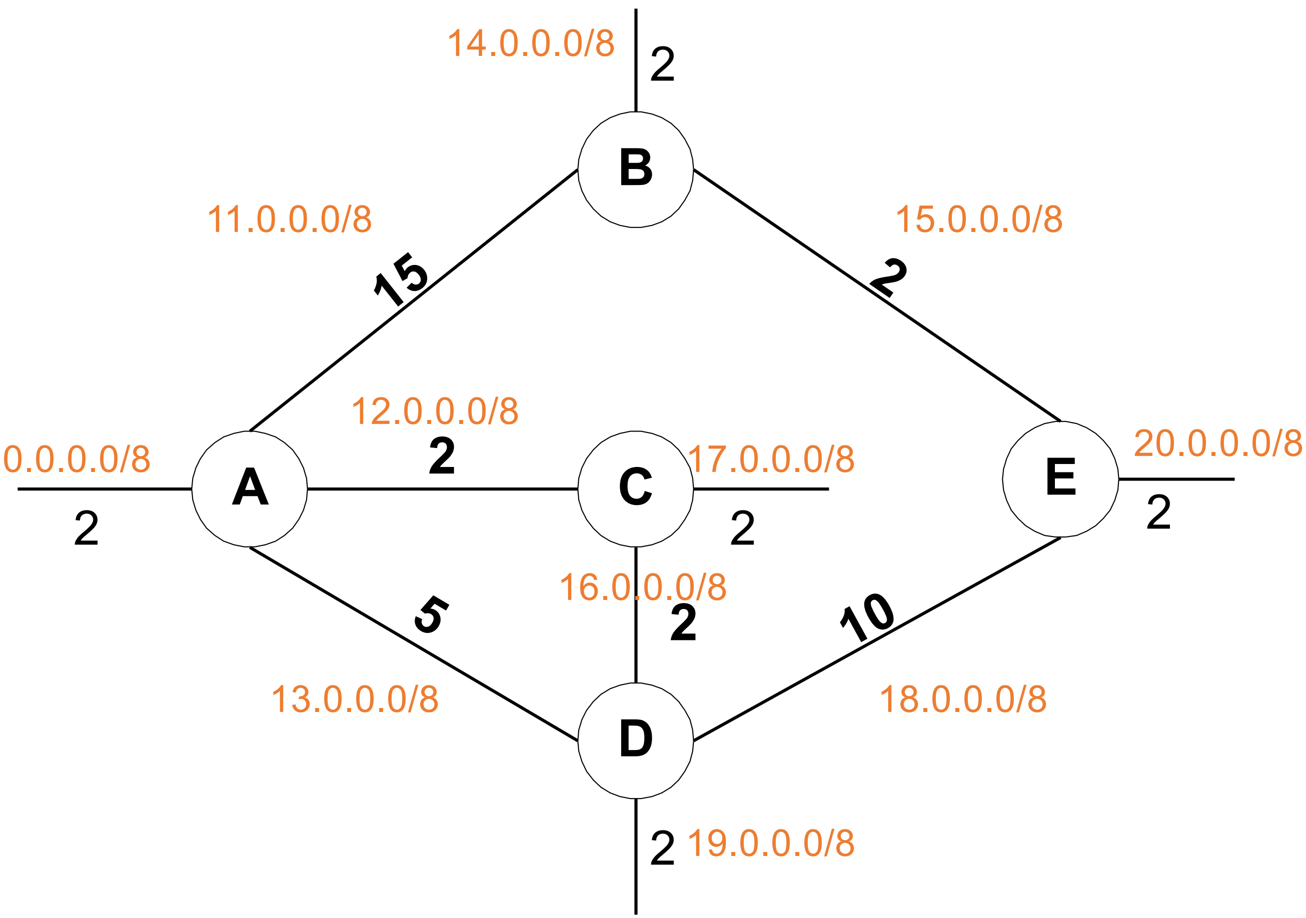
- **Connected to RouterA on network 13.0.0.0/8, cost of 5**
- **Connected to RouterC on network 16.0.0.0/8, cost of 2**
- **Connected to RouterE on network 18.0.0.0/8, cost of 2**
- **Has a "leaf" network 19.0.0.0/8, cost of 2**

RouterE:

- **Connected to RouterB on network 15.0.0.0/8, cost of 2**
- **Connected to RouterD on network 18.0.0.0/8, cost of 10**
- **Has a "leaf" network 20.0.0.0/8, cost of 2**

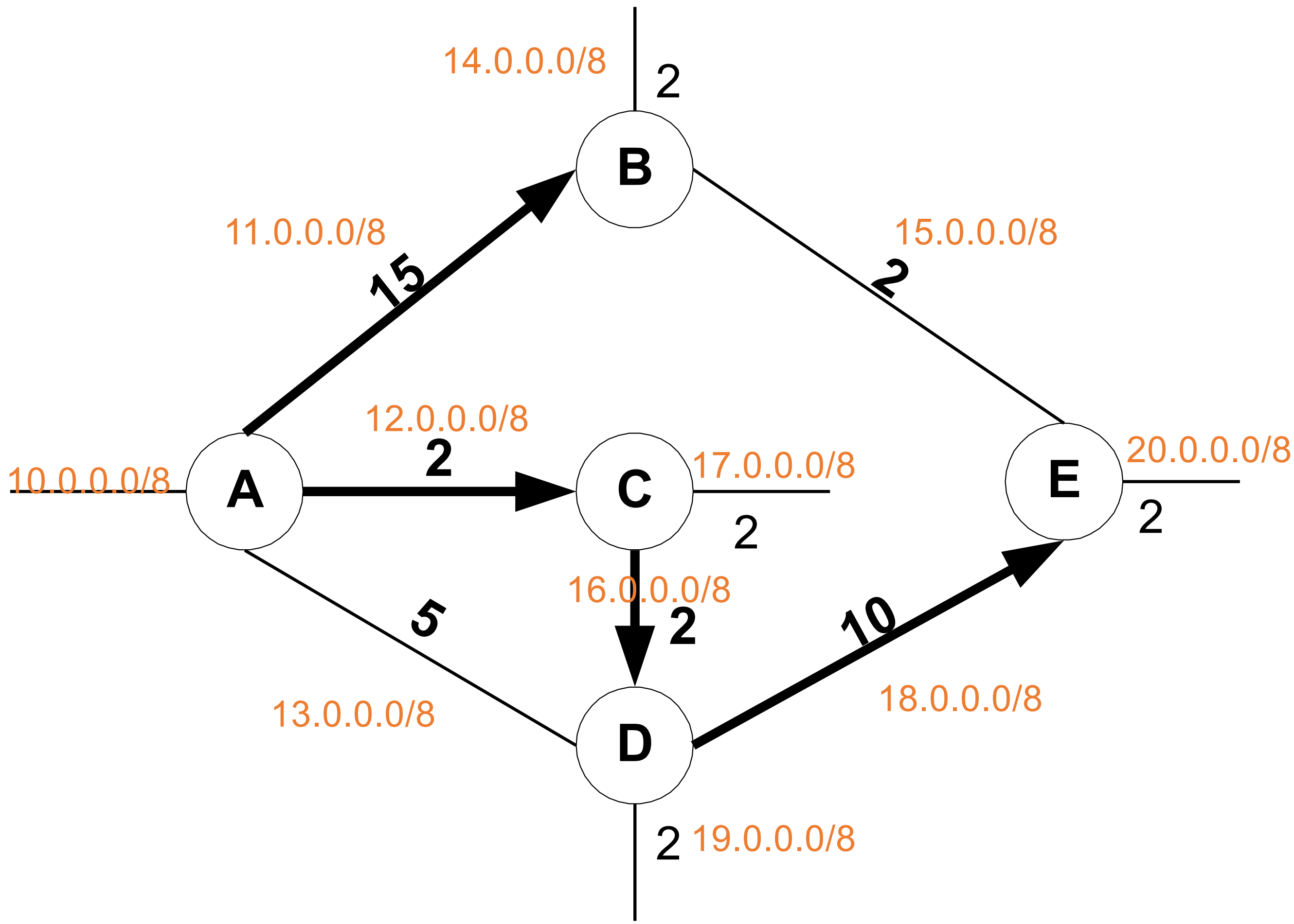
Choosing the Best Path

- Using the link-state algorithm RouterA can now proceed to find the shortest path to each leaf network.



Choosing the Best Path

Now RouterA knows the best path to each network, creating an SPT (Shortest Path Tree).

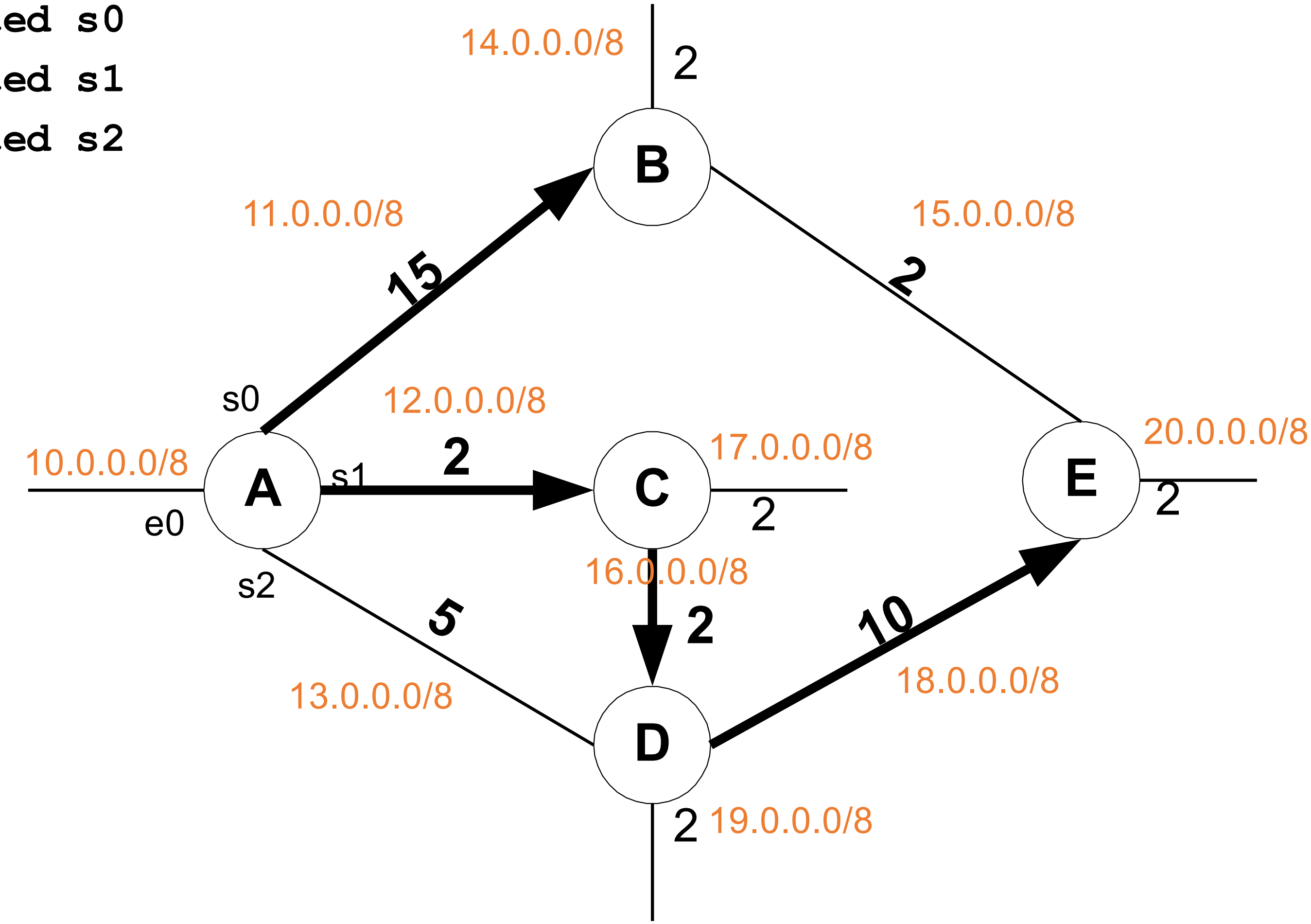


SPT Results Get Put into the Routing Table

<u>RouterA's Routing Table</u>

```
10.0.0.0/8   connected e0
11.0.0.0/8   connected s0
12.0.0.0/8   connected s1
13.0.0.0/8   connected s2

14.0.0.0/8   17 s0
15.0.0.0/8   17 s1
16.0.0.0/8    4 s1
17.0.0.0/8    4 s1
18.0.0.0/8   14 s1
19.0.0.0/8    6 s1
20.0.0.0/8   16 s1
```



Introduction to OSPF Concepts

Introducing OSPF and Link State Concepts

- **Advantages of OSPF**
- **Brief History**
- **Terminology**
- **Link State Concepts**

Introducing the OSPF Routing Protocol

- **Metric based on Cost (Bandwidth)**
- **Hello Protocol**
- **Steps to OSPF Operation**
- **DR/BDR**
- **OSPF Network Types**

OSPF’s Metric is Cost (Bandwidth)

RFC 2328, OSPF version 2, J. Moy

- “A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic.”
- RFC 2328 does not specify any values for cost.
- Bay and some other vendors use a default cost of 1 on all interfaces, essentially making the OSPF cost reflect hop counts.

OSPF's Metric is Cost (Bandwidth)

Cisco: Cost = Bandwidth

- Cisco uses a default cost of **10^8/bandwidth**
- Default bandwidth of the interface (bandwidth command)
- ***10^8 (100,000,000) as the reference bandwidth***: This is used so that the faster links (higher bandwidth) have lower costs.
  - Routing metrics, lower the cost the better the route.
  - I.e. RIP: 3 hops is better than 10 hops
  - Extra: The reference bandwidth can be modified to accommodate networks with links faster than 100,000,000 bps (100 Mbps). See **`ospf auto-cost reference-bandwidth`** command.
- ***Cost of a route is the cumulative costs of the outgoing interfaces from this router to the network.***

OSPF’s Metric is Cost (Bandwidth)

Cisco default interface costs:

- 56-kbps serial link = 1785
- 64-kbps serial link = 1562   128-kbps serial link = 781
- T1 (1.544-Mbps serial link) = 64
- E1 (2.048-Mbps serial link) = 48
- 4-Mbps Token Ring = 25
- Ethernet = 10
- 16-Mbps Token Ring = 6
- Fast Ethernet = 1
- Problem: Gigabit Ethernet and faster = 1

Cost = 100,000,000/Bandwidth

Notes:

- Cisco routers default to T1 (1.544 Mbps) on all serial interfaces and require manual modification with the bandwidth command.
- **`ospf auto-cost reference-bandwidth`** *reference-bandwidth* can be used to modify the reference-bandwidth for higher speed interfaces

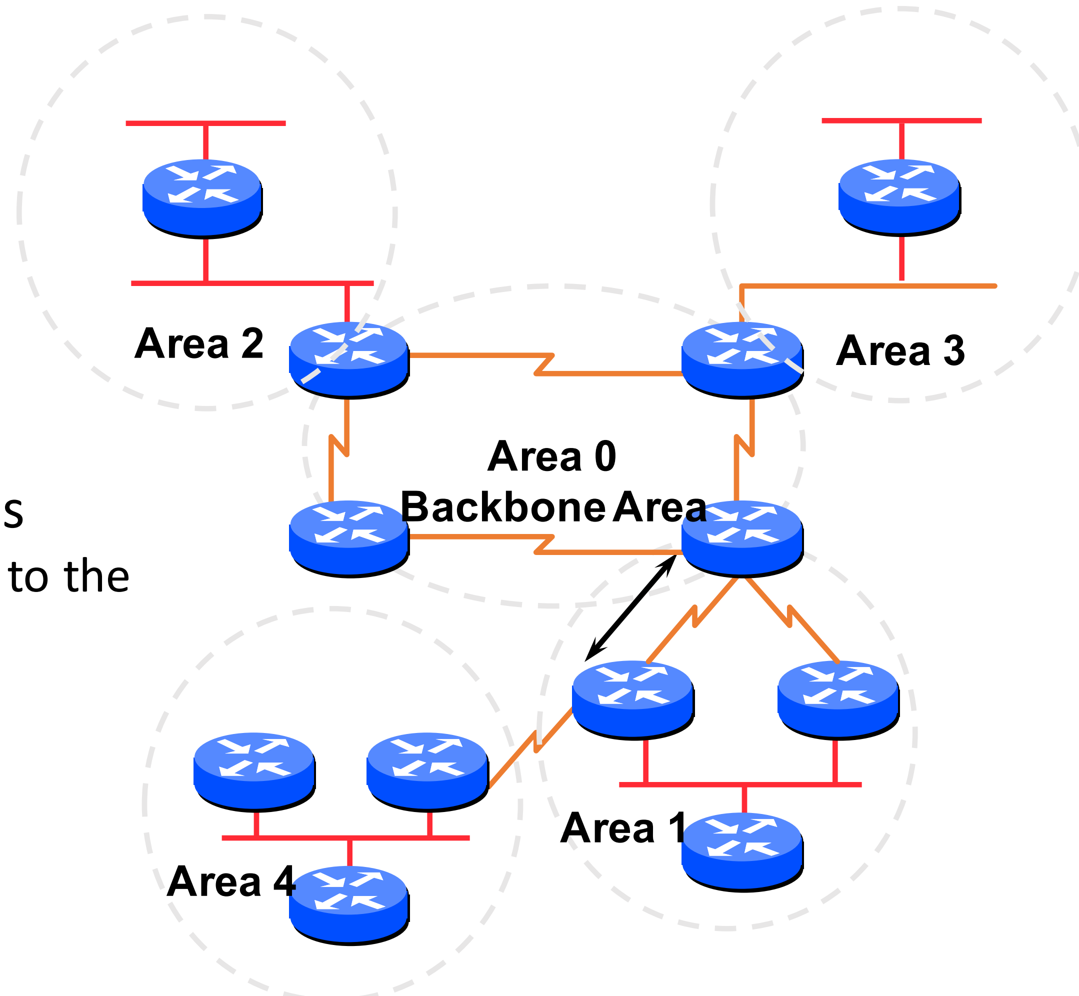
OSPF’s Metric is Cost (Bandwidth)

Few final notes

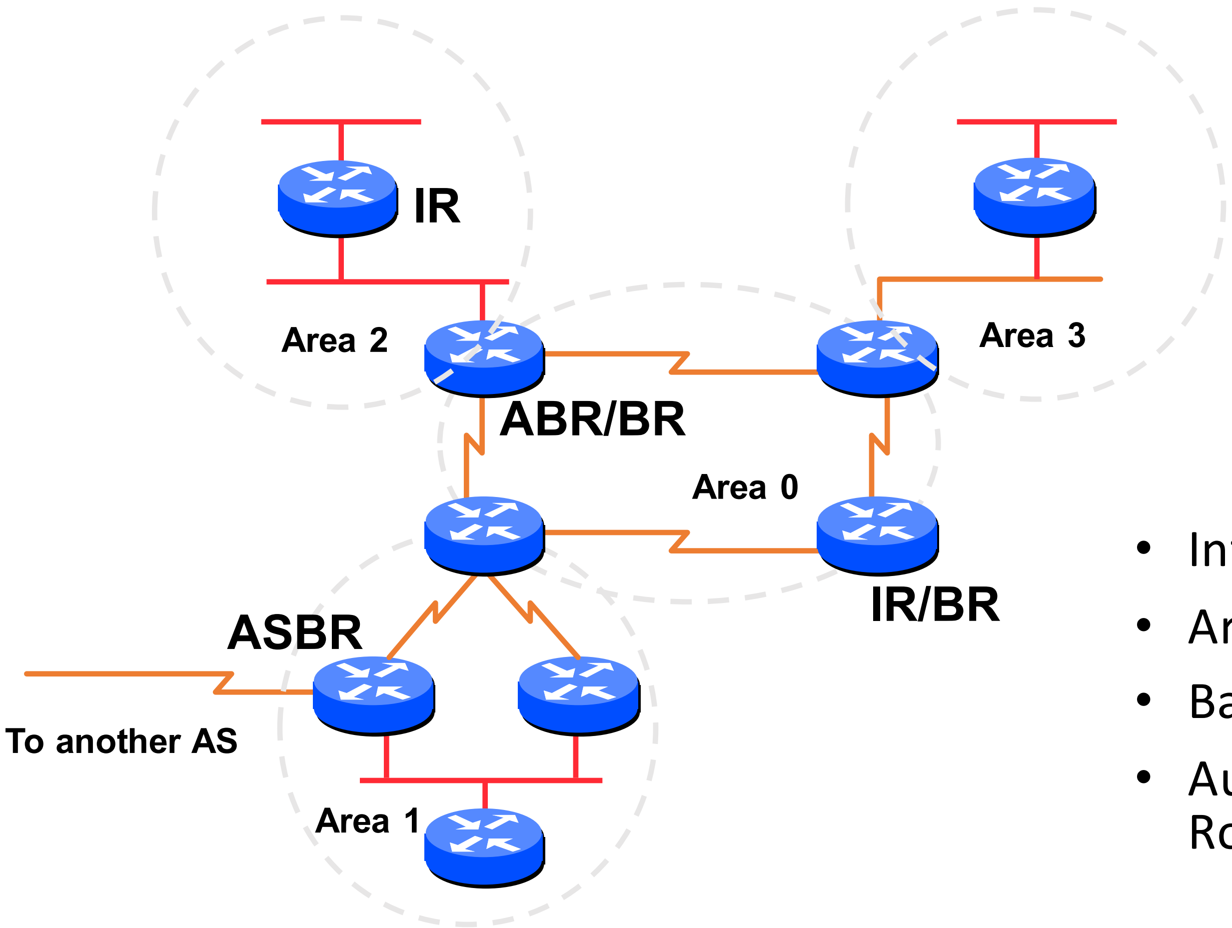
- For serial links, if it is not a T1 line, use the bandwidth command to configure the interface to the right bandwidth
- Both sides of the link should have the same bandwidth value
- If you use the command **`ospf auto-cost reference-bandwidth`** ***`reference-bandwidth`***, configure all of the routers to use the same value.

OSPF Areas

- Group of contiguous nodes/networks
- Per area topology DB
  - Invisible outside the area
  - Reduces routing traffic
- Backbone Area is contiguous
  - All others areas must connect to the backbone
- Virtual Links

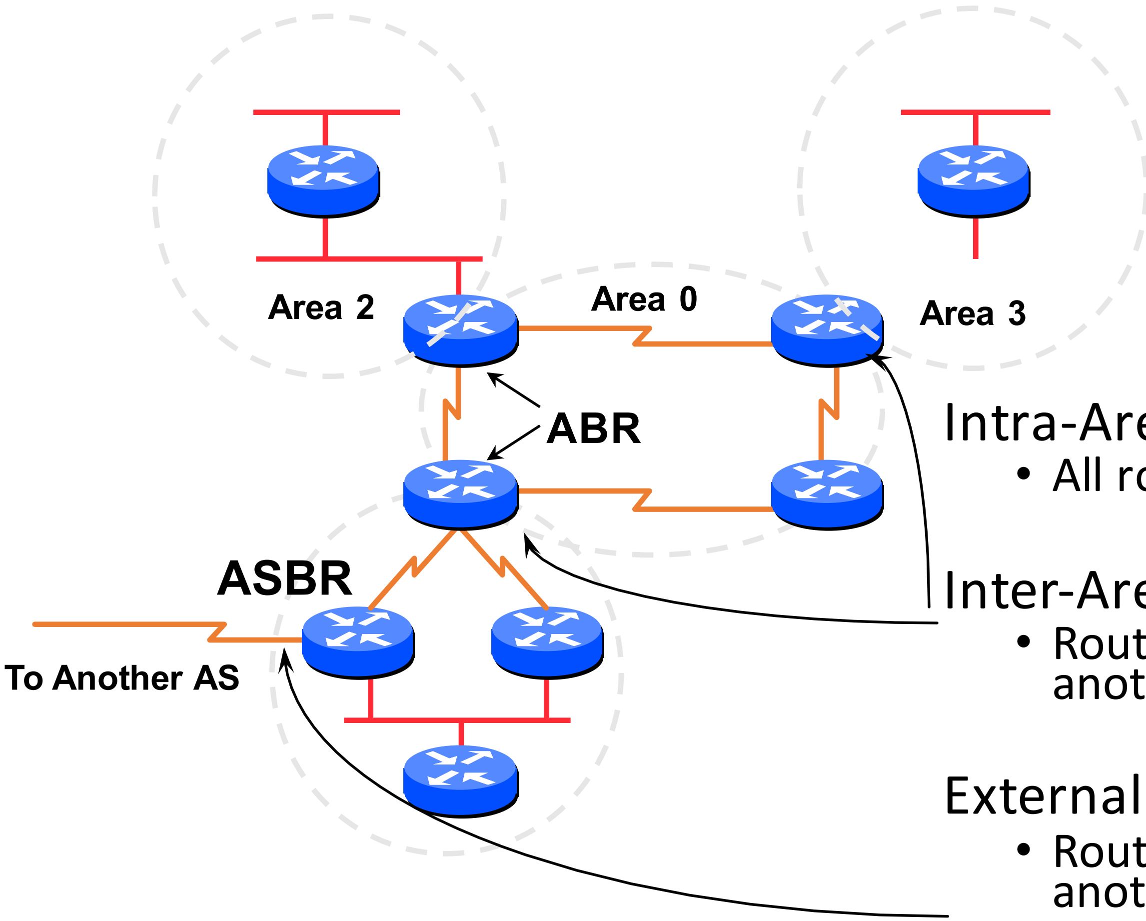


Router Classification



- Internal Router (IR)
- Area Border Router (ABR)
- Backbone Router (BR)
- Autonomous System Border Router (ASBR)

OSPF Route Types



Intra-Area Route

- All routes within an area

Inter-Area Route

- Routes announced from area to another by an ABR

External Route

- Routes imported into OSPF from another protocol or Static routes

Topology/Links-State DB

- A router has a separate DB for each area it belongs
- All routers within an area have an identical DB
- SPF calculation is done separately for each area
- LSA flooding is limited to the particular area

OSPF Packet Types

OSPF Packet Type	Description
Type 1 - Hello	Establishes and maintains adjacency information with neighbors
Type 2 - Database description packet (DBD)	Describes the content of the link-state database on an OSPF router
Type 3 - Link-state request (LSR)	Requests specific pieces of a link-state database
Type 4 - Link-state update (LSU)	Transports link-state advertisements (LSAs) to neighbor routers
Type 5 - Link-state acknowledgement (LSAck)	Acknowledges receipt of a neighbor's LSA

The OSPF Packet Header

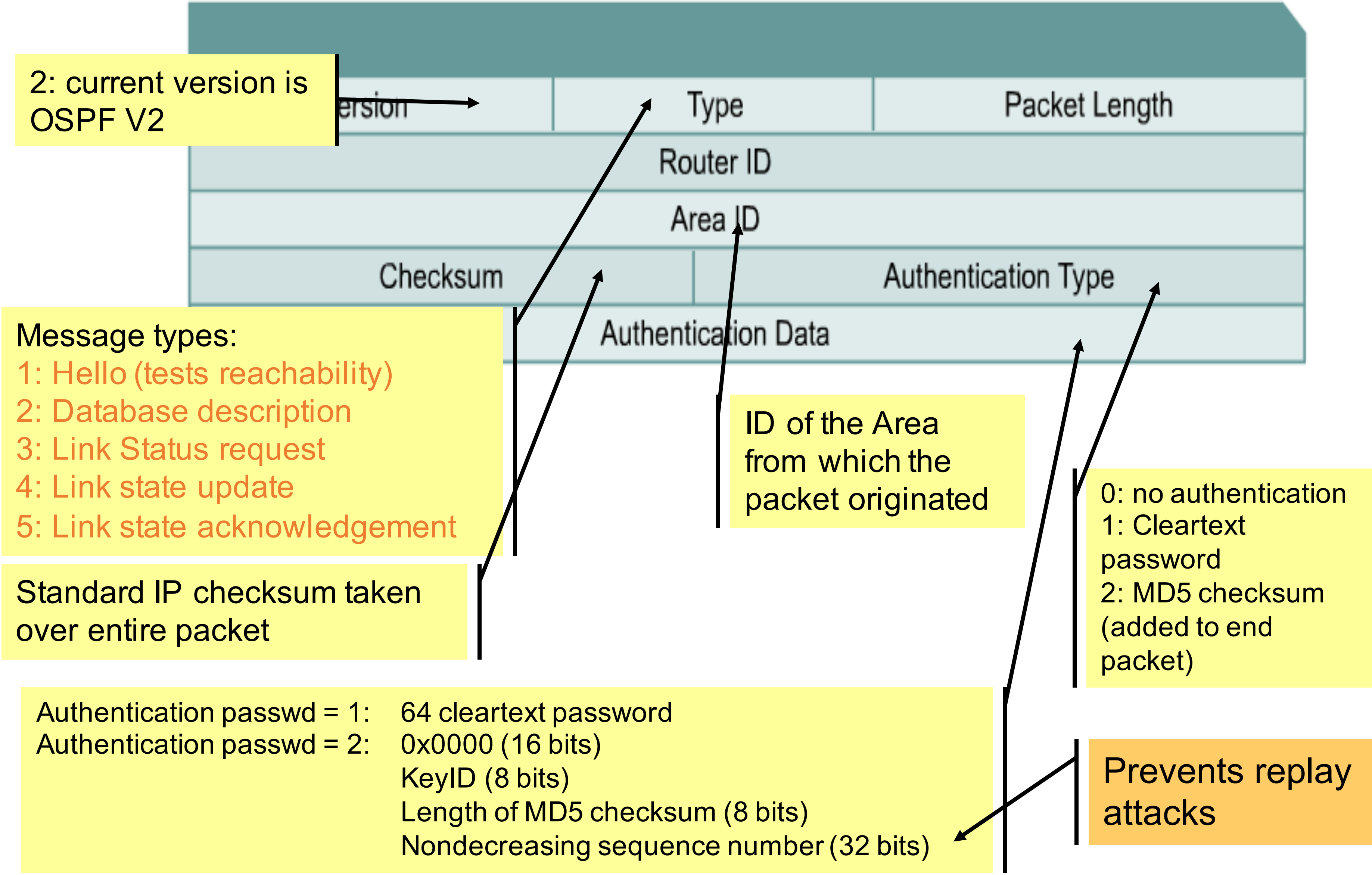
Version	Type	Packet Length
Router ID		
Area ID		
Checksum	Authentication Type	
Authentication Data		

OSPF Packet Format

MAC Hdr	IP Hdr	OSPF Hdr	OSPF type-specific Data

- MAC Header
  - Src – MAC addr of sending interface
  - Dst – 0100:5E00:0005 or 0100:5E00:0006
  - Type – 0x0800
- IP Header
  - Src – IP addr of sending interface
  - Dst – 224.0.0.5 or 224.0.0.6
  - Protocol – 89
- OSPF Header
  - Version
  - Type
  - Router ID
  - Area ID

OSPF Packet Format



OSPF Hello Protocol

Network Mask		
Hello Interval	Options	Router Priority
Dead Interval		
Designated Router		
Backup Designated Router		
Neighbor Router ID		
Neighbor Router ID		
(additional Neighbor Router ID fields can be added to the end of the header, if necessary)		

Hello subprotocol is intended to perform the following tasks within OSPF:

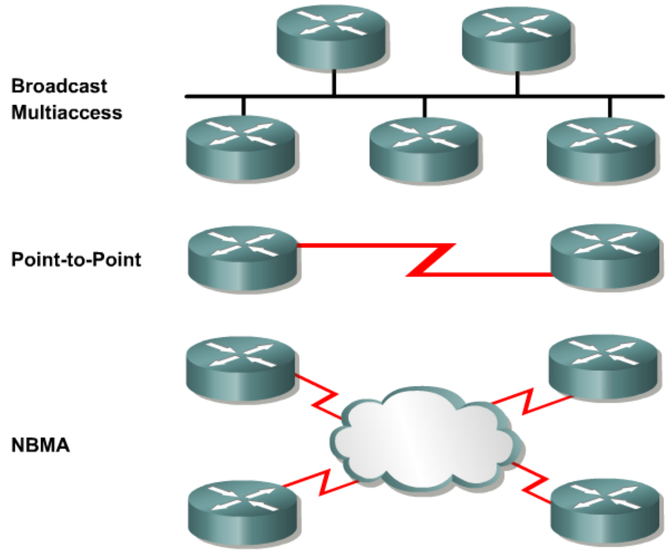
- Dynamic neighbor discovery
- Detect unreachable neighbors
- Ensure two-way communications between neighbors
- Ensure correctness of basic interface parameters between neighbors
- Provide necessary information for the election of the Designated and Backup Designated routers on a LAN segment (coming)

OSPF Hello Protocol

Network Mask		
Hello Interval	Options	Router Priority
Dead Interval		
Designated Router		
Backup Designated Router		
Neighbor Router ID		
Neighbor Router ID		
(additional Neighbor Router ID fields can be added to the end of the header, if necessary)		

- OSPF routers send **Hellos** on OSPF enabled interfaces:
  - Default every 10 seconds on multi-access and point-to-point segments
  - Default every 30 seconds on NBMA segments (Frame Relay, X.25, ATM)
  - Most cases OSPF Hello packets are sent as multicast to ALLSPFRouters (224.0.0.5)
- **HelloInterval** - Cisco default = 10 seconds or 30 seconds and can be changed with the command **`ip ospf hello-interval`**.
- **RouterDeadInterval** - The period in seconds that the router will wait to hear a Hello from a neighbor before declaring the neighbor down.
  - Cisco uses a default of ***four-times the*** **HelloInterval** (4 x 10 sec. = 40 seconds, 120 secconds for NBMA) and can be changed with the command **`ip ospf dead-interval`**.
- **Note**: For routers to become adjacent, the **Hello**, **DeadInterval** and **network types** must be identical between routers or Hello packets get dropped!

Network Types – *more later*




```
show ip ospf interface
```

Network Type	Table Title
Broadcast multiaccess	Ethernet, Token Ring, or FDDI
Nonbroadcast multiaccess	Frame Relay, X.25, SMDS
Point-to-point	PPP, HDLC
Point-to-multipoint	Configured by an administrator

Unless you are configuring an NBMA network like Frame Relay, this won't be an issue.

- Many administrators prefer to use point-to-point or point-to-multipoint for NMBA to avoid the DR/BDR and full-mesh issues.

OSPF packet types

Type	Description
1	Hello (establishes and maintains adjacency relationships with neighbors)
2	Database description packet (describes the contents of an OSPF router's link-state database) OSPF Type-2 (DBD)
3	Link-state request (requests specific pieces of a neighbor router's link-state database) OSPF Type-3 (LSR)
4	Link-state update (transports link-state advertisements (LSAs) to neighbor routers) OSPF Type-4 (LSU)
5	Link-state acknowledgement (Neighbor routers acknowledge receipt of the LSAs) OSPF Type-5 (LSAck)

Steps to OSPF Operation

Steps of OSPF Operation

- Establish router adjacencies
- Elect a designated router and a backup designated router
- Discover routes
- Select appropriate route to use
- Maintain routing information

OSPF States

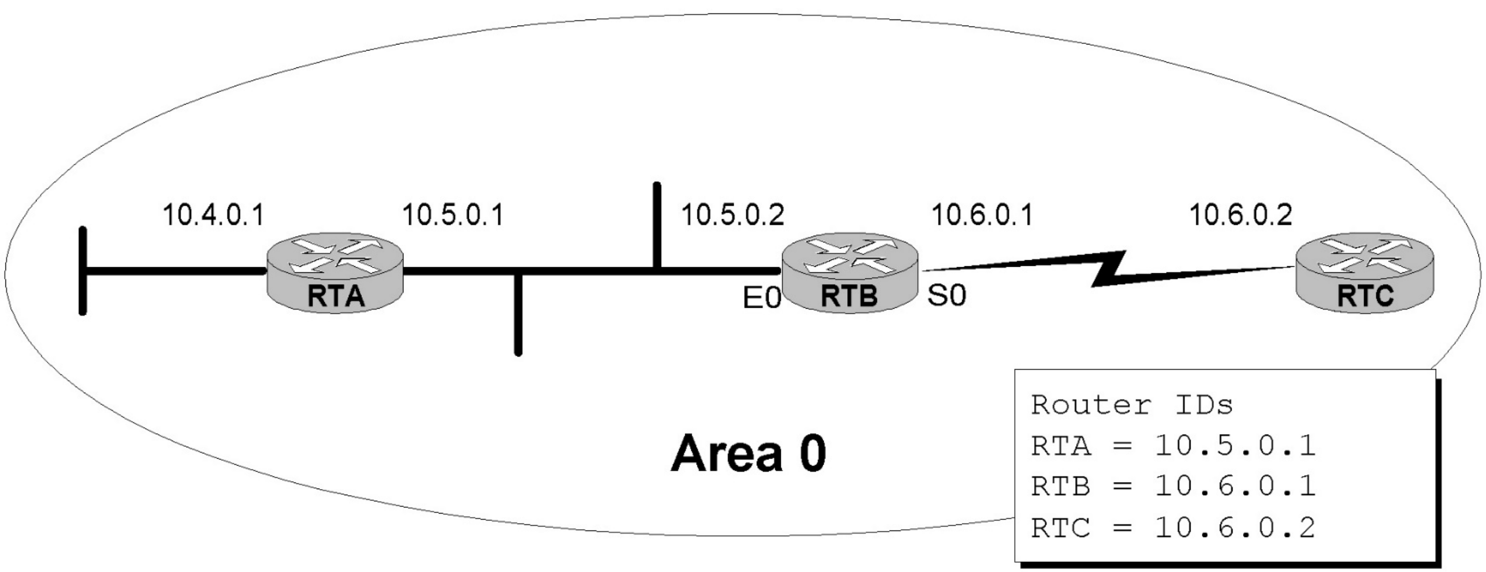
- Down
- Init
- Two-way
- ExStart
- Exchange
- Loading
- Full adjaceny

Steps to OSPF Operation with States

1. **Establishing router adjacencies (Routers are adjacent)**
   - Down State – No Hello received
   - Init State – Hello received, but not with this router's Router ID
     - "Hi, my name is Carlos." "Hi, my name is Maria."
   - Two-way State – Hello received, and with this router's Router ID
     - "Hi, Maria, my name is Carlos." "Hi, Carlos, my name is Maria."
2. **Electing DR and BDR – Multi-access (broadcast) segments only**
   - ExStart State with DR and BDR
   - Two-way State with all other routers
3. **Discovering Routes**
   - ExStart State
   - Exchange State
   - Loading State
   - Full State (Routers are "fully adjacent")
4. Calculating the Routing Table
5. Maintaining the LSDB and Routing Table

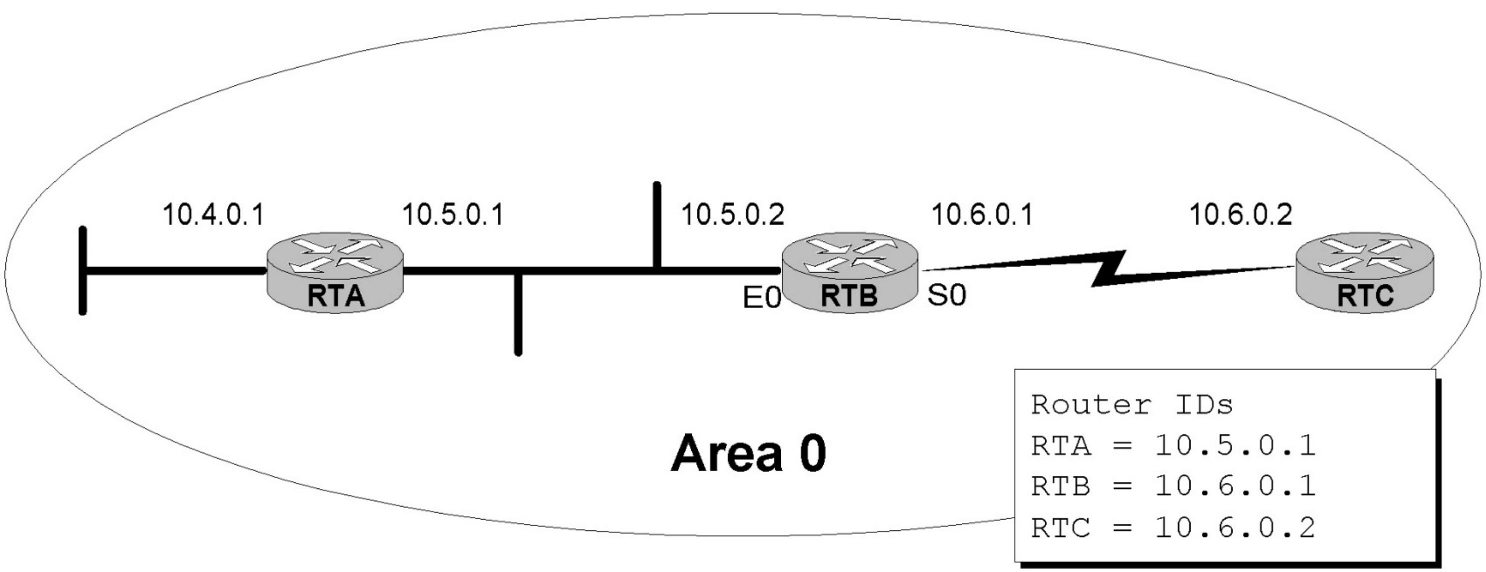
1. Establishing Adjacencies

- Initially, an OSPF router interface is in the **down state**.
- An OSPF interface can transition back to this state if it has not received a Hello packet from a neighbor within the RouterDeadInterval time (40 seconds unless NBMA, 120 seconds).
- In the **down state**, the OSPF process has not exchanged information with any neighbor.
- OSPF is waiting to enter the **init state**.
- An OSPF router tries to form an adjacency with at least one neighbor for each IP network it's connected to.

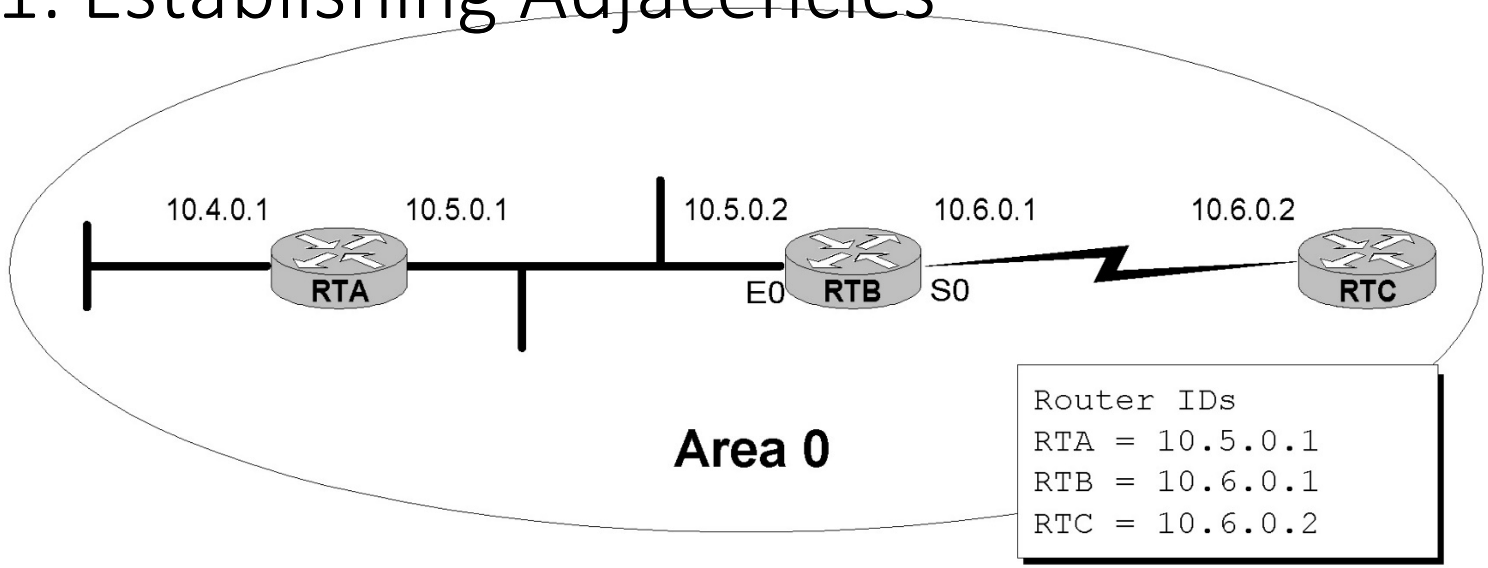


1. Establishing Adjacencies

- The process of establishing adjacencies is asymmetric, meaning the states between two adjacent routers may be different as they both transition to full state.
- RTB perspective and assuming routers are configured correctly.
- Trying to start a relationship and wanting to enter the **init state** or really the **two-way-state**
- RTB begins multicasts **OSPF Hello packets** (**224.0.0.5**, AllSPFRouters), advertising its own **Router ID**.
  - 224.0.0.5: All OSPF routers should be able to transmit and listen to this address.

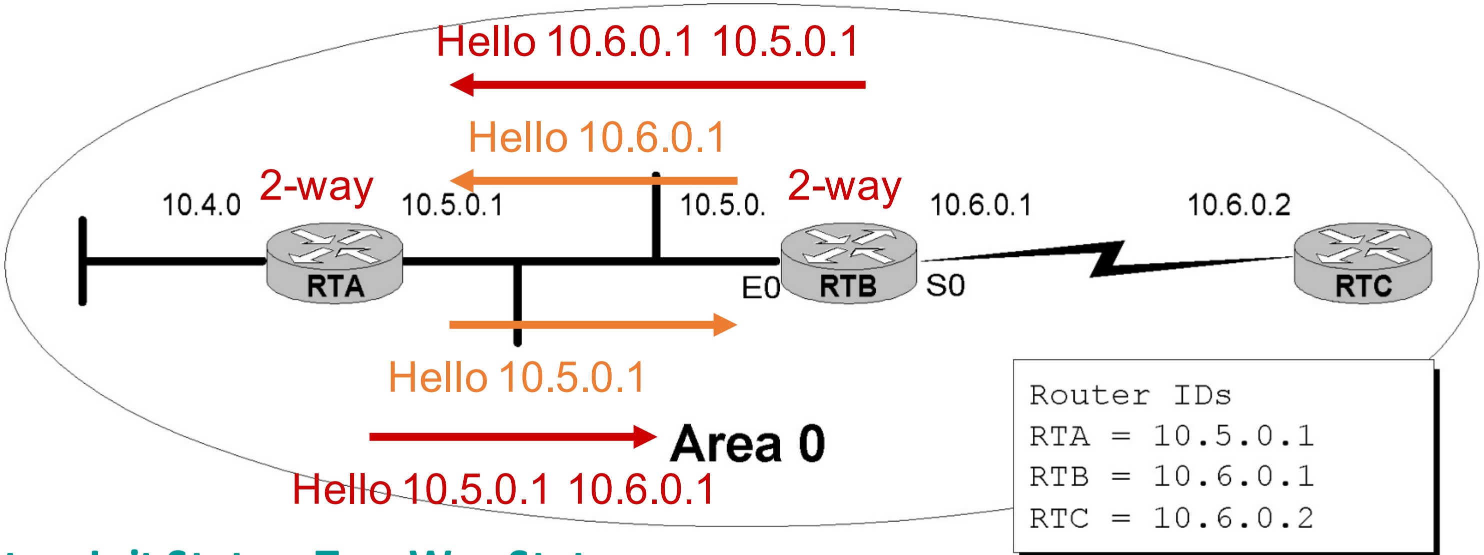


1. Establishing Adjacencies



- **Router ID** = Highest loopback address else highest active IP address.
- **Loopback address** has the advantage of never going down, thus diminishing the possibility of having to re-establish adjacencies. (more in a moment)
  - **Use private ip addresses for loopbacks**, so you do not inadvertently advertise a route to a real network that does not exist on your router.
- **For routers to become adjacent**, the **Hello**, **DeadInterval** and **network types** must be identical between routers or Hello packets get dropped!

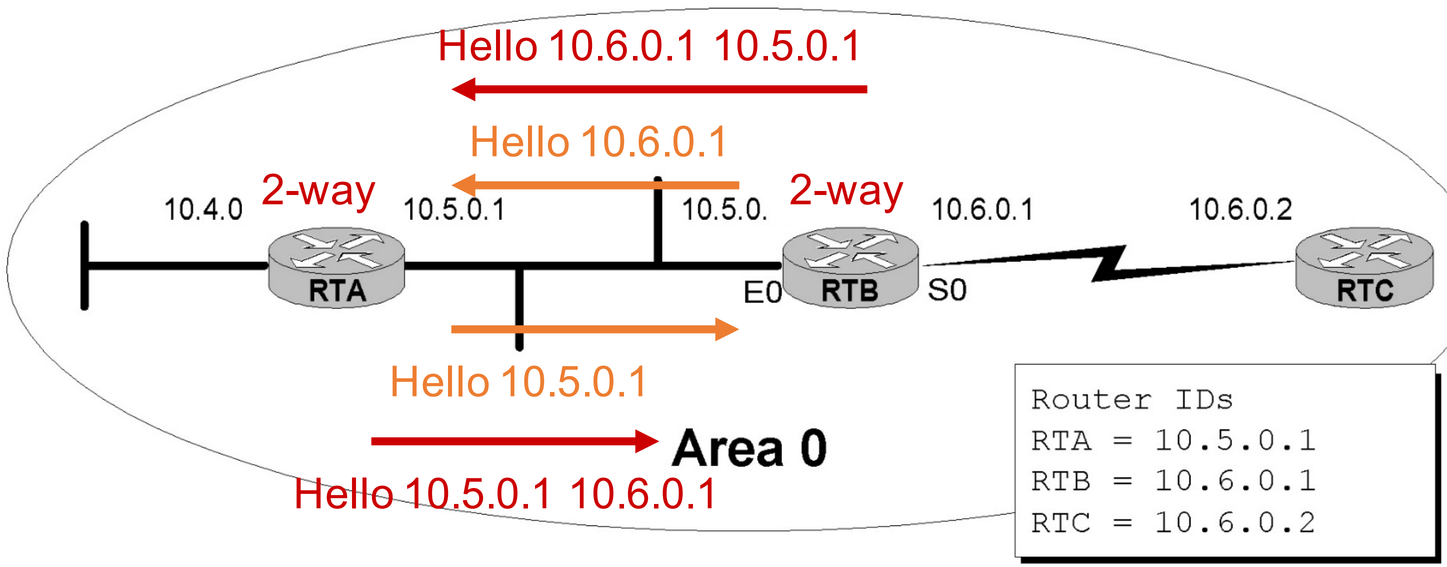
1. Establishing Adjacencies



Down State - Init State – Two Way State

- **Down State** - OSPF routers send Type 1 Hello packets at regular intervals (10 sec.) to establish neighbors.
- When a router (sends or) receives its first **Hello packet**, it enters the **init state**, indicating that the Hello packet was received but did not contain the Router ID of the receiving router in the list of neighbors, so two-way communications is not yet ensured.
- As soon as the router sends a Hello packet to the neighbor with its RouterID and the neighbor sends a Hello packet packet back with that Router ID, the router's interface will transition to the **two-way state**.
- Now, the router is ready to take the relationship to the next level.

1. Establishing Adjacencies



From **Init state** to the **Two-way state**

- RTB receives Hello packets from RTA and RTC (its neighbors), and sees its own Router ID (10.6.0.1) in the Neighbor ID field.
- RTB declares takes the relationship to a new level, and declares a **two-way state** between itself and RTA, and itself and RTC.
- As soon as the router sends a **Hello packet** to the neighbor with its RouterID and the neighbor sends a Hello packet packet back with that Router ID, the router's interface will transition to the two-way state.
- Now, the router is ready to take the relationship to the next level.

1. Establishing Adjacencies

Two-way state

- RTB now decides who to establish a “full adjacency” with depending upon the type of network that the particular interfaces resides on.
- **Note**: The term adjacency is used to both describe routers reaching 2-way state and when they reach full-state. Not to go overboard on this, but technically OSPF routers are adjacent when the FSM reaches full-state and IS-IS is considered adjacent when the FSM reaches 2-way state.

Two-way state to ExStart state

- If the interface is on a point-to-point link, the routers becomes adjacent with its sole link partner (aka “soul mates”), and take the relationship to the next level by entering the **ExStart state**. (coming soon)

Remaining in the two-way state

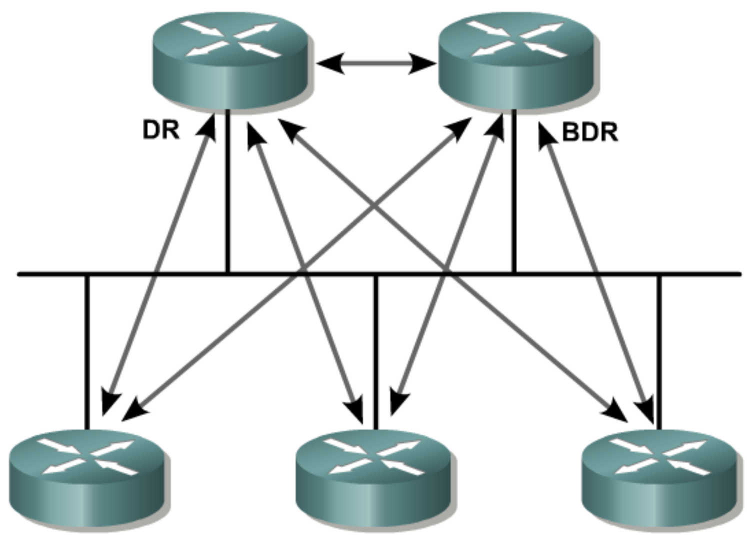
- If the interface is on a multi-access link (Ethernet, Frame Relay, …) RTB must enter an election process to see who it will establish a full adjacency with, and remains in the **two-way state**. (Next!)

Steps to OSPF Operation with States

1. **Establishing router adjacencies (Routers are adjacent)**
   - Down State – No Hello received
   - Init State – Hello received, but not with this router's Router ID
     - "Hi, my name is Carlos." "Hi, my name is Maria."
   - Two-way State – Hello received, and with this router's Router ID
     - "Hi, Maria, my name is Carlos." "Hi, Carlos, my name is Maria."
2. **Electing DR and BDR – Multi-access (broadcast) segments only**
   - ExStart State with DR and BDR
   - Two-way State with all other routers
3. **Discovering Routes**
   - ExStart State
   - Exchange State
   - Loading State
   - Full State (Routers are "fully adjacent")
4. Calculating the Routing Table
5. Maintaining the LSDB and Routing Table

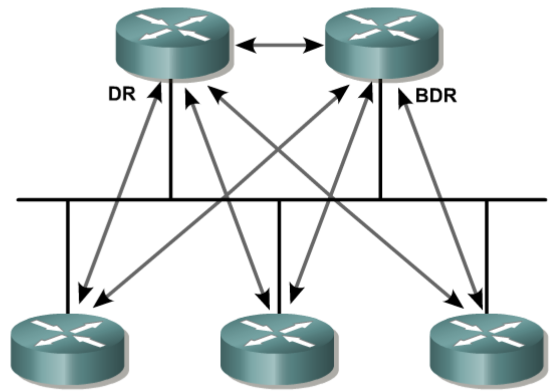
Electing the DR and BDR

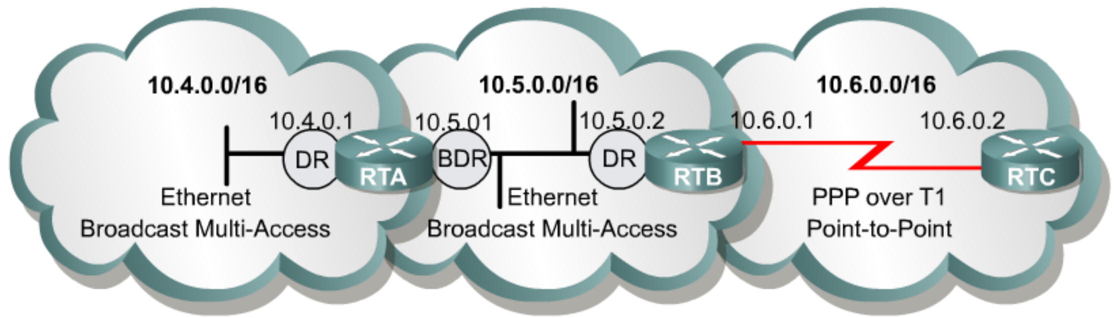
- On multi-access, broadcast links (Ethernet), a DR and BDR (if there is more than one router) need to be elected.
- *DR* - Designated Router
- *BDR* – Backup Designated Router
- DR's serve as collection points for Link State Advertisements (LSAs) on multi-access networks
- A BDR back ups the DR.
- If the IP network is *multi-access*, the OSPF routers will elect one DR and one BDR



- **Without a DR, the formation of an adjacency between every attached router would create many unnecessary LSA (Link State Advertisements), n(n-1)/2 adjacencies.**
- **Flooding on the network itself would be chaotic.**

Electing the DR and BDR



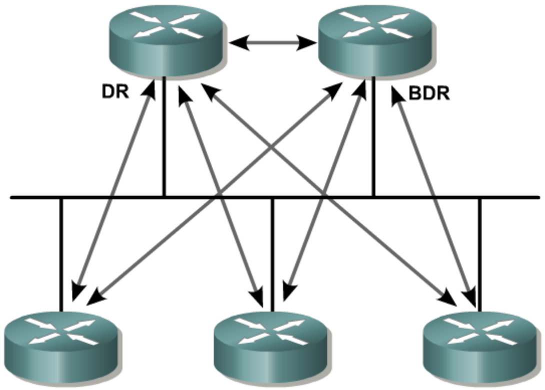


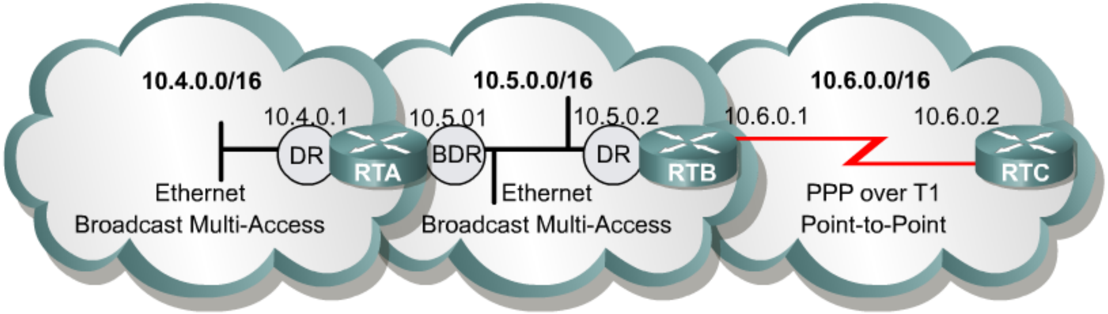
- Router with the **highest Router ID** is elected the DR, next is BDR.
- But like other elections, this one can be rigged.
- The router's priority field can be set to either ensure that it becomes the DR or prevent it from being the DR.

  `Rtr(config-if)#` **`ip ospf priority <0-255>`**

  - Higher priority becomes DR/BDR
  - Default = 1
  - 0 = Ineligible to become DR/BDR
- The router can be assigned a priority between 0 and 255, with 0 preventing this router from becoming the DR (or BDR) and 255 ensuring at least a tie. (The highest Router ID would break the tie.)

Electing the DR and BDR



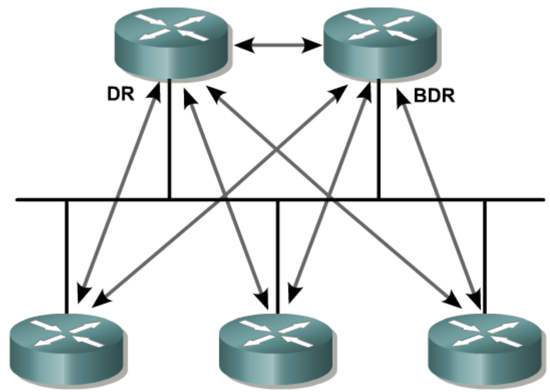


- All other routers, “**DROther**”, establish adjacencies with only the DR and BDR.
- DRother routers multicast LSAs to only the DR and BDR
  - (224.0.0.6 - all DR routers)
- DR sends LSA to all adjacent neighbors (DROthers)
  - (224.0.0.5 - all OSPF routers)

Backup Designated Router - BDR

- Listens, but doesn’t act.
- If LSA is sent, BDR sets a timer.
- If timer expires before it sees the reply from the DR, it becomes the DR and takes over the update process.
- The process for a new BDR begins.

Electing the DR and BDR



A new router enters the network:

- Once a DR is established, a new router that enters the network with a higher priority or Router ID it will **NOT** become the DR or BDR. (Bug in early IOS 12.0)
- Regardless of the priority or Router ID, that router will become a DROther.
- If DR fails, BDR takes over as DR and selection process for new BDR begins.

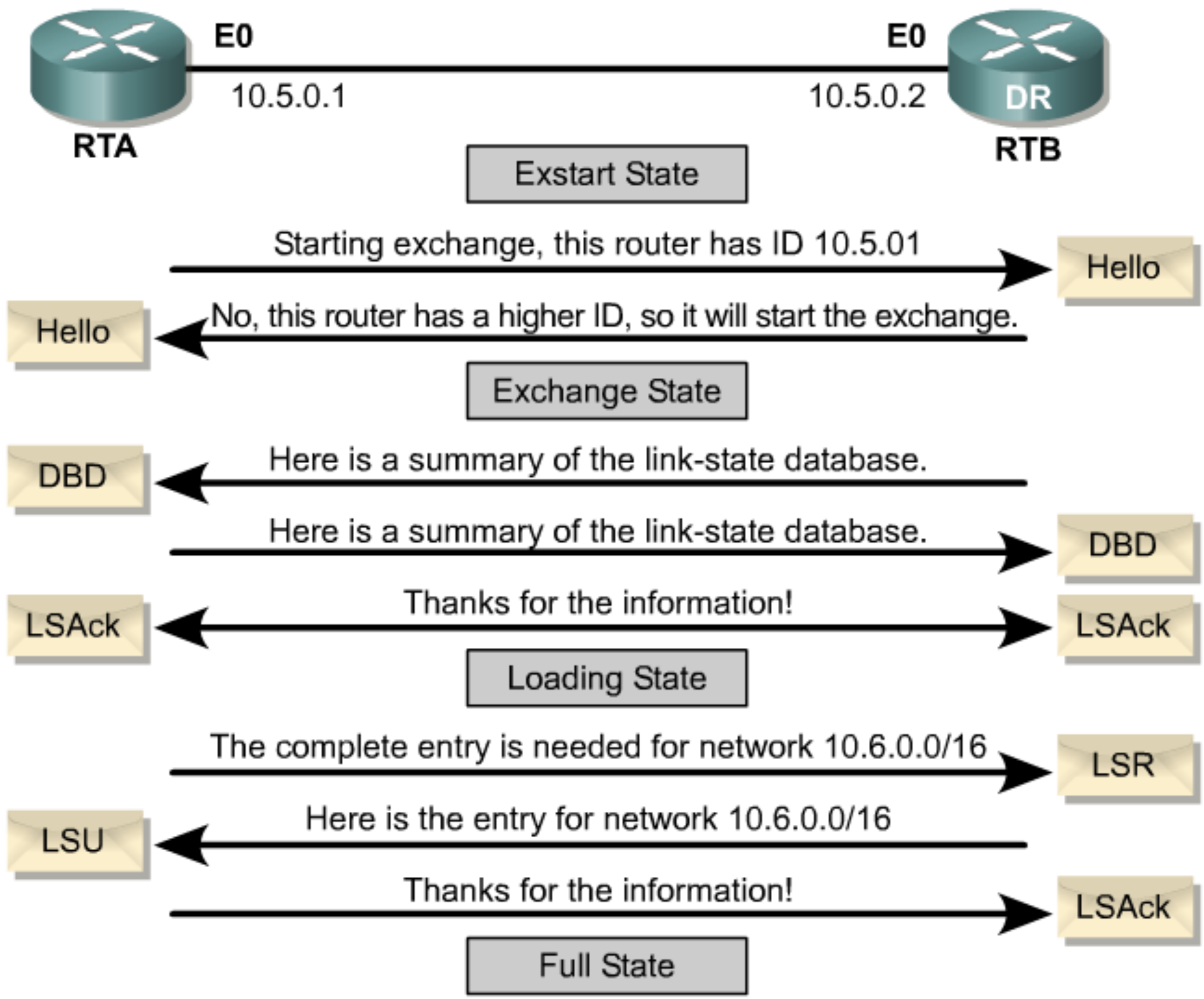
Clarifications

- **Hello packets** are still exchanged between all routers on a multi-access segment (DR, BDR, DROthers,….) to maintain neighbor adjacencies.
- **OSPF LSA packets** (coming) are packets which are sent from the BDR/DROthers to the DR, and then from the DR to the BDR/DROthers. (The reason for a DR/BDR.)
- Normal routing of **IP packets** still takes the lowest cost route, which might be between two DROthers.

Steps to OSPF Operation with States - Extra

1. Establishing router adjacencies
   - Down State – No Hello received
   - Init State – Hello received, but not with this router's Router ID
     - "Hi, my name is Carlos." "Hi, my name is Maria."
   - Two-way State – Hello received, and with this router's Router ID
     - "Hi, Maria, my name is Carlos." "Hi, Carlos, my name is Maria."
2. Electing DR and BDR – Multi-access (broadcast) segments only
   - ExStart State with DR and BDR
   - Two-way State with all other routers
3. Discovering Routes
   - ExStart State
   - Exchange State
   - Loading State
   - Full State
4. Calculating the Routing Table
5. Maintaining the LSDB and Routing Table

Steps to OSPF Operation with States Discovering Routes and Reaching Full State



ExStart State – the explanation

ExStart State

- This state starts the LSDB (Link State Data Base) synchronization process.
- This will prepare for initial database exchange.
- Routers are now ready to exchange routing information.
  - Between routers on a point-to-point network
  - On a multi-access network between the DRothers and the DR and BDR.
- Formally, routers in **ExStart state** are characterized as adjacent, but have not yet become "fully adjacent" as they have not exchanged data base information.

But who goes first in the exchange?

- **ExStart** is established by exchanging OSPF Type-2 DBD (Database Description) packets
- Purpose of **ExStart** is to establish a "***master/slave relationship"*** between the two routers decided by the **higher router id**.
- Once the roles are established they enter the **Exchange state**.

OSPF Database Description Packet Format

Version	Type	Packet Length
Router ID		
Area ID		
Checksum	Authentication Type	
Authentication Data		

Interface MTU	Options	00000I M MS
DD Sequence #		
LSA Headers		

I – Initial bit M – More bit MS – 1: master, 0: slave

Sequence #: Set by master in the 1st DD packet, and incremented in subsequent packets

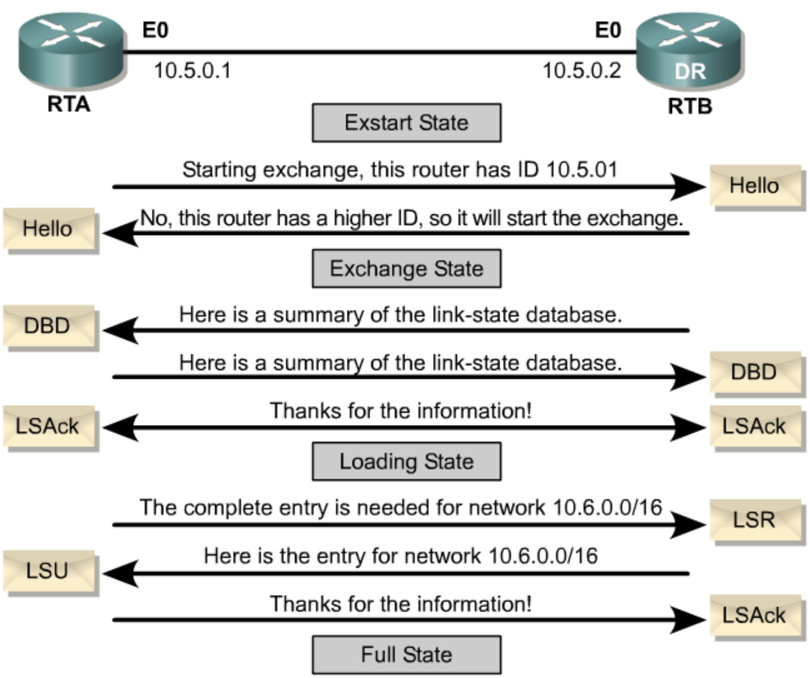
Exchange State – the explanation

- Routers exchange one or more Type-2 DBDs (Database Description) packets, which is a **summary of the link-state database**
  - send LSAcks to verify
- Routers compare these DBDs with information in its own database.
- When a DBD packet is received the router looks through the LSA (Link State Advertisement) headers and identifies LSAs that are not in the router's LSDB or are a different version from its LSDB version (older or newer).
- If the LSA is not in its LSDB or the LSA is a more recent version, the router adds an entry to its **Link State Request list**.
- This process ends when both routers stop have sent and received acknowledgements for all their DBD packets – that is they have successfully sent all their DBD packets to each other.

Exchange State – the explanation

Exchange State

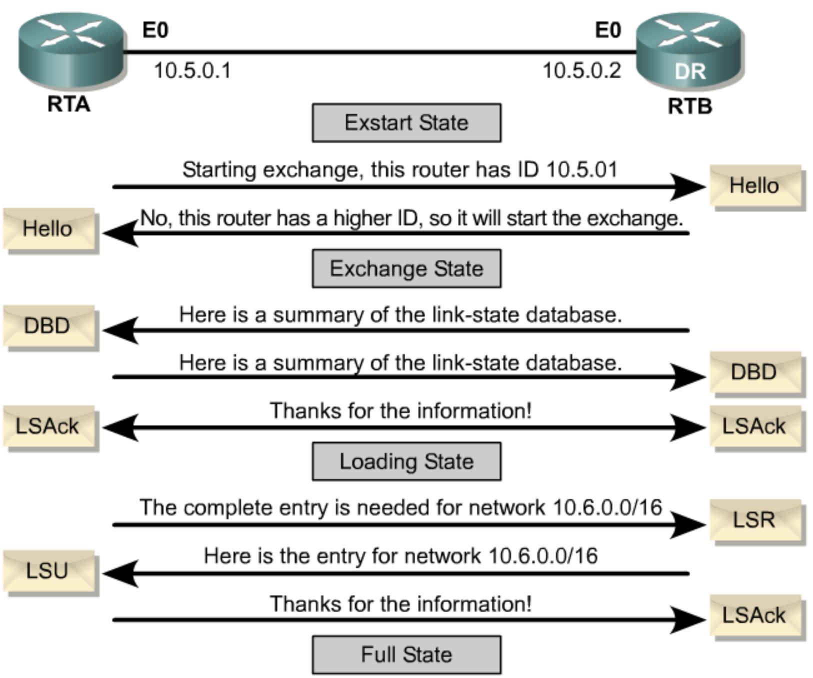
- If a router has entries in its **Link State Request list**, meaning that it needs additional information from the other router for routes that are not in its LSDB or has more recent versions, then it enters the **loading state.**
- If there are no entries in its **Link State Request list**, than the router's interface can transition directly to **full state**.
- Complete routing information is exchanged in the **loading state**, discussed next.



Loading State - the explanation

Loading State

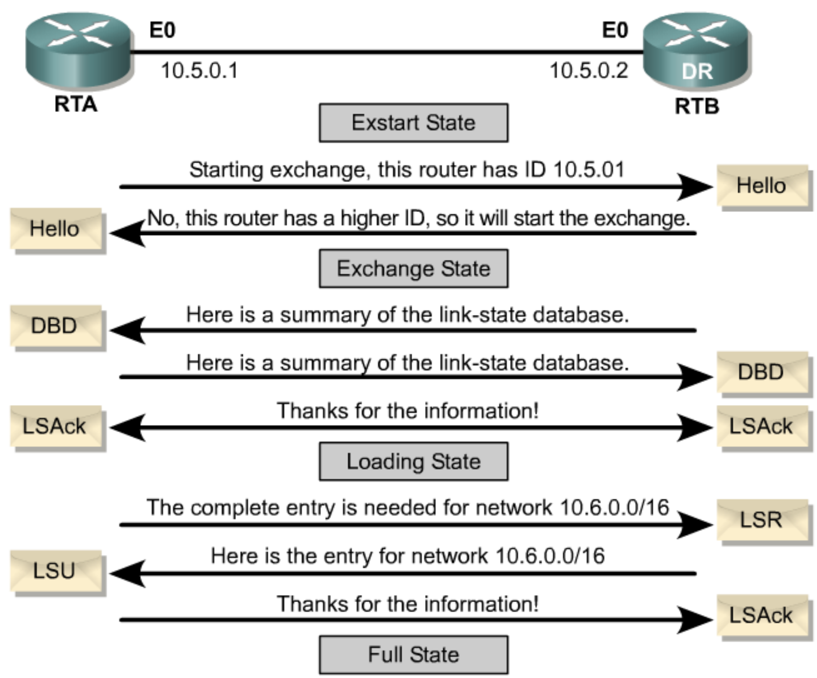
- If a router has entries in its **Link State Request list**, meaning that it needs additional information from the other router for routes that are not in its LSDB or has more recent versions, then it enters the **loading state.**
- The router needing additional information sends **LSR (Link State Request) packets** using LSA information from its LSR list.



Loading State - the explanation

Loading State

- The other routers replies by sending the requested LSAs in the **Link State Update (LSU) packet.**
- The receiving router sends **LSAck** to acknowledge receipt.
- When all LSAs on the neighbors **Link State Request list** have been received, the "neighbor FSM" transitions this interface to **Full state.**



Link State Requests (LSR)

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       3       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          LS type                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Link State ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
```

LSR

(LSRs)

Link State Advertisement (Update)

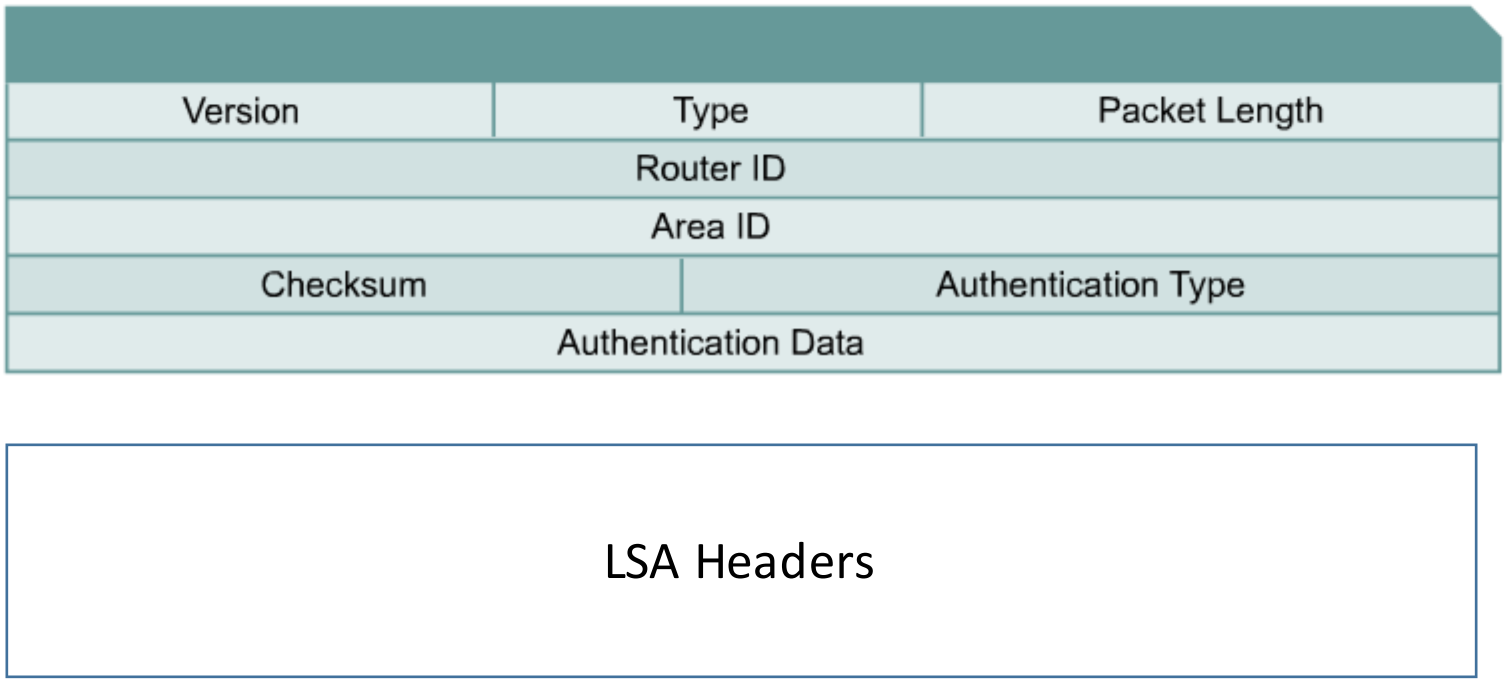
```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Version #   |       4       |         Packet length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          Router ID                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Area ID                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Checksum            |             AuType            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       Authentication                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       Authentication                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                            # LSAs                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +-                                                            +-+
 |                             LSAs                              |
 +-                                                            +-+
 |                              ...                              |
```

LSAs included in this msg

LSAs: Types 1, 2, 3, 4, or 5

LSAs

OSPF Link State ACK Packet Format



OSPF Link State Advertisement Header Format

<table>
<tr><td>LS Age</td><td>Options</td><td>Type</td></tr>
<tr><td colspan="3">Link State ID</td></tr>
<tr><td colspan="3">Advertising Router</td></tr>
<tr><td colspan="3">LS Sequence Number</td></tr>
<tr><td>LS Checksum</td><td colspan="2">Length</td></tr>
</table>

LS Age -- #seconds elapsed since the LSA was created

OSPF LS Types

Value	Link Type	Description
1	Router –LSA	Link to a Router
2	Network-LSA	Link to a network
3	Summary –LSA	Summary information about a network
4	ASBR Summary-LSA	Summary information about a link to an ASBR
5	AS-Extenal LSA	External link outside the AS
6	Multicast-OSPF-LSA	Deprecated
7	Not-so-stubby area LSA	
8	External attribute LSA for BGP	
9	link-local "opaque" LSA	Carry application-specification info such as traffic engineering or MPLS throughput the OSPF domain
10	area-local "opaque" LSA	
11	AS "opaque" LSA	

OSPF LS Types -- Continue

Value	LSA Type	Originated By	Advertised To
1	Router LSA	Every Router	Within the area originated
2	Network LSA	Designated Router	Within the area originated
3	Network Summary LSA	ABR	Flooded into a singe area
4	ASBR Summary LSA	ABR	Flooded into a single area
5	AS External LSA	ASBR	All non-stub areas
7	NSSA External LSA	ASBR	Within the NSSA originated

OSPF Router LSA

<table>
<tr><td colspan="2">LS Age</td><td>Options</td><td>Type</td></tr>
<tr><td colspan="4">Link State ID</td></tr>
<tr><td colspan="4">Advertising Router</td></tr>
<tr><td colspan="4">LS Sequence Number</td></tr>
<tr><td colspan="2">LS Checksum</td><td colspan="2">Length</td></tr>
<tr><td>00000VEB</td><td>0x00</td><td colspan="2">#Links</td></tr>
<tr><td colspan="4">Link ID</td></tr>
<tr><td colspan="4">Link Data</td></tr>
<tr><td>Link Type</td><td>#TOS</td><td colspan="2">Metric</td></tr>
<tr><td>TOS</td><td>0x00</td><td colspan="2">TOS Metric</td></tr>
<tr><td colspan="4">.....</td></tr>
<tr><td colspan="4">Link ID</td></tr>
<tr><td colspan="4">Link Data</td></tr>
</table>

Originating router’s router ID

Cost of the ink

V: Virtual Link Endpoint

E: External, originating router is ASBR

B: Border, originating router is ABR

Router LSA Link Types

Link Type	Connection	Link ID	Link Data
1	Point-to-point	Neighboring Router’s Router ID	IP addr of the originating router’s intf to the network
2	Connection to a transit network	IP addr of the DR’s intf	IP addr of the originating router’s intf to the network
3	Connection to a stub network	IP network or subnet addr	Network’s IP addr or subnet mask
4	Virtual Link	Neighboring Router’s Router ID	Ifindex value for the originating router’s intf

OSPF Network LSA

LS Age		Options	Type	
Link State ID				IP addr of the DR’s intf to the network
Advertising Router				
LS Sequence Number				
LS Checksum		Length		
00000VEB	0x00	#Links		
Network Mask				
Attached Router				Router IDs of all routers on the network
.....				
Attached Router				

Advertise the multi-access network and al routers attached to the network

OSPF Network and ASBR Summary LSA

LS Age		Options	Type
Link State ID			
Advertising Router			
LS Sequence Number			
LS Checksum		Length	
00000VEB	0x00	#Links	
Network Mask			
0x00	Metric		
TOS	0x00	TOS Metric	
.....			
0x00	Metric		
TOS	0x00	TOS Metric	

- IP addr of the network (Type 3)
- Router ID of the ASBR (Type 4)

Cost of the route to this destination

Network Summary LSA (Type 3) – Advertise networks external to an area
ASBR Summary LSA (Type 4) – Advertise ASBRs external to an area

OSPF AS External LSA

<table>
<tr><td colspan="2">LS Age</td><td>Options</td><td>Type</td><td></td></tr>
<tr><td colspan="4">Link State ID</td><td>• IP addr of the destination</td></tr>
<tr><td colspan="4">Advertising Router</td><td></td></tr>
<tr><td colspan="4">LS Sequence Number</td><td></td></tr>
<tr><td colspan="2">LS Checksum</td><td colspan="2">Length</td><td></td></tr>
<tr><td>00000VEB</td><td>0x00</td><td colspan="2">#Links</td><td></td></tr>
<tr><td colspan="4">Network Mask</td><td></td></tr>
<tr><td>E0000000</td><td colspan="3">Metric</td><td>Cost of the route to this destination</td></tr>
<tr><td colspan="4">Forwarding Address</td><td></td></tr>
<tr><td colspan="4">External Route Tag</td><td></td></tr>
<tr><td colspan="4">.....</td><td></td></tr>
</table>

AS External LSA (Type 5) – Advertise destinations external to the Autonomous System

OSPF NSSA External LSA (Type 7)

<table>
<tr><td colspan="2">LS Age</td><td>Options</td><td>Type</td></tr>
<tr><td colspan="4">Link State ID</td></tr>
<tr><td colspan="4">Advertising Router</td></tr>
<tr><td colspan="4">LS Sequence Number</td></tr>
<tr><td colspan="2">LS Checksum</td><td colspan="2">Length</td></tr>
<tr><td>00000VEB</td><td>0x00</td><td colspan="2">#Links</td></tr>
<tr><td colspan="4">Network Mask</td></tr>
<tr><td>E0000000</td><td colspan="3">Metric</td></tr>
<tr><td colspan="4">Forwarding Address</td></tr>
<tr><td colspan="4">External Route Tag</td></tr>
<tr><td colspan="4">.....</td></tr>
</table>

- IP addr of the destination

Cost of the route to this destination

- Flooded only within the NSSA in which it was originated
- Forwarding Address –
    - next hop address for internal route
    - NSSA ASBR's Router ID for non-internal route

Full State - the explanation

Full State

- **Full state** - after all LSRs have been updated.
- At this point the routers should have identical LSDBs (link-state databases).

Flooding LSAs

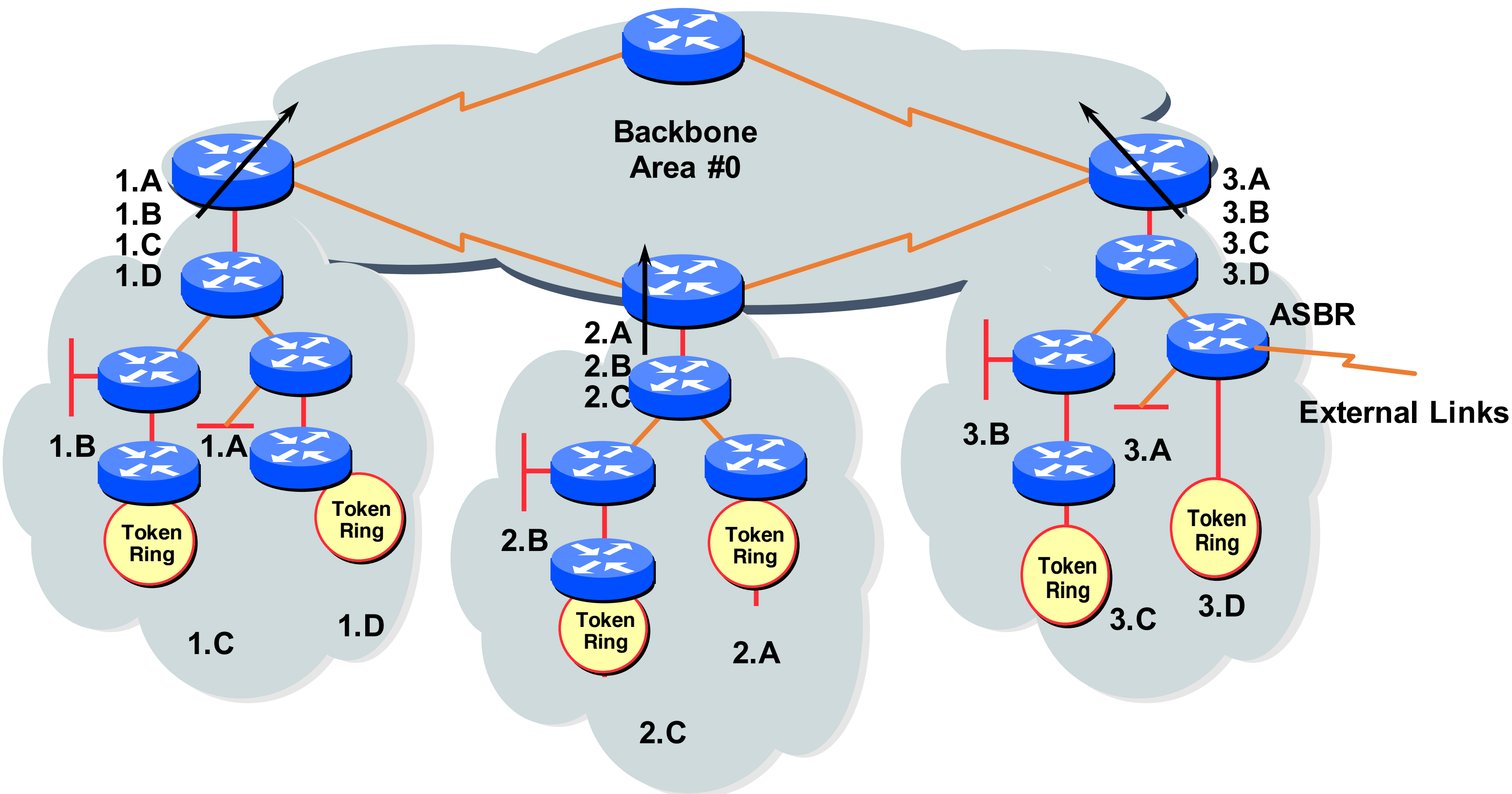
- Once this interface transitions to or from **Full state** the router **originates** a new version of a **Router LSA** (coming) and **floods it to its neighbors**, distributing the new topological information – out all OSPF enabled interfaces.
- Broadcast networks:
  - DR: If the LSA was received on this interface, send it out this interface so DROthers receive it (224.0.0.5 - all OSPF routers)
  - BDR/DROther: If the LSA was received on this interface, do <u>not</u> send out this interface (received from DR).

Calculating Routing Table

- The router still must calculate its routing table – ***Next!***

Couple of notes on link state flooding…

- OSPF is a link state routing protocol and **does not send periodic updates** like RIP.
- OSPF only **floods** link state state advertisements when there is a **change in topology** (this includes when a routers are first booted).
- OSPF uses **hop-by-hop flooding** of LSAs; an LSA received on one interface are flooded out other OSPF enabled interfaces.
- If a link state entry in the LSDB (Link State DataBase) reaches an age of **60 minutes (MaxAge)** without being updated, it is removed and SPF is recalculated.
- Every **30 minutes (LSRefreshTime),** OSPF routers flood only their link states to all other routers (in the area).
  - This is known as a "**paranoid update**"
  - These do not trigger SPF recalculations.
- Special note: When a **link goes down** and a router wants to send a LSA to tell other routers to remove this link state, it sends this link state with a **value of 60 minutes (MAXAGE).**


Backbone
Area #0
1.A
1.B
1.C
1.D
1.B
1.A
Token Ring
Token Ring
1.C
1.D
2.A
2.B
2.C
2.B
Token Ring
Token Ring
2.A
2.C
3.A
3.B
3.C
3.D
ASBR
External Links
3.B
3.A
Token Ring
Token Ring
3.C
3.D