

Chapter 4: Manipulating Routing Updates



CCNP ROUTE: Implementing IP Routing

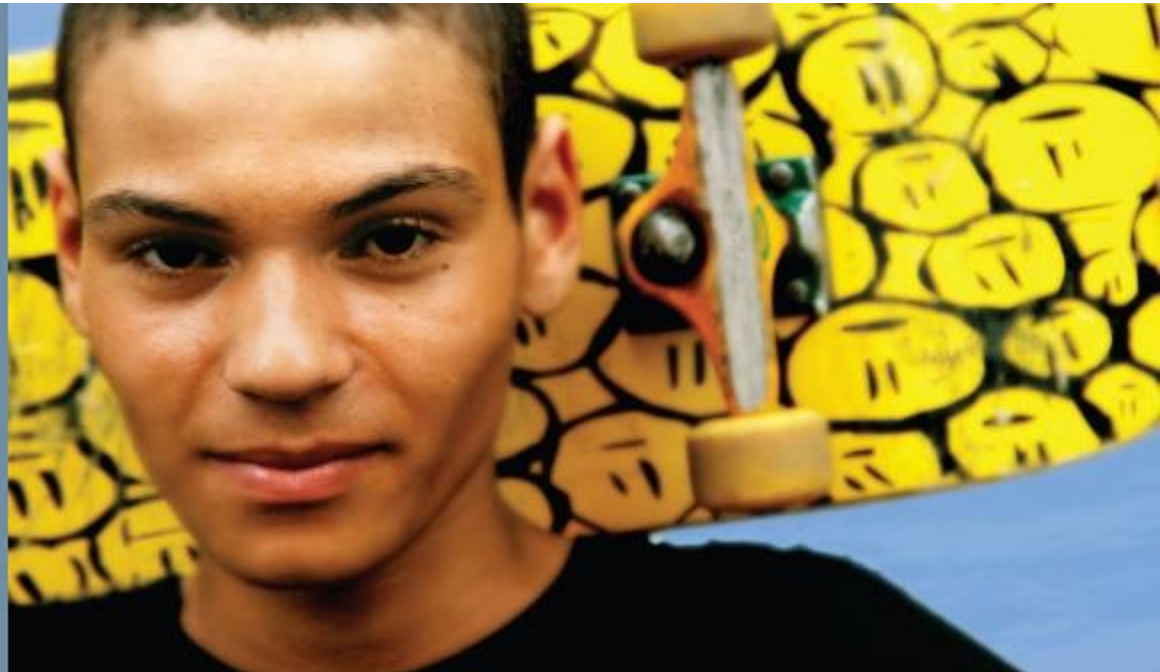
Cisco | Networking Academy®
Mind Wide Open™



Chapter 4 Objectives

- This chapter covers the following topics:
- Using Multiple IP Routing Protocols on a Network
- Implementing Route Redistribution
- Controlling Routing Update Traffic

Using Multiple IP Routing Protocols on a Network





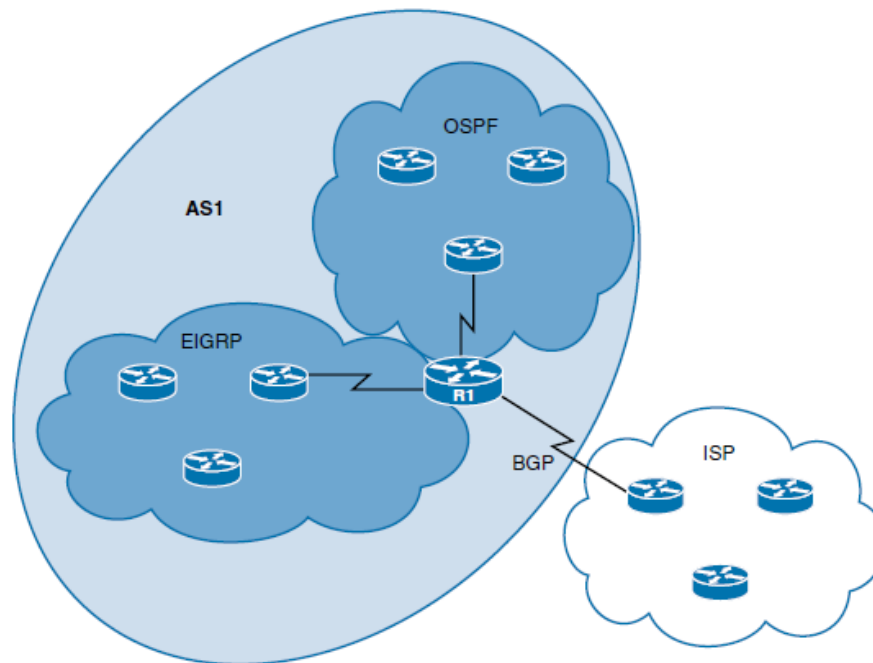
Using Multiple IP Routing Protocols on a Network

- Describe the need for using more than one protocol in a network
- Describe how routing protocols interact
- Describe solutions for operating in a multiple routing protocol environment



Using Multiple IP Routing Protocols on a Network

- Simple routing protocols work well for simple networks, but as networks grow and become more complex, it may be necessary to change the routing protocols.
- Often, the transition between routing protocols takes place gradually, so there are multiple routing protocols that are operating in the network for variable lengths of time.





Why Run Multiple Routing Protocols?

- When **migrating** from an older Interior Gateway Protocol (IGP) to a new IGP.
- The same applies to **company mergers** between companies that are each using a different routing protocol.
- In mixed-router vendor environments.
- When the use of a **new protocol is desired**, but the old routing protocol is needed for host systems
- When some departments **do not want to upgrade** their routers to support a new routing protocol.



Running Multiple Routing Protocols

- When running multiple routing protocols, a router may learn of a same route from different routing sources.
- If a router learns of a specific destination from two different routing domains, the route with the **lowest administrative distance** would get installed in routing table.



Administrative Distance

- The administrative distance is used to rate a routing protocol's **believability** (also called its **trustworthiness**). Each routing protocol is prioritized in order from most to least believable using an assigned value called the *administrative distance*.

Route Source	Default Administrative Distance
EIGRP and EIGRP for IPv6 summary route	5
External BGP	20
Internal EIGRP, EIGRP for IPv6	90
OSPFv2, OSPFv3	110
RIPv1, RIPv2, RIPv6	120
Internal BGP	200
Unreachable	255



Multiple Routing Protocols Solutions

- **Careful routing protocol design** and **traffic optimization solutions** should be implemented when supporting complex multiprotocol networks.
- These solutions include the following:
 - Summarization (Chapter 2 & 3)
 - Redistribution between routing protocols
 - Route filtering

Implementing Route Redistribution





Implementing Route Redistribution

- Describe the need for route redistribution
- Identify some considerations for route redistribution
- Describe how to configure and verify route redistribution
- Identify the different types of route redistribution



Defining Route Redistribution

- Cisco routers **allow** internetworks **using multiple routing protocols** to exchange routing information using the **route redistribution feature**.
- Route redistribution is defined as the capability of boundary routers connecting different routing domains to exchange and advertise routing information between those routing domains (autonomous systems).
- **Redistribution shares routing information** about routes that the router has learned with other routing protocols.

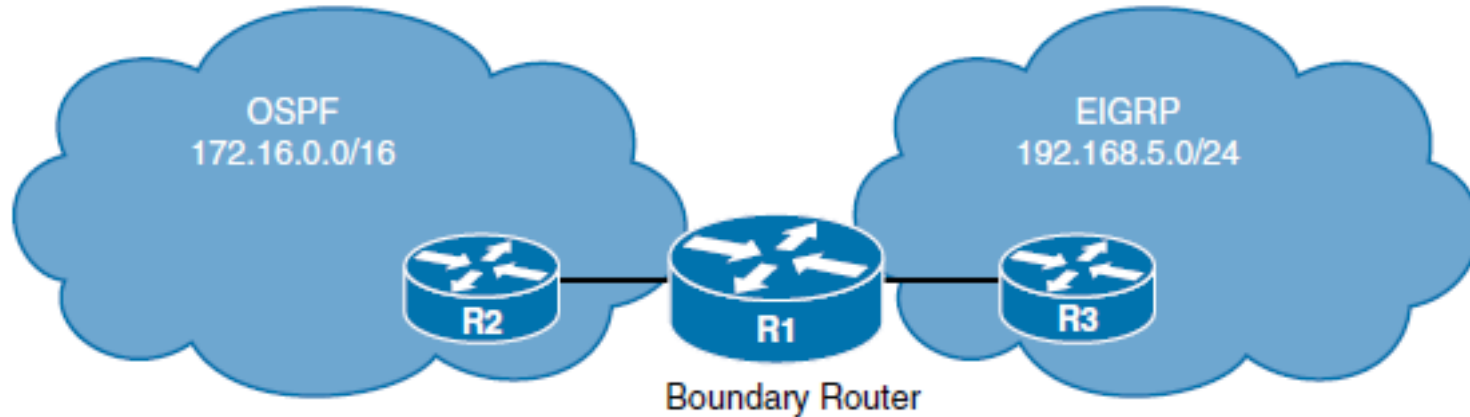


Planning to Redistribute Routes

- Network administrators must **manage redistribution carefully** because it can lead to **routing loops**, which negatively affect an internetwork.
- Different routing protocols have different requirements and capabilities, so it is important for network administrators to create a **detailed plan before making any routing protocol changes**.
- An accurate topology map of the network and an inventory of all network devices are critical for success.
- To have a scalable solution and limit the amount of routing update traffic, the redistribution process must selectively insert the routes that are learned.
- **When a router redistributes routes, it only propagates routes that are in the routing table.** Therefore, a router can redistribute *dynamically learned routes, static routes, and direct connected routes*.



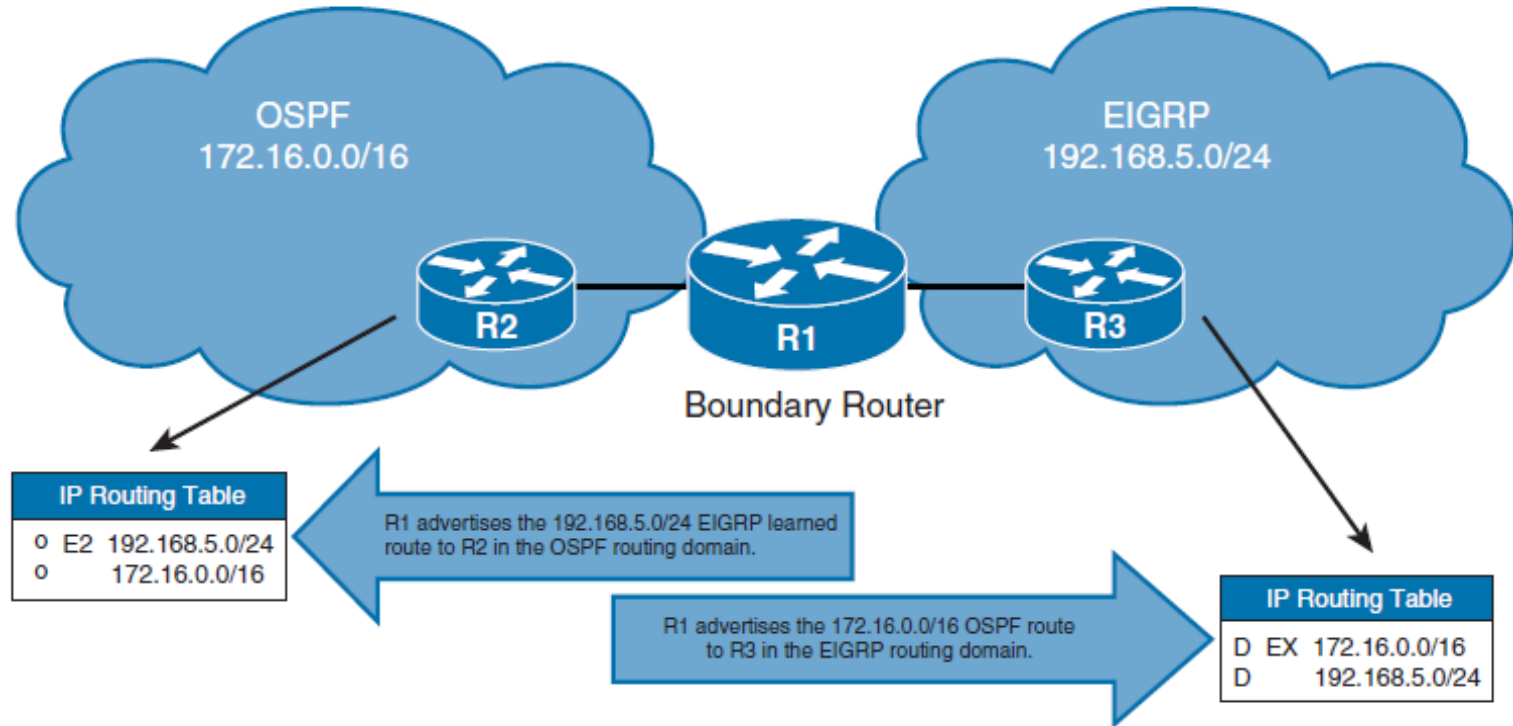
Redistributing Routes



- Redistribution is always performed *outbound*.
- This means that the router **doing redistribution does not change its own routing table**.
- Only downstream routers receiving the redistributed routes could add them to their respective routing tables.



Redistributing Routes



- Without redistribution, routers in the OSPF domain are not aware of EIGRP routes, and routers in the EIGRP domain are not aware of OSPF routes.



Seed Metrics

- When a router is redistributing, the redistributed route **must have a metric appropriate for the receiving protocol.**
- The seed or **default metric** is defined during redistribution configuration. After the seed metric for a redistributed route is established, the metric increments normally within the autonomous system.

The seed metric can be configured using either of the following:

- The **default-metric** router configuration command, which establishes the seed metric for all redistributed routes. **The default metric specified applies to all protocols being redistributed into this protocol.**
- The **redistribute** router configuration command using either the **metric** option or a route map. Using the **metric** parameter in the **redistribute** command, set a specific metric for the protocol being redistributed.
- A metric configured in a **redistribute** command overrides the value in the **default-metric** command for that one protocol.



Default Seed Metrics

Default seed metric value for redistributed routes for each IP routing protocol is as follows:

- Routes redistributed **into EIGRP** and **RIP** are assigned a metric of **infinity**. This informs the router that the route is **unreachable** and should **not be advertised**. Therefore, a seed metric *must* be specified. **Exceptions to this rule are redistributed *connected* or *static routes* and routes that are being redistributed between two EIGRP autonomous systems.**
- Routes redistributed **into OSPF** are assigned a default type 2 (E2) **metric of 20**. However, redistributed BGP routes are assigned a default type 2 **metric of 1**.
- Routes redistributed into BGP **maintain their IGP routing** metrics.
- Routes redistributed into Intermediate System-to-Intermediate System (**IS-IS**) Protocol are assigned a **default metric of 0**. But unlike RIP or EIGRP, a seed metric of 0 is not treated as unreachable by IS-IS.

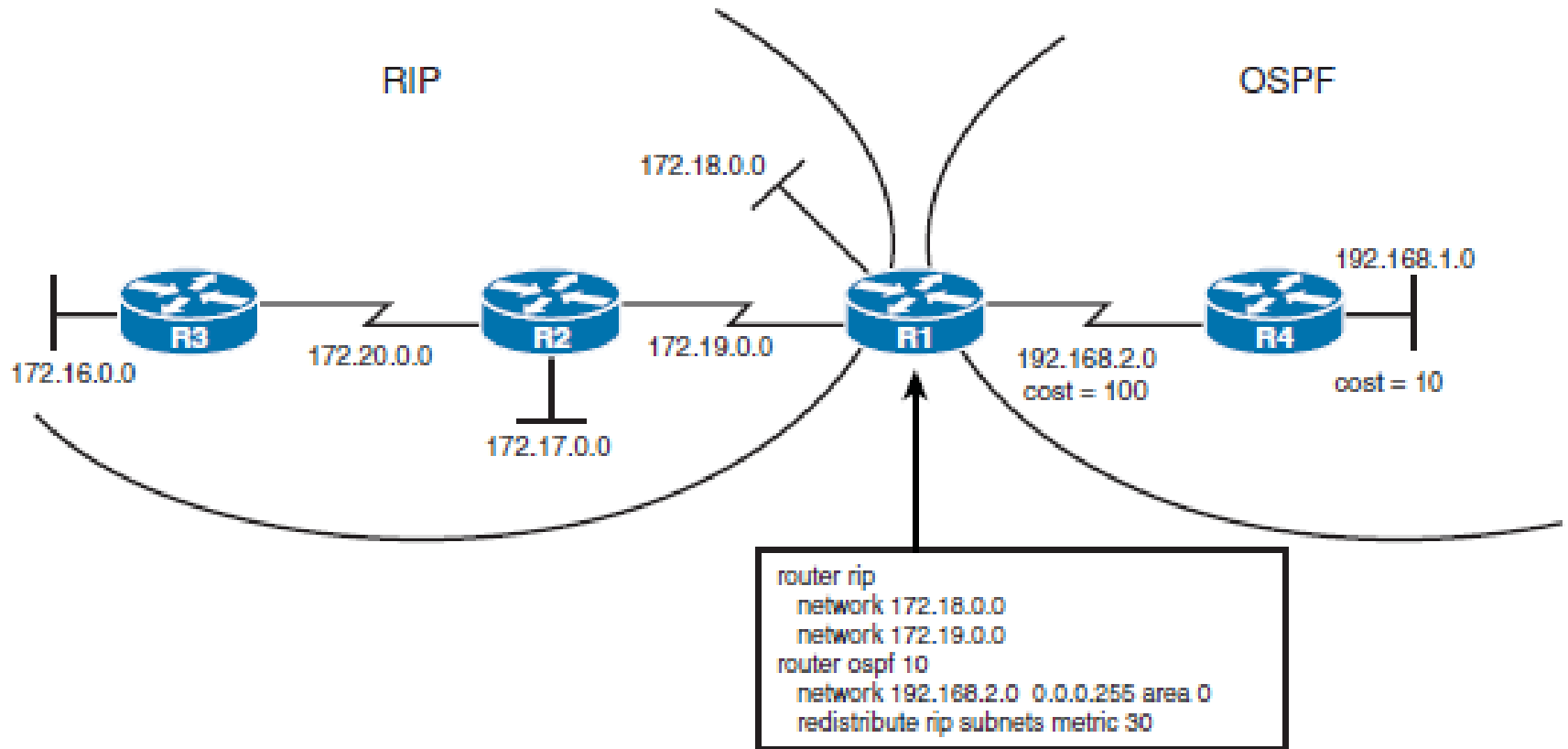


Default Seed Metrics

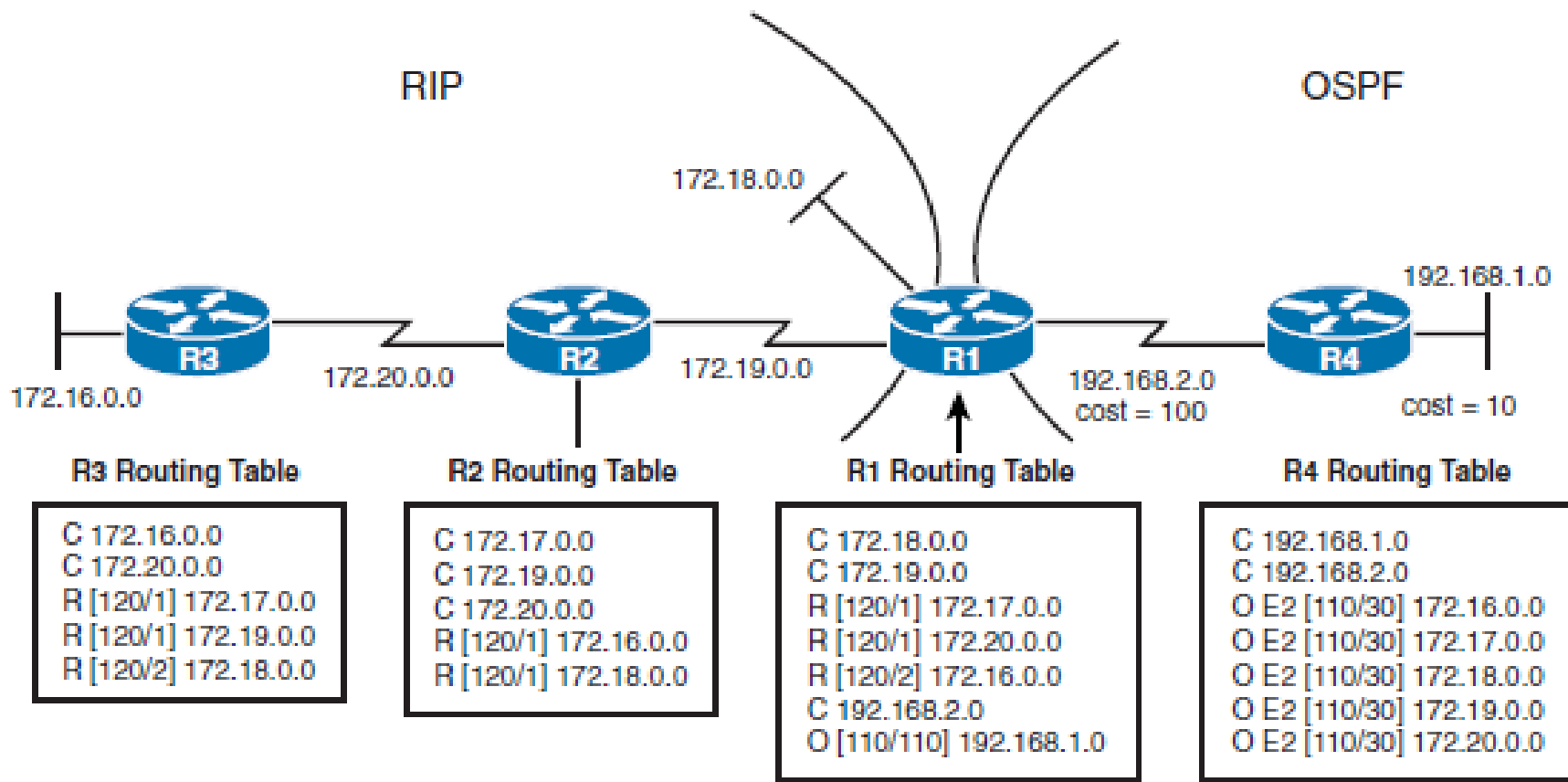
Protocol That Route Is Redistributed Into	Default Seed Metric
RIP	0, which is interpreted as infinity and unreachable.
EIGRP	0, which is interpreted as infinity and unreachable.
OSPF	20. The exception is for BGP routes, which have a default seed metric of 1. (All default to type E2.)
BGP	BGP metric is set to IGP metric value.



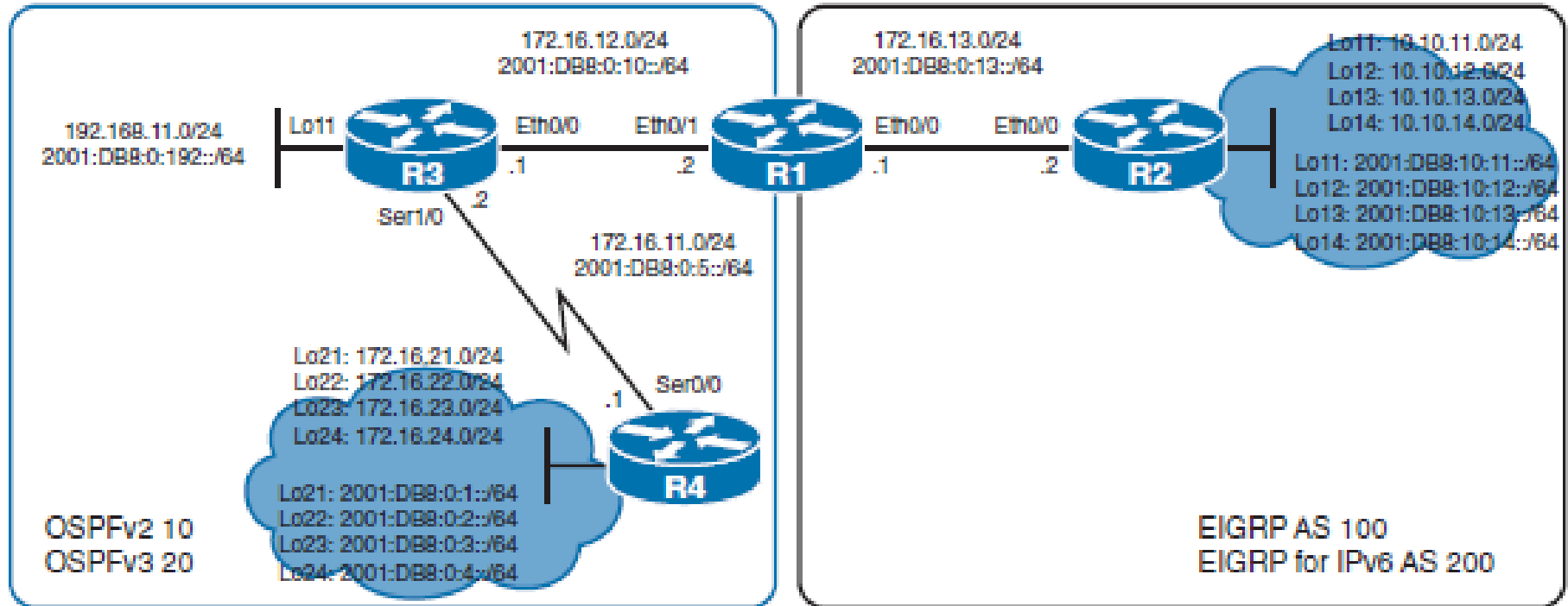
Redistributing RIP Routes into OSPF



Redistributing RIP Routes into OSPF



Configuring and Verifying Basic Redistribution in IPv4 and IPv6





Redistributing OSPFv2 Routes into the EIGRP Routing Domain

- Router(config-router)# **redistribute** *protocol process-id* [*metric bandwidth-metric delay-metric reliability-metric effective-bandwidth-metric mtu-bytes*] [**route-map map-tag**]

Parameter	Description
<i>protocol</i>	The source protocol from which routes are redistributed. Common keywords include connected , static , rip , ospf , and bgp .
<i>process-id</i>	For BGP or EIGRP, this value is an autonomous system number. For OSPF, this value is an OSPF process ID.
metric	(Optional) Specifies the metric for redistributed routes.
<i>bandwidth-metric</i>	Maximum bandwidth of the route, in kilobits per second (Kbps). The range is from 1 to 4,294,967,295.
<i>delay-metric</i>	EIGRP route delay metric, in microseconds. The range is from 1 to 4,294,967,295.
<i>reliability-metric</i>	EIGRP reliability metric. The range is from 0 to 255. An EIGRP metric of 255 signifies 100 percent reliability.
<i>effective-bandwidth-metric</i>	Effective bandwidth of the route. The range is from 1 to 255. Effective bandwidth of 255 denotes 100 percent load.
route-map	(Optional) Route map that should be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol. If not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes will be imported.
<i>map-tag</i>	Route map name.



Redistributing OSPFv2 Routes into the EIGRP Routing Domain

```
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 10 metric 1500 100 255 1 1500
```

```
R1(config)# router eigrp 100
R1(config-router)# default-metric 1500 100 255 1 1500
R1(config-router)# redistribute ospf 10
```



R2# show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
 ia - IS-IS inter area, * - candidate default, U - per-user static route
 o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
 + - replicated route, % - next hop override

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks

C 10.10.11.0/24 is directly connected, Loopback11
 L 10.10.11.1/32 is directly connected, Loopback11
 C 10.10.12.0/24 is directly connected, Loopback12
 L 10.10.12.1/32 is directly connected, Loopback12
 C 10.10.13.0/24 is directly connected, Loopback13
 L 10.10.13.1/32 is directly connected, Loopback13
 C 10.10.14.0/24 is directly connected, Loopback14
 L 10.10.14.1/32 is directly connected, Loopback14

172.16.0.0/16 is variably subnetted, 8 subnets, 3 masks

D EX 172.16.11.0/30 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0
 D EX 172.16.12.0/24 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0
 C 172.16.13.0/24 is directly connected, Ethernet0/0
 L 172.16.13.2/32 is directly connected, Ethernet0/0
 D EX 172.16.21.1/32 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0
 D EX 172.16.22.1/32 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0
 D EX 172.16.23.1/32 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0
 D EX 172.16.24.1/32 [170/1757696] via 172.16.13.1, 00:00:05, Ethernet0/0

192.168.11.0/32 is subnetted, 1 subnets



Redistributing OSPFv3 Routes into the EIGRP for IPv6 Routing Domain

```
R1(config)# ipv6 router eigrp 200
R1(config-router)# redistribute ospf 20 metric 1500 100 255 1 1500
```

```
R2# show ipv6 route eigrp
IPv6 Routing Table - default - 17 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, 1 - LISP
EX 2001:DB8:0:1::1/128 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
EX 2001:DB8:0:2::1/128 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
EX 2001:DB8:0:3::1/128 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
EX 2001:DB8:0:4::1/128 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
EX 2001:DB8:0:5::/64 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
EX 2001:DB8:0:192::1/128 [170/1757696]
    via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
R2#
```



Redistributing Connected Routes into EIGRP for IPv6

```
R1(config)# ipv6 router eigrp 200
```

```
R1(config-rtr)# redistribute ospf 20 metric 1500 100 255 1 1500 include-connected
```

```
R2# show ipv6 route eigrp
```

```
IPv6 Routing Table - default - 18 entries
```

```
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
```

```
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
```

```
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
```

```
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
```

```
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
```

```
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
```

```
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, 1 - LISP
```

```
EX 2001:DB8:0:1::1/128 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:2::1/128 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:3::1/128 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:4::1/128 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:5::/64 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:10::/64 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
EX 2001:DB8:0:192::1/128 [170/1757696]
   via FE80::A8BB:CCFF:FE01:6C00, Ethernet0/0
```

```
R1(config-router)# redistribute ospf 20 metric 1500 100 255 1 1500 include-connected
```



Redistributing EIGRP Routes into the OSPFv2 Routing Domain

- Router(config-router)# **redistribute** *protocol process-id* [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-tag*] [**subnets**]

Parameter	Description
<i>protocol</i>	The source protocol from which routes are redistributed. Common keywords include <code>connected</code> , <code>static</code> , <code>rip</code> , <code>eigrp</code> , and <code>bgp</code> .
<i>process-id</i>	For BGP or EIGRP, this value is an autonomous system number. For OSPF, this value is an OSPF process ID.
metric <i>metric-value</i>	(Optional) This parameter is used to specify the metric for the redistributed route. If it is not explicitly specified, then the redistributed routes are assigned a metric of 20 by default.
metric-type <i>type-value</i>	(Optional) This OSPF parameter specifies the external link type. This can be 1 for type 1 external routes, or 2 for type 2 external routes. The default is 2.
route-map	(Optional) Route map that should be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol. If not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes will be imported.
<i>map-tag</i>	Route map name.
subnets	(Optional) For redistributing routes into OSPF, the scope of redistribution for the specified protocol. By default, no subnets are defined.



Redistributing EIGRP Routes into OSPF

```
R1(config)# router ospf 10
R1(config-router)# redistribute eigrp 100 subnets
```

```
R3# show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 4 subnets
O E2   10.10.11.0 [110/20] via 172.16.12.2, 00:02:29, Ethernet0/0
O E2   10.10.12.0 [110/20] via 172.16.12.2, 00:02:29, Ethernet0/0
O E2   10.10.13.0 [110/20] via 172.16.12.2, 00:02:29, Ethernet0/0
O E2   10.10.14.0 [110/20] via 172.16.12.2, 00:02:29, Ethernet0/0
172.16.0.0/16 is variably subnetted, 9 subnets, 3 masks
O E2   172.16.13.0/24 [110/20] via 172.16.12.2, 00:02:29, Ethernet0/0
O      172.16.21.1/32 [110/65] via 172.16.11.1, 2d20h, Serial1/0
O      172.16.22.1/32 [110/65] via 172.16.11.1, 2d20h, Serial1/0
O      172.16.23.1/32 [110/65] via 172.16.11.1, 2d20h, Serial1/0
O      172.16.24.1/32 [110/65] via 172.16.11.1, 2d20h, Serial1/0
R3#
```



OSPF Metric-Type

The following external packet types can be configured:

- **E1:** Type O E1 external routes **calculate the cost by adding the external cost to the internal cost** of each link that the packet crosses. Use this type when there are multiple ASBRs advertising an external route to the same autonomous system to avoid suboptimal routing.
- **E2 (default):** The external cost of O E2 routes is fixed and **does not change across OSPF domain**. Use this type if **only one ASBR** is advertising an external route to the autonomous system



Redistributing EIGRP Routes into OSPF as External Type 1 Routes

```
R1(config)# router ospf 10
R1(config-router)# redistribute eigrp 100 metric-type 1 subnets
```

```
R3# show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 4 subnets
O E1   10.10.11.0 [110/30] via 172.16.12.2, 00:00:02, Ethernet0/0
O E1   10.10.12.0 [110/30] via 172.16.12.2, 00:00:02, Ethernet0/0
O E1   10.10.13.0 [110/30] via 172.16.12.2, 00:00:02, Ethernet0/0
O E1   10.10.14.0 [110/30] via 172.16.12.2, 00:00:02, Ethernet0/0
    172.16.0.0/16 is variably subnetted, 9 subnets, 3 masks
O E1   172.16.13.0/24 [110/30] via 172.16.12.2, 00:00:02, Ethernet0/0
O      172.16.21.1/32 [110/65] via 172.16.11.1, 2d21h, Serial1/0
O      172.16.22.1/32 [110/65] via 172.16.11.1, 2d21h, Serial1/0
O      172.16.23.1/32 [110/65] via 172.16.11.1, 2d21h, Serial1/0
O      172.16.24.1/32 [110/65] via 172.16.11.1, 2d21h, Serial1/0

R3#
```



Redistributing EIGRP for IPv6 Routes into OSPFv3

```
R1(config)# ipv6 router ospf 20
R1(config-rtr)# redistribute eigrp 200 include-connected
```

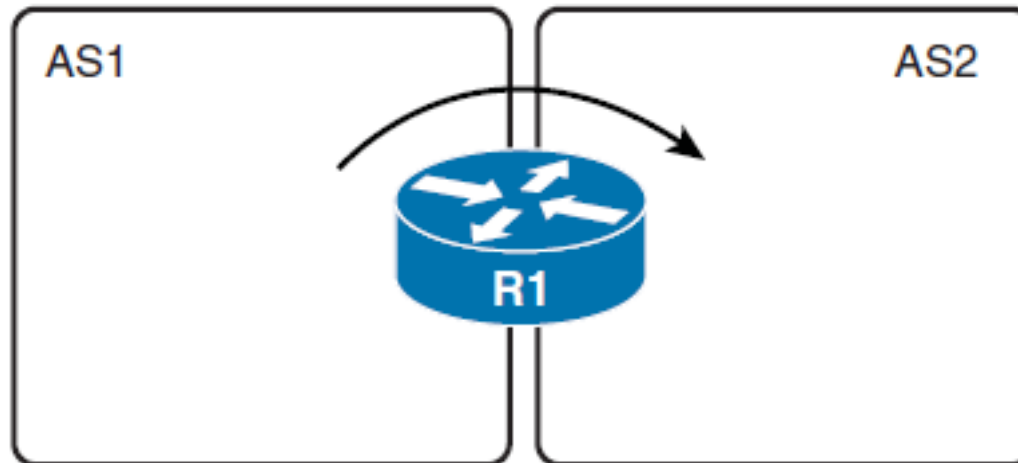
```
R4# show ipv6 route ospf
IPv6 Routing Table - default - 18 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, 1 - LISP
O   2001:DB8:0:10::/64 [110/74]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OE2 2001:DB8:0:13::/64 [110/20]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OI  2001:DB8:0:192::1/128 [110/64]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OE2 2001:DB8:10:11::/64 [110/20]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OE2 2001:DB8:10:12::/64 [110/20]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OE2 2001:DB8:10:13::/64 [110/20]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
OE2 2001:DB8:10:14::/64 [110/20]
    via FE80::A8BB:CCFF:FE01:6E00, Serial10/0
R4#
```



Types of Redistribution Techniques

One-Point Redistribution

- **One-way redistribution** - This method only redistributes the networks learned from one routing protocol into the other routing protocol.
- With this method, R1 performs one-way redistribution because it only redistributes AS1 routes into the AS2 routing domain.
- AS2 routes are not being redistributed in AS1.
- Typically, AS1 routers would require the use of a default route or one or more static routes to reach AS2 routes.

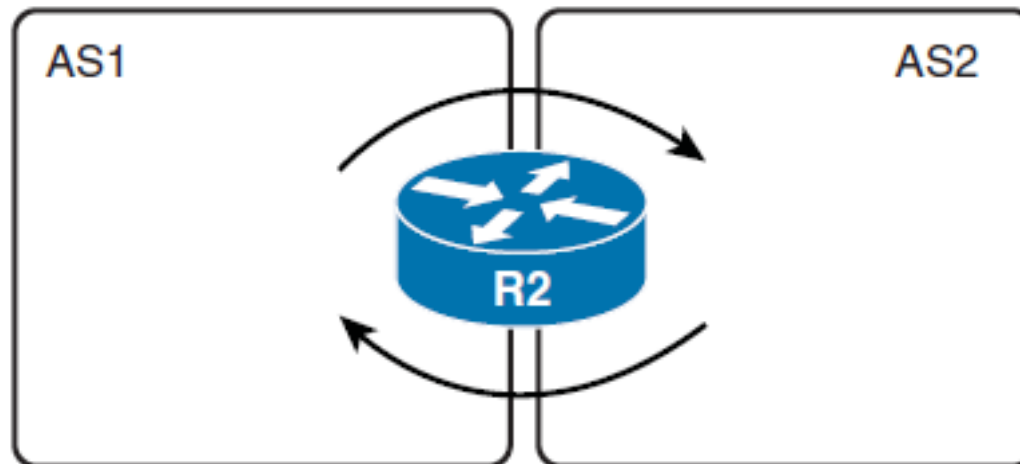




Types of Redistribution Techniques

One-Point Redistribution

- **Two-way redistribution:** This method redistributes routes between the two routing processes in both directions.
- R1 is the one-point of redistribution between AS1 and AS2.
- R1 provides two-way redistribution because it redistributes AS1 routes into AS2 and AS2 routes into AS1.

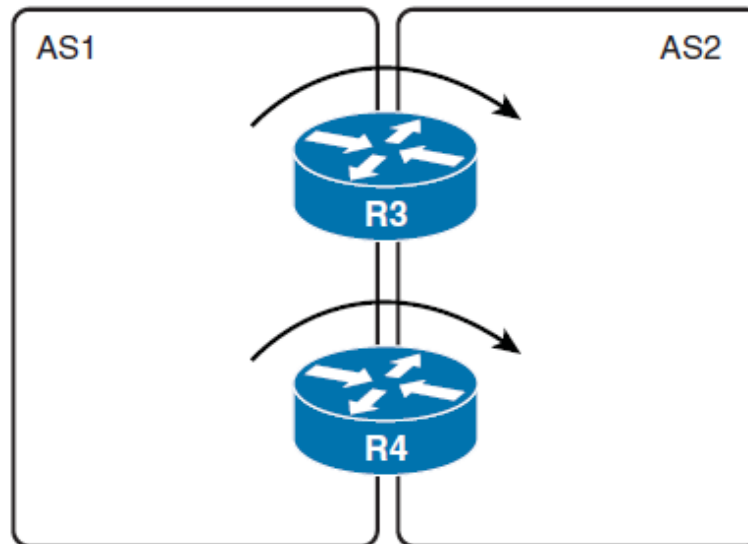




Types of Redistribution Techniques

Multipoint Redistribution

- **One-way redistribution:** This method consists of two or more boundary routers only redistributing networks learned from one routing protocol into the other routing protocol.
- The boundary routers R3 and R4 are both redistributing AS1 routes into the AS2 routing domain.
- Again, AS1 routers would require the use of a default route or one or more static routes to reach AS2 routes.

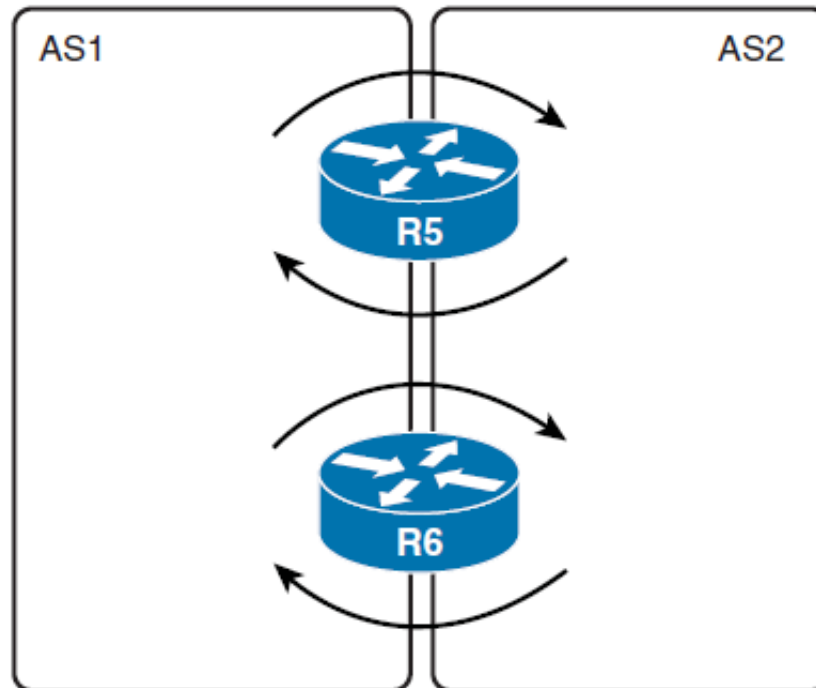




Types of Redistribution Techniques

Multipoint Redistribution

- Two-way redistribution** : Also referred to as *mutual redistribution* , this method consists of two or more boundary routers redistributing routes in both directions.
- The boundary routers R3 and R4 provide two-way redistribution because they redistribute AS1 routes into AS2 and AS2 routes into AS1.



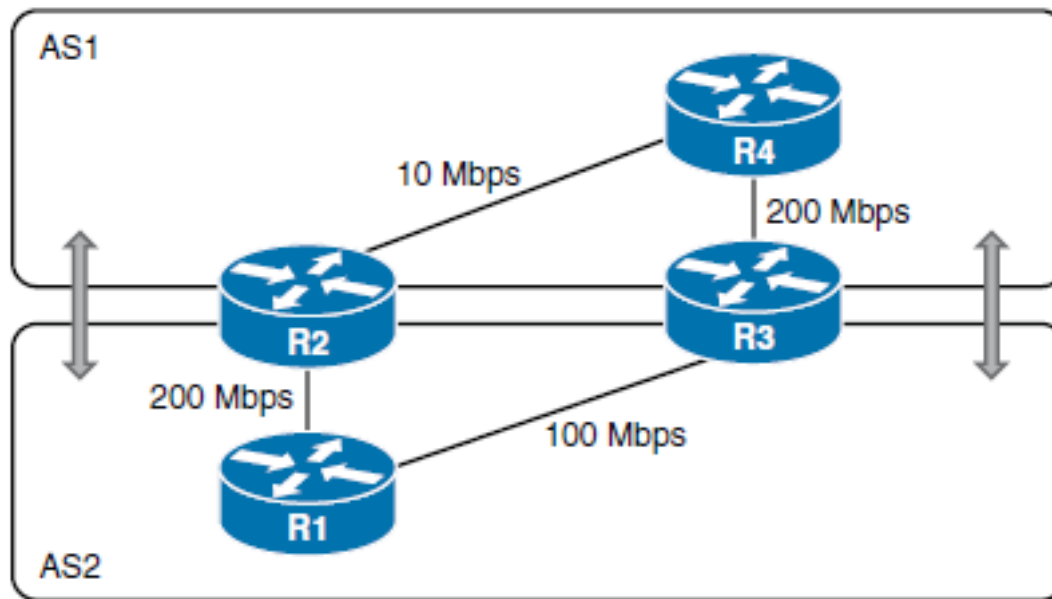


Redistribution Problems

- Generic multipoint two-way redistribution requires careful design and configuration.
- Problems that can occur during multipoint two-way redistribution include the following:
 - Suboptimal routing. (Only part of the total cost is considered in routing decisions.)
 - Self-sustained routing loops upon route loss.



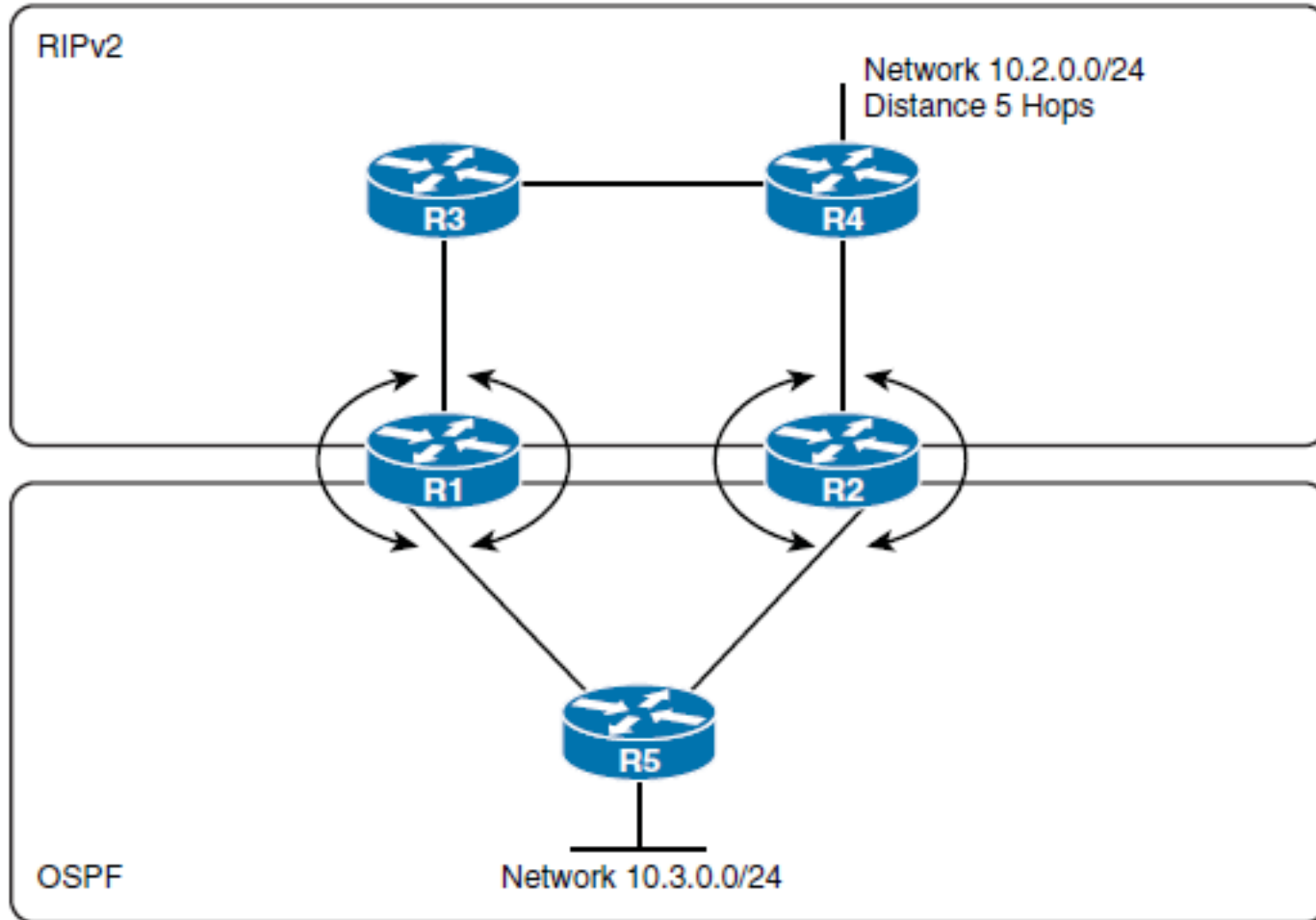
Two-Way Multipoint Redistribution Issue



- The cost of the internal links in AS1 (that is, 10 Mbps) differs from the cost of the internal links in AS2 (that is, 100 Mbps). In the figure, it is obvious that the best path between R1 and R4 is via R3, but during redistribution from AS2 to AS1, the metric is lost, and R1 is sending the packets toward R4 via R2, resulting in suboptimal routing.



Two-Way Multipoint Redistribution Issue





Two-Way Multipoint Redistribution Issue

- The 10.2.0.0/24 network is learned natively within the RIP part of the network, R4 first sees it with a hop count of 5.
- R4 then propagates this route to R3 and R2 with a hop count of 6.
- R3 propagates the route to R1 with a hop count of 7, and R2 redistributes it into OSPF.
- Now R1 has a choice to make. It has a route to the 10.2.0.0/24 network from RIP with an AD of 120 (RIP) and the same network with an AD of 110 (OSPF).
- **Because OSPF has a better (lower) AD, R1 redistributes the network back to RIP with the metric that is set in the redistribute command.**
- If the **redistribute** command is configured to assign a static metric of 3 hops (or lower); however, R3 starts preferring the path R1-R2-R4 to reach 10.2.0.0/24, because the hop count advertised by R1 is 3, and the hop count advertised by R4 is 6.
- **This results in suboptimal routing.**
- Worse, because R3 now prefers the path to R1, it will advertise this to R4 with a hop count of 4. R4 now has the choice of the route from R3 with a hop count of 4 or the true path to the 10.2.0.0/24 network with a hop count of 5.
- R4 will select the path to R3 and advertise this to R2. **There is now a routing loop (R4, R2, R1, R3, and R4).** Packets destined for the 10.2.0.0/24 network that enter this loop will bounce around the loop and never reach the destination. Network **10.2.0.0/24 becomes unreachable.**



Preventing Routing Loops in a Redistribution Environment

- Redistribute routes **in only one direction, on only one boundary router** within the network.
- If redistribution must be done in both directions or on multiple boundary routers, the redistribution **should be tuned**.
- To prevent routing loops in a multipoint redistribution scenario:
 - Only redistribute **internal routes** from one autonomous system to another (and vice versa).
 - Tag routes in redistribution points and **filter based on these tags** when configuring redistribution in the other direction.
 - **Propagate metrics from one autonomous system to another** autonomous system properly. (Even though this is not sufficient to prevent loops.)
 - Use default routes to avoid having to do two-way redistribution.



Verifying Redistribution Operation

- Know your network topology, particularly where redundant routes exist.
- Examine the topology table of each configured routing protocol to ensure that all appropriate prefixes are being learned.
- Perform a trace using the **traceroute** [*ip-address*] EXEC command on some of the routes that go across the autonomous systems to verify that the shortest path is being used for routing.
- If you encounter routing problems, use the **traceroute** and **debug** commands to observe the routing update traffic on the boundary routers and on the internal routers.

Controlling Routing Update Traffic



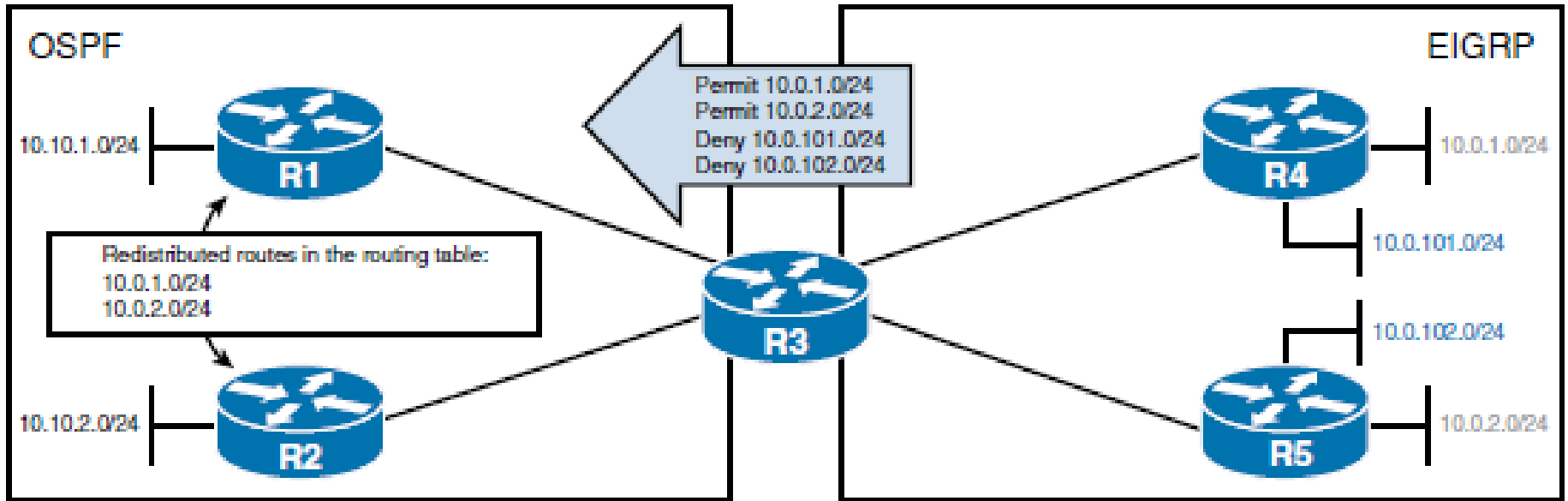


Controlling Routing Update Traffic

- Describe the general mechanics and need for **route filtering**
- Identify how to use and configure **distribute lists**
- Identify how to use and configure **prefix lists**
- Identify how to use and configure **route maps**
- Describe how to modify administrative distance



Why Filter Routes?





Route Filtering Methods

■ Distribute lists

- A distribute list **allows** an access control lists (**ACLs**) to be applied to control routing updates.

■ Prefix lists

- A prefix list is an **alternative to ACLs** designed to filter routes. **It can be used with distribute lists, route maps,** and other commands.

■ Route maps

- Route maps **are complex access lists that allow conditions** to be tested against a packet or route, and then **actions taken to modify attributes** of the packet or route.



Using Distribute Lists

- A distribute list allows an ACL to be applied to routing updates.
- Classic ACLs do not affect traffic that is originated by the router, so applying one to an interface has no effect on the outgoing routing advertisements.
- When you link an ACL to a distribute list, routing updates can be controlled no matter what their source is.
- ACLs are configured in the global configuration mode and are then associated with a distribute list under the routing protocol.
- The ACL should permit the networks that should be advertised or redistributed and deny the networks that should be filtered.



Using Distribute Lists

- The router then applies the ACL to the routing updates for that protocol. Options in the **distribute-list** command allow updates **to be filtered based on** three factors:
 - Incoming interface
 - Outgoing interface
 - Redistribution from another routing protocol
- Using a distribute list gives the administrator great flexibility in determining just **which routes will be permitted and which will be denied.**



Configuring Distribute Lists [out]

- distribute-list** [*access-list-number* | *name*] **out** [*interface-type* *interface-number* | *routing process* | *autonomous-system-number*] command.

Parameter	Description
<i>access-list-number</i> <i>name</i>	Specifies the standard access list number or name
out	Applies the access list to outgoing routing updates
<i>interface-type</i> <i>interface-number</i>	(Optional) Specifies the name of the interface out of which updates are filtered
<i>routing process</i> <i>autonomous-system-number</i>]	(Optional) Used when redistribution from another routing process or autonomous system number has been specified.

- The **distribute-list out** command filters updates **going out of the interface** or **routing protocol** specified in the command, *into* the routing process under which it is configured.



Configuring Distribute Lists [in]

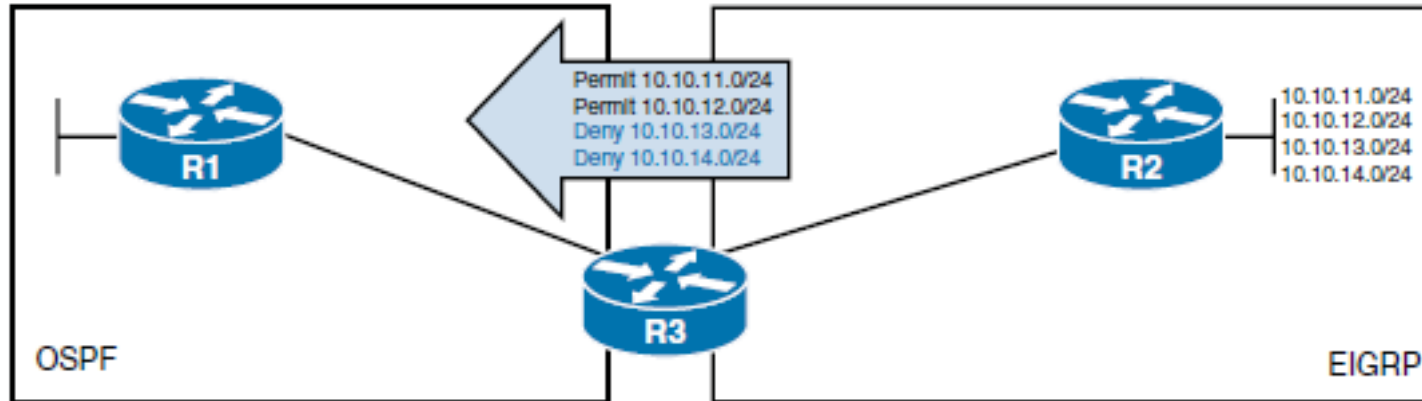
- distribute-list** [*access-list-number* | *name*] **in** [*interface-type interface-number*]

Parameter	Description
<i>access-list-number</i> <i>name</i>	Specifies the standard access list number or name
in	Applies the access list to incoming routing updates
<i>interface-type</i> <i>interface-number</i>	(Optional) Specifies the interface type and number from which updates are filtered

- The **distribute-list in** command filters updates going *into* the interface specified in the command, *into* the routing process under which it is configured.



Distribute List and ACL Example [out]

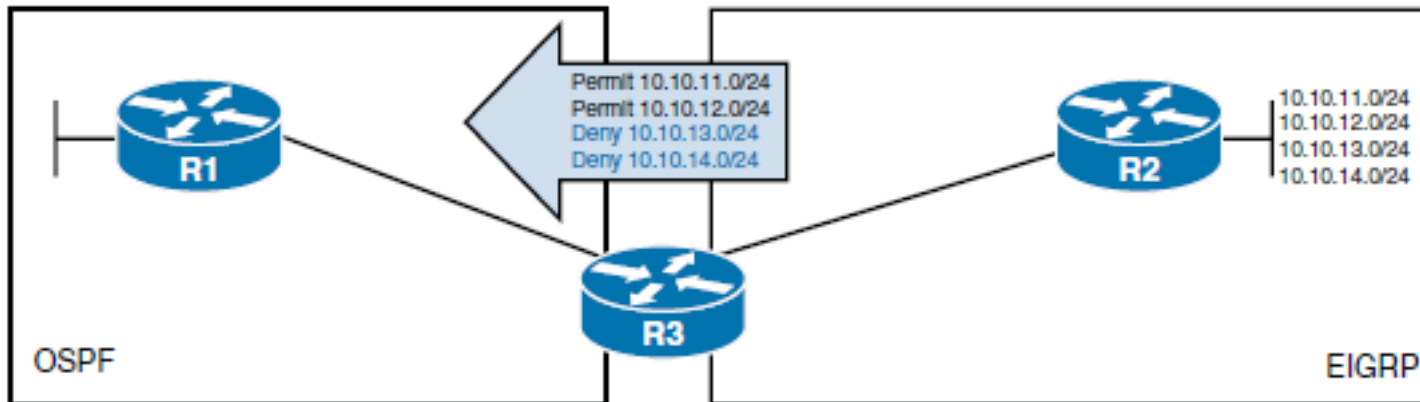


```

R3(config)# ip access-list standard ROUTE-FILTER
R3(config-std-nacl)# remark Outgoing Route Filter used with Distribute List
R3(config-std-nacl)# permit 10.10.11.0 0.0.0.255
R3(config-std-nacl)# permit 10.10.12.0 0.0.0.255
R3(config-std-nacl)# exit
R3(config)# router ospf 10
R3(config-router)# redistribute eigrp 100 metric 40 subnets
R3(config-router)# distribute-list ROUTE-FILTER out eigrp 100
R3(config-router)#
    
```



Distribute List and ACL Example [in]



```
R1(config)# ip access-list standard ROUTE-FILTER
R1(config-std-nacl)# remark Incoming Route Filter used with Distribute List
R1(config-std-nacl)# permit 10.10.11.0 0.0.0.255
R1(config-std-nacl)# permit 10.10.12.0 0.0.0.255
R1(config-std-nacl)# exit
R1(config)# router ospf 10
R1(config-router)# distribute-list ROUTE-FILTER in Ethernet 0/0
R1(config-router)#
```



Distribute List Issues

- Traditionally, route filtering was accomplished using ACLs with the **distribute-list** command; however, using ACLs as route filters for distribute lists has several drawbacks, including the following:
 - A **subnet mask** cannot be easily matched.
 - Access lists are **evaluated sequentially** for every IP prefix in the routing update.
 - Extended access lists can be cumbersome to configure.



Prefix List Characteristics

- Prefix lists are similar to access lists in many ways.
- **A prefix list can consist of any number of lines, each of which indicates a test and a result.**
- The router can interpret the lines in the specified order, although **Cisco IOS Software optimizes prefix lists** for processing in a tree structure.
- When a router evaluates a route against the prefix list, the first line that matches will result in either a “permit” or “deny.”
- **If none of the lines in the list match, the result is “implicitly deny.”**



The Advantages of Using Prefix Lists

- **Friendlier command-line interface**
- **Faster processing**
 - **A significant performance improvement** over access lists in loading and route lookup of large lists.
- **Support for incremental modifications**
 - **Sequence numbers** are assigned to **ip prefix-list** statements, making it easier to edit.
- **Greater flexibility**
 - Routers match networks in a routing update against the prefix list using as many bits as indicated. A prefix list can specify the exact size of the subnet mask, or it can indicate that the subnet mask must be in a **specified range**.

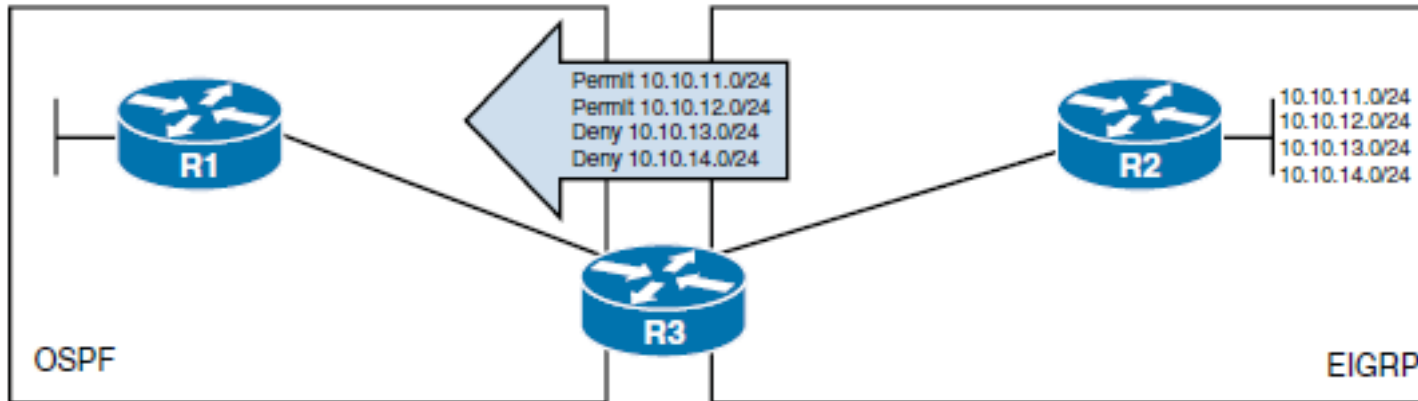


Configuring Prefix Lists

- ```
ip prefix-list { list-name | list-number } [seq seq-value] { deny | permit } network/ length [ge ge-value] [le le-value]
```

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>list-name</i>      | The name of the prefix list that will be created (case sensitive).                                                                                                                                                                                                                                                                                  |
| <i>list-number</i>    | The number of the prefix list that will be created.                                                                                                                                                                                                                                                                                                 |
| <i>seq seq-value</i>  | A 32-bit sequence number of the prefix-list statement, used to determine the order in which the statements are processed when filtering. Default sequence numbers are in increments of 5 (5, 10, 15, and so on). If no sequence value is configured, a new entry is assigned a sequence number equal to the current maximum sequence number plus 5. |
| <i>deny   permit</i>  | The action taken when a match is found.                                                                                                                                                                                                                                                                                                             |
| <i>network/length</i> | The prefix to be matched and the length of the prefix. The network is a 32-bit address. The length is a decimal number.                                                                                                                                                                                                                             |
| <i>ge ge-value</i>    | The range of the prefix length to be matched for prefixes that are more specific than network/length. The range is assumed to be from <i>ge value</i> to 32 if only the <i>ge</i> attribute is specified.                                                                                                                                           |
| <i>le le-value</i>    | The range of the prefix length to be matched for prefixes that are more specific than network/length. The range is assumed to be from length to <i>le value</i> if only the <i>le</i> attribute is specified.                                                                                                                                       |

# Distribute List and Prefix List Example

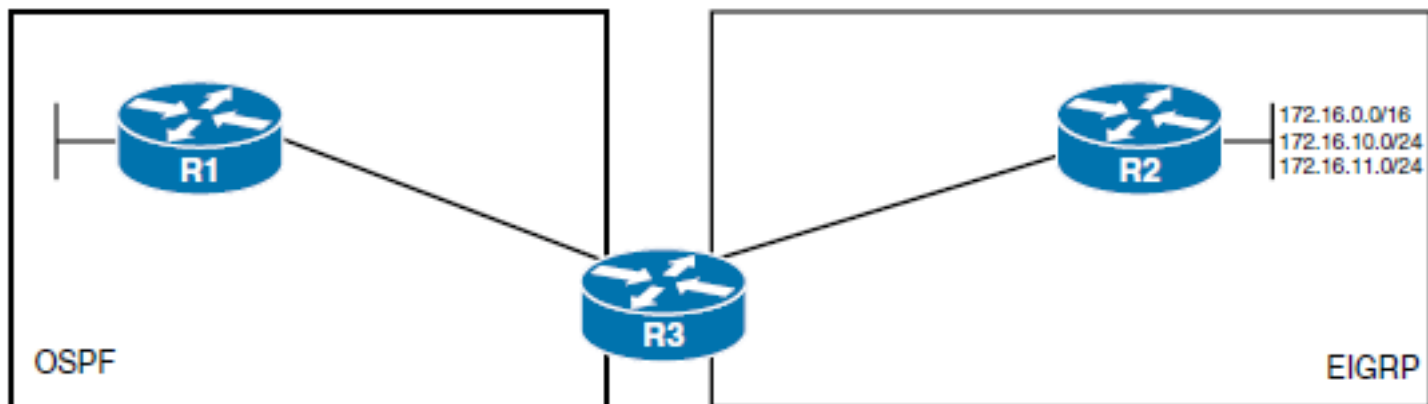


```
R3(config)# ip prefix-list FILTER-ROUTES description Outgoing Route Filter
R3(config)# ip prefix-list FILTER-ROUTES seq 5 permit 10.10.11.0/24
R3(config)# ip prefix-list FILTER-ROUTES seq 10 permit 10.10.12.0/24
R3(config)# router ospf 10
R3(config-router)# redistribute eigrp 100 metric 40 subnets
R3(config-router)# distribute-list prefix FILTER-ROUTES out eigrp 100
```





# Prefix List Examples



| Examples                                                      | Resulting Filter                                                                                                                                                                       |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>ip prefix-list TEST permit 172.0.0.0/8 le 24</pre>       | R1 learns about 172.16.0.0/16, 172.16.10.0/24, and 172.16.11.0/24. These are the routes that match the first 8 bits of 172.0.0.0 and have a prefix length between 8 and 24.            |
| <pre>ip prefix-list TEST permit 172.0.0.0/8 le 16</pre>       | R1 learns only about 172.16.0.0/16. This is the only route that matches the first 8 bits of 172.0.0.0 and has a prefix length between 8 and 16.                                        |
| <pre>ip prefix-list TEST permit 172.0.0.0/8 ge 17</pre>       | R1 learns only about 172.16.10.0/24 and 172.16.11.0/24. (In other words, Router A ignores the /8 parameter and treats the command as if it has the parameters ge 17 le 32.)            |
| <pre>ip prefix-list TEST permit 172.0.0.0/8 ge 16 le 24</pre> | R1 learns about 172.16.0.0/16, 172.16.10.0/24, and 172.16.11.0/24. (In other words, Router A ignores the /8 parameter and treats the command as if it has the parameters ge 16 le 24.) |
| <pre>ip prefix-list TEST permit 172.0.0.0/8 ge 17 le 23</pre> | R1 does not learn about any networks.                                                                                                                                                  |
| <pre>ip prefix-list TEST permit 0.0.0.0/0 le 32</pre>         | R1 learns about all EIGRP routes.                                                                                                                                                      |
| <pre>ip prefix-list TEST permit 0.0.0.0/0</pre>               | R1 learns only about a default route (if one exists).                                                                                                                                  |

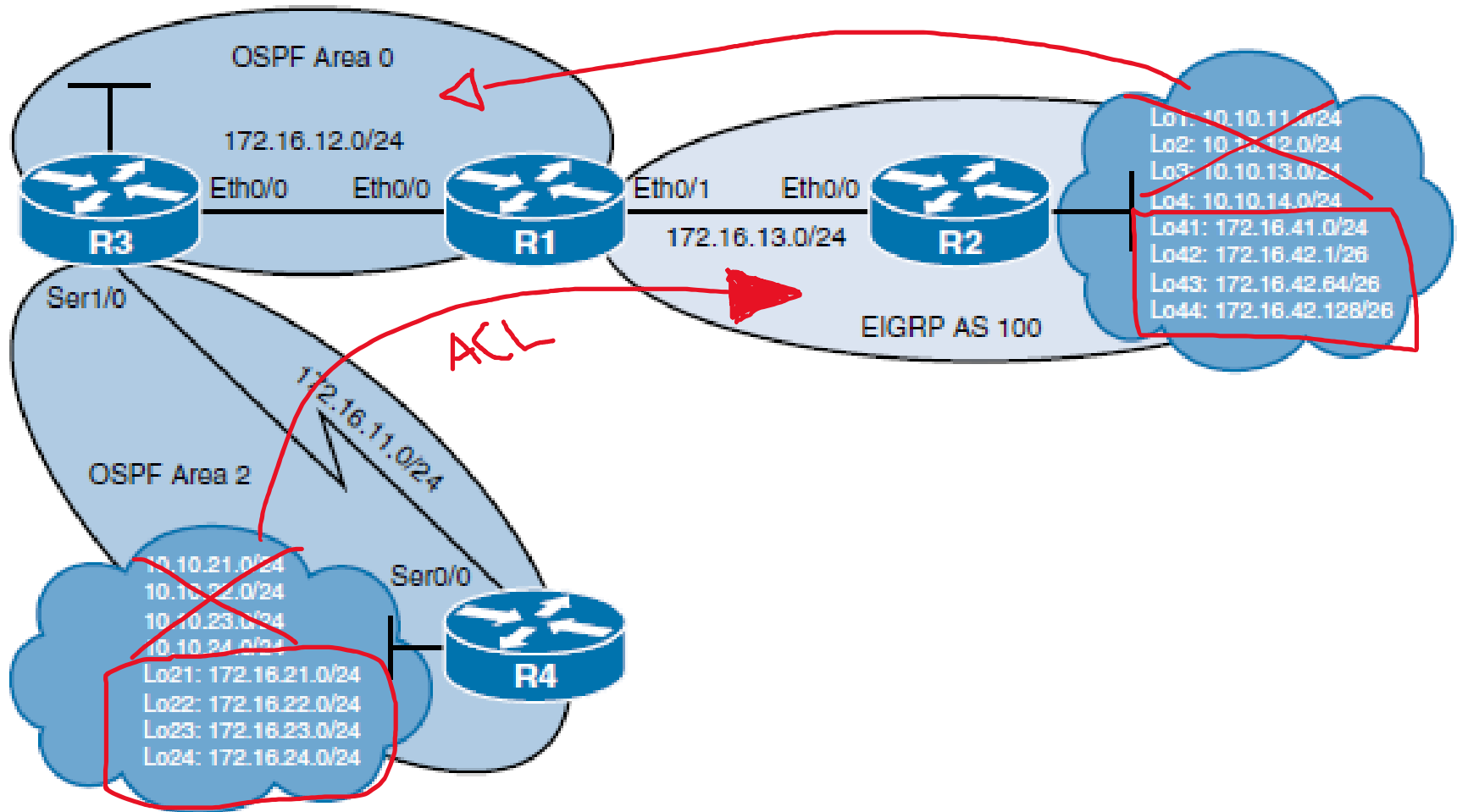


# Verifying Prefix Lists

| Command                                                                               | Description                                                                                                                                                                        |
|---------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>show ip prefix-list [detail   summary]</code>                                   | Displays information on all prefix lists. Specifying the detail keyword includes the description and the hit count (the number of times the entry matches a route) in the display. |
| <code>show ip prefix-list [detail   summary] <i>prefix-list-name</i></code>           | Displays a table showing the entries in a specific prefix list.                                                                                                                    |
| <code>show ip prefix-list <i>prefix-list-name</i> [network/length]</code>             | Displays the policy associated with a specific network/length in a prefix list.                                                                                                    |
| <code>show ip prefix-list <i>prefix-list-name</i> [seq <i>sequence-number</i>]</code> | Displays the prefix list entry with a given sequence number.                                                                                                                       |
| <code>show ip prefix-list <i>prefix-list-name</i> [network/length] longer</code>      | Displays all entries of a prefix list that are more specific than the given network and length.                                                                                    |
| <code>show ip prefix-list <i>prefix-list-name</i> [network/length] first-match</code> | Displays the entry of a prefix list that matches the network and length of the given prefix.                                                                                       |
| <code>clear ip prefix-list <i>prefix-list-name</i> [network/length]</code>            | Resets the hit count shown on prefix list entries.                                                                                                                                 |



# Manipulating Redistribution Using ACLs, Prefix Lists, and Distribute Lists





# Redistributing OSPFv2 Routes into the EIGRP Routing Domain Using an ACL and Distribute List

```
R1(config)# access-list 5 deny 10.10.21.0 0.0.0.255
R1(config)# access-list 5 deny 10.10.22.0 0.0.0.255
R1(config)# access-list 5 deny 10.10.23.0 0.0.0.255
R1(config)# access-list 5 deny 10.10.24.0 0.0.0.255
R1(config)# access-list 5 permit any
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 10 metric 1500 100 255 1 1500
R1(config-router)# distribute-list 5 out ospf 10
```

```
R2# show ip route eigrp
<Output omitted>
```

```
172.16.0.0/16 is variably subnetted, 16 subnets, 4 masks
D EX 172.16.11.0/30 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
D EX 172.16.12.0/24 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
D EX 172.16.21.1/32 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
D EX 172.16.22.1/32 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
D EX 172.16.23.1/32 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
D EX 172.16.24.1/32 [170/1757696] via 172.16.13.1, 1w0d, Ethernet0/0
```



# Redistributing EIGRP Routes into the OSPF Routing Domain Using a Prefix List and Distribute List

```
R1(config)# ip prefix-list EIGRP-TO-OSPF seq 5 permit 172.16.0.0/16 le 24
R1(config)# router ospf 10
R1(config-router)# redistribute eigrp 100 metric 40 subnets
R1(config-router)# distribute-list prefix EIGRP-TO-OSPF out eigrp 100
```

```
R1# show ip prefix-list detail
Prefix-list with the last deletion/insertion: EIGRP_TO_OSPF
ip prefix-list EIGRP_TO_OSPF:
 count: 1, range entries: 1, sequences: 5 - 5, refcount: 3
 seq 5 permit 172.16.0.0/16 le 24 (hit count: 2, refcount: 1)
```

```
R4# show ip route ospf
<Output omitted>

Gateway of last resort is not set

 172.16.0.0/16 is variably subnetted, 13 subnets, 3 masks
O IA 172.16.12.0/24 [110/74] via 172.16.11.2, 1w1d, Serial0/0
O E2 172.16.13.0/24 [110/20] via 172.16.11.2, 00:17:38, Serial0/0
O E2 172.16.41.0/24 [110/20] via 172.16.11.2, 00:17:38, Serial0/0
```



# Understanding Route Maps



Route maps are **complex access lists** that allow some conditions to be tested against the **packet or route** in question using **match** commands.



If the conditions match, some **actions can be taken** to modify attributes of the packet or route.



These actions are specified by **set** commands.



A collection of **route-map** statements that have the same route map name is considered one route map.



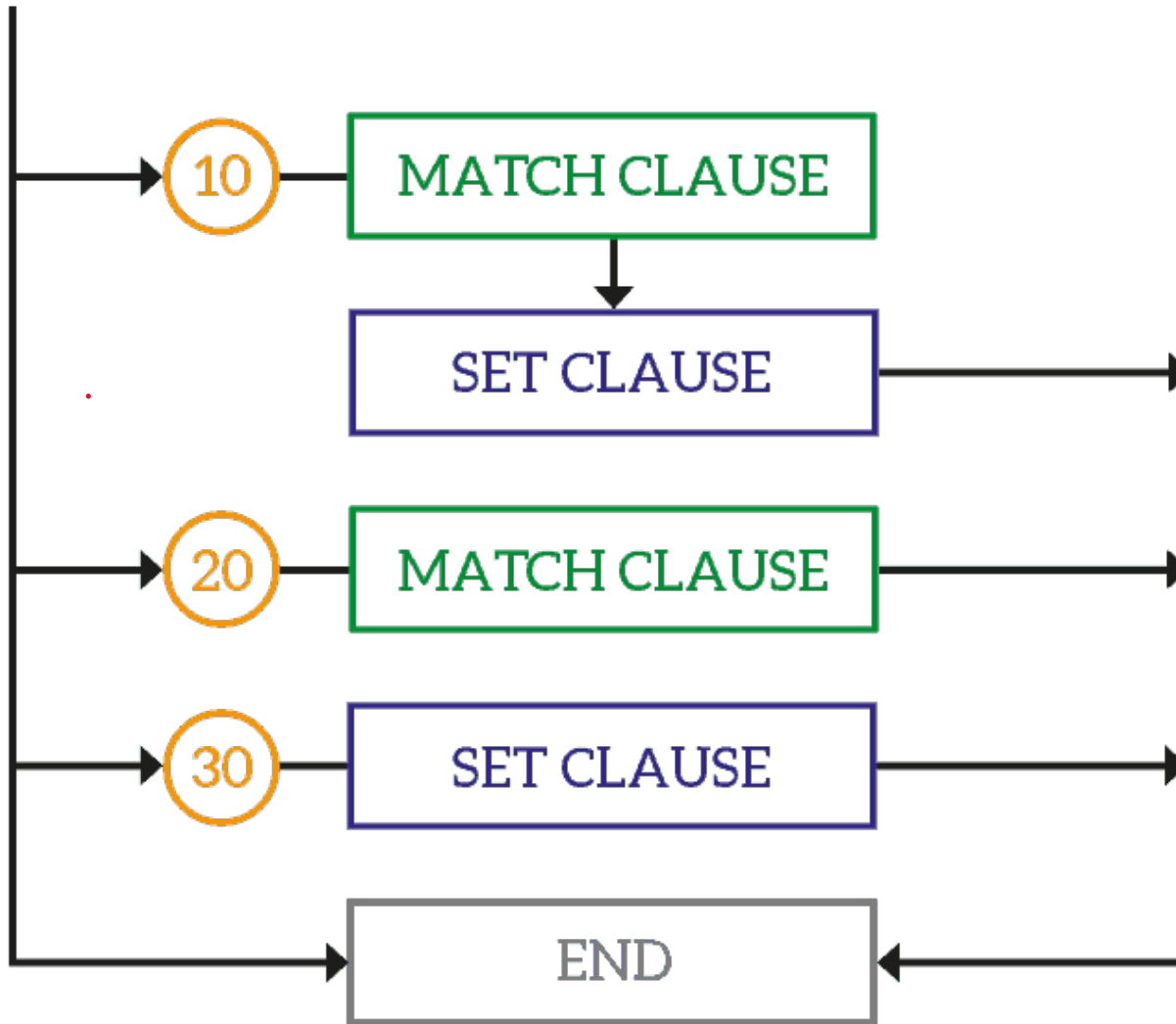
Within a route map, each **route-map** statement is numbered and therefore can be edited individually.



The statements in a route map correspond to the lines of an access list.



# Understanding Route Maps





# Route Map Applications

## ■ Route filtering during redistribution

- Route maps offer the **benefit of manipulating routing metrics** through the use of **set** commands.
- The route map is **applied using** the **redistribute** command.

## ■ Policy-based routing (PBR)

- Route maps can be used to match **source** and **destination addresses**, **protocol types**, and **end-user applications**. When a match occurs, a **set** command can be used to define the **interface** or **next-hop address** to which the packet should be sent.
- The route map is applied to an interface using the **ip policy route-map** interface configuration command.

## ■ BGP

- In addition to filtering, route maps **provide sophisticated** manipulation of **BGP path attributes**. The route map is applied using the BGP **neighbor** router configuration command.





# Configuring Route Maps

**Step 1.** Define the route map using the **route-map** global configuration command.

**Step 2.** Define the **matching conditions** using the **match** command and **optionally the action to be taken** when each condition is matched using the **set** command.

**Step 3.** Apply the route map.



# Define the Route Map

- `route-map map-tag [ permit | deny ] [ sequence-number ]`

| Parameter                  | Description                                                                                                                                                                                                                                                                                  |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>map-tag</i>             | Name of the route map.                                                                                                                                                                                                                                                                       |
| <code>permit   deny</code> | (Optional) A parameter that specifies the action to be taken if the route map match conditions are met. The meaning of <b>permit</b> or <b>deny</b> is dependent on how the route map is used. The default for the <b>route-map</b> command is <b>permit</b> , with a sequence number of 10. |
| <i>sequence-number</i>     | (Optional) A sequence number that indicates the position that a new <b>route-map</b> statement will have in the list of <b>route-map</b> statements already configured with the same name.                                                                                                   |



# Route Map Rules

- Route map **sequence numbers do not automatically increment**. When the sequence-number parameter of the route-map command is not used, the following occurs:
  - If no other entry is already defined with the supplied route-map map-tag , an entry is created, with the sequence-number set to 10.
  - If only one entry is already defined with the supplied route-map tag, that entry is the default entry for the route-map command, and the sequence-number of the entry is unchanged.
  - If more than one entry is already defined with the supplied route-map tag, an error message is displayed, indicating that the sequence-number is required.
  - If the no route-map map-tag command is specified (without the sequence-number parameter), the whole route map is deleted.
- Like an access list, an implicit **deny any appears at the end of a route map**. The consequences of this deny depend on how the route map is being used.
- The **match condition** route map configuration commands are used to define the conditions to be checked. The **set condition** route map configuration commands are used to define the actions to be followed if there is a match and the action to be taken is permit.
- A route-map statement **without any match statements** will be considered **matched**.



# Demonstration of the route-map Command

```

route-map DEMO permit 10
 match X Y Z
 match A
 set B
 set C
route-map DEMO permit 20
 match Q
 set R
route-map DEMO permit 30

```

The route map named DEMO in Example 4-27 is interpreted as follows:

If {(X or Y or Z) and (A) match} then {set B and C}

Else

If Q matches then set R

Else

Set nothing



# Define the Matching Conditions

| Command                                                                                                                                                                                | Description                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>match ip address {<i>access-list-number</i>   <i>name</i>} [...<i>access-list-number</i>   <i>name</i>]   prefix-list <i>prefix-list-name</i> [..<i>prefix-list-name</i>]</code> | Matches any routes that have a network number that is permitted by a standard or extended access list or prefix list. Multiple access lists or prefix lists can be specified; matching any one results in a match. |
| <code>match length <i>min max</i></code>                                                                                                                                               | Matches based on a packet's Layer 3 length.                                                                                                                                                                        |
| <code>match interface <i>type number</i></code>                                                                                                                                        | Matches any routes that have the next hop out of one of the interfaces specified.                                                                                                                                  |
| <code>match ip next-hop {<i>access-list-number</i>   <i>access-list-name</i>} [...<i>access-list-number</i>   ... <i>access-list-name</i>]</code>                                      | Matches any routes that have a next-hop router address permitted by one of the access lists specified.                                                                                                             |
| <code>match ip route-source {<i>access-list-number</i>   <i>access-list-name</i>} [...<i>access-list-number</i>   ... <i>access-list-name</i>]</code>                                  | Matches routes that have been advertised by routers and access servers that have an address permitted by one of the access lists specified.                                                                        |
| <code>match metric <i>metric-value</i></code>                                                                                                                                          | Matches routes that have the metric specified.                                                                                                                                                                     |
| <code>match route-type [external   internal   level-1   level-2   local]</code>                                                                                                        | Matches routes of the specified type.                                                                                                                                                                              |
| <code>match community {<i>list-number</i>   <i>list-name</i>}</code>                                                                                                                   | Matches a BGP community.                                                                                                                                                                                           |
| <code>match tag <i>tag-value</i></code>                                                                                                                                                | Matches based on the tag of a route.                                                                                                                                                                               |



# Define the Set Actions

| Command                                                                       | Description                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>set metric <i>metric-value</i></code>                                   | Sets the metric value for a routing protocol.                                                                                                                                                                                                                                              |
| <code>set metric-type [type-1   type-2   internal   external]</code>          | Sets the metric type for the destination routing protocol.                                                                                                                                                                                                                                 |
| <code>set default interface <i>type number</i> [...<i>type number</i>]</code> | Indicates where to send output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS Software has no explicit route to the destination.                                                                                                           |
| <code>set interface <i>type number</i> [...<i>type number</i>]</code>         | Indicates where to send output packets that pass a match clause of a route map for policy routing.                                                                                                                                                                                         |
| <code>set ip default next-hop <i>ip-address</i> [...<i>ip-address</i>]</code> | Indicates where to send output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS Software has no explicit route to the destination.                                                                                                           |
| <code>set ip default next-hop verify-availability</code>                      | Forces the router to check the CDP database to determine if an entry is available for the next hop that is specified by the <code>set ip default next-hop</code> command. This command is used to prevent traffic from being “black holed” if the configured next hop becomes unavailable. |
| <code>set ip next-hop <i>ip-address</i> [...<i>ip-address</i>]</code>         | Indicates where to send output packets that pass a match clause of a route map for policy routing.                                                                                                                                                                                         |

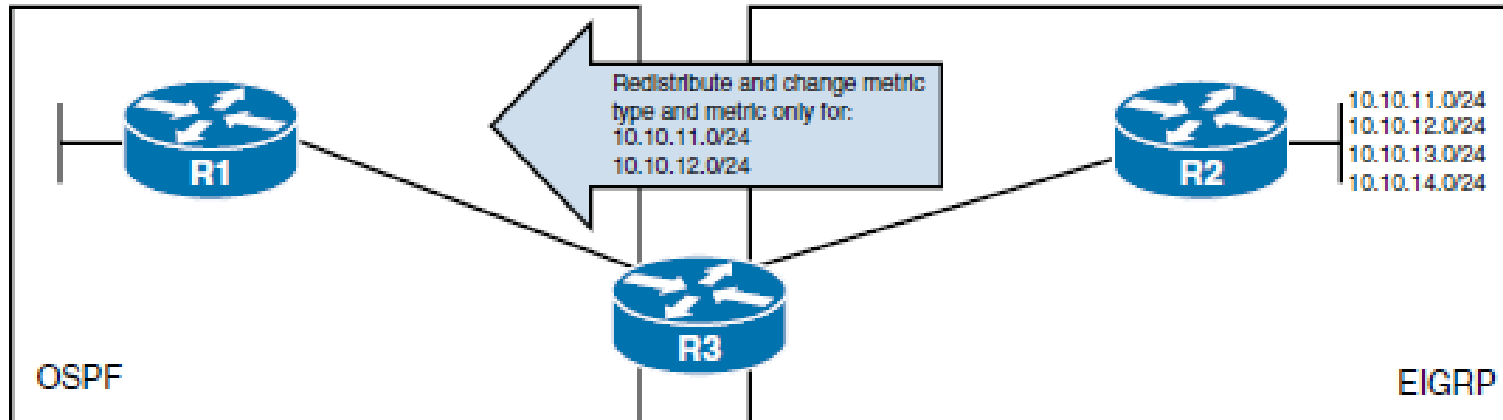


# Define the Set Actions

| Command                                                                                | Description                                                                                                                                                                          |
|----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>set ip next-hop verify-availability</code>                                       | Forces the router to check the Cisco Discovery Protocol (CDP) database or use object tracking to determine if the next hop that is specified for policy-based routing is available.  |
| <code>set ip vrf</code>                                                                | Indicates where to forward packets that pass a match clause of a route map for policy routing when the next hop must be under a specified Virtual Routing and Forwarding (VRF) name. |
| <code>set next-hop</code>                                                              | Specifies the address of the next hop.                                                                                                                                               |
| <code>set level [level-1   level-2   stub-area   backbone]</code>                      | Indicates at what level or type of area to import routes into (for IS-IS and OSPF routes).                                                                                           |
| <code>set as-path {tag   prepend <i>as-path-string</i>}</code>                         | Modifies an autonomous system path for BGP routes.                                                                                                                                   |
| <code>set automatic-tag</code>                                                         | Automatically computes the BGP tag value.                                                                                                                                            |
| <code>set community {community-number [additive] [well-known-community]   none}</code> | Sets the BGP community attribute.                                                                                                                                                    |
| <code>set local-preference <i>bgp-path-attributes</i></code>                           | Specifies a local preference value for the BGP autonomous system path.                                                                                                               |
| <code>set weight <i>bgp-weight</i></code>                                              | Specifies the BGP weight value.                                                                                                                                                      |
| <code>set origin <i>bgp-origin-code</i></code>                                         | Specifies the BGP origin code.                                                                                                                                                       |
| <code>set tag</code>                                                                   | Specifies the tag value for destination routing protocol.                                                                                                                            |



# Using Route Maps with Redistribution



```

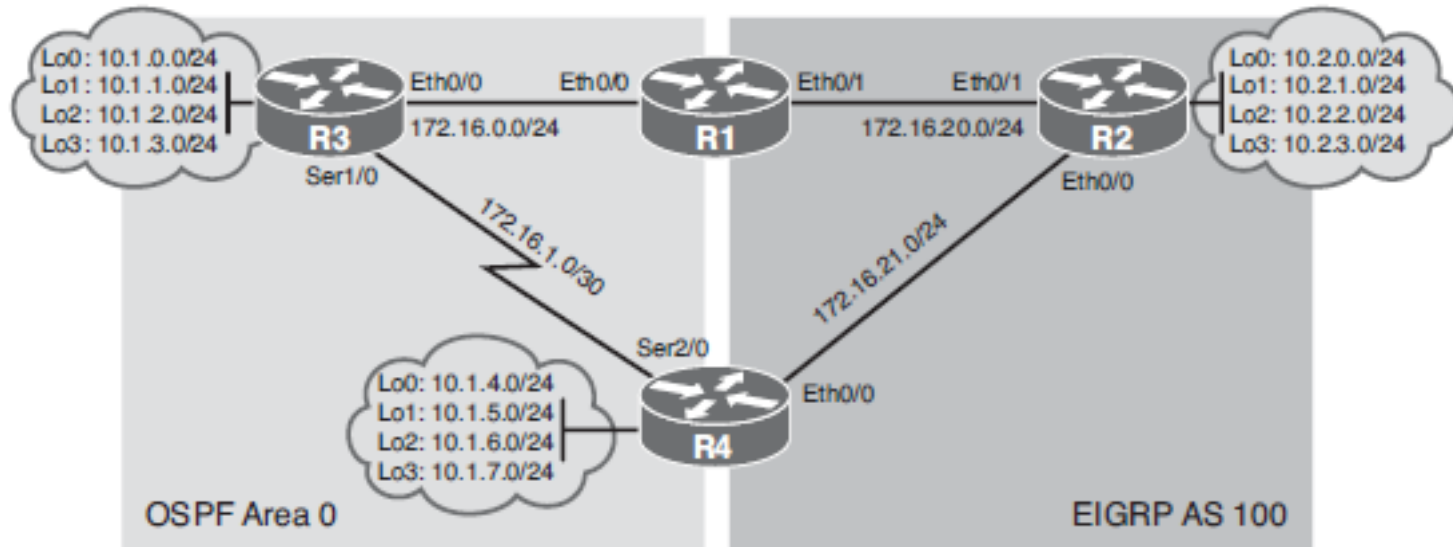
R3(config)# ip prefix-list FILTER-ROUTES permit 10.10.11.0/24
R3(config)# ip prefix-list FILTER-ROUTES permit 10.10.12.0/24
R3(config)# route-map RM-INTO-OSPF permit 10
R3(config-route-map)# match ip address prefix-list FILTER-ROUTES
R3(config-route-map)# set metric 25
R3(config-route-map)# set metric-type type-1
R3(config-route-map)# exit
R3(config)# router ospf 10
R3(config-router)# redistribute eigrp 100 subnets route-map RM-INTO-OSPF

```





# Manipulating Redistribution Using Route Maps



- R1 and R4 will be configured to support mutual redistribution without any filtering mechanism.
- R1 and R4 will be configured to support mutual redistribution using route maps.
- Change administrative distance for certain routes to enable optimal routing.



# Mutual Redistribution Without Filters

```
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 10 metric 10000 10 200 5 1500
R1(config-router)# exit
R1(config)# router ospf 10
R1(config-router)# redistribute eigrp 100 subnets
```

```
R4(config)# router eigrp 100
R4(config-router)# redistribute ospf 10 metric 10000 10 200 5 1500
R4(config-router)# exit
R4(config)# router ospf 10
R4(config-router)# redistribute eigrp 100 subnets
```



# Verifying Redistributed Routes

```

R3# show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
 ia - IS-IS inter area, * - candidate default, U - per-user static route
 o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
 + - replicated route, % - next hop override

Gateway of last resort is not set

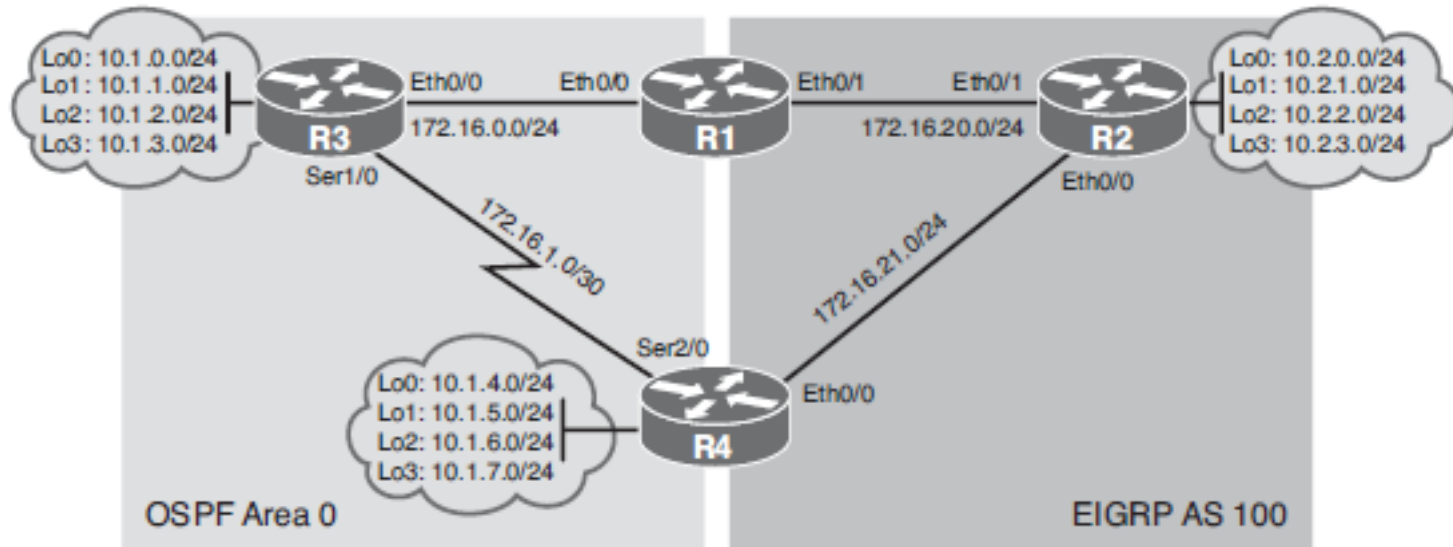
 10.0.0.0/8 is variably subnetted, 16 subnets, 2 masks
O 10.1.4.0/24 [110/65] via 172.16.1.2, 00:20:02, Serial1/0
O 10.1.5.0/24 [110/65] via 172.16.1.2, 00:20:02, Serial1/0
O 10.1.6.0/24 [110/65] via 172.16.1.2, 00:20:02, Serial1/0
O 10.1.7.0/24 [110/65] via 172.16.1.2, 00:20:02, Serial1/0
O E2 10.2.0.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0
O E2 10.2.1.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0
O E2 10.2.2.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0
O E2 10.2.3.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0

 172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
O E2 172.16.20.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0
O E2 172.16.21.0/24 [110/20] via 172.16.0.2, 00:58:31, Ethernet0/0

```



# Manipulating Redistribution Using Route Maps



- R1 and R4 will be configured to support mutual redistribution without any filtering mechanism.
- R1 and R4 will be configured to support mutual redistribution using route maps.
- Change administrative distance for certain routes to enable optimal routing.



# Mutual Redistribution with Route Maps

```

R1(config)# access-list 10 permit 10.2.0.0 0.0.3.255
R1(config)# access-list 20 permit 10.1.0.0 0.0.7.255
R1(config)# route-map INTO-OSPF permit 10
R1(config-route-map)# match ip address 10
R1(config-route-map)# exit
R1(config)# route-map INTO-EIGRP permit 10
R1(config-route-map)# match ip address 20
R1(config-route-map)# set metric 10000 10 200 5 1500

```

```

R4(config)# access-list 10 permit 10.2.0.0 0.0.3.255
R4(config)# access-list 20 permit 10.1.0.0 0.0.7.255
R4(config)# route-map INTO-OSPF permit 10
R4(config-route-map)# match ip address 10
R4(config-route-map)# exit
R4(config)# route-map INTO-EIGRP permit 10
R4(config-route-map)# match ip address 20
R4(config-route-map)# set metric 10000 10 200 5 1500

```



# Mutual Redistribution with Route Maps

```
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 10 route-map INTO-EIGRP
R1(config-router)# exit
R1(config)# router ospf 10
R1(config-router)# redistribute eigrp 100 subnets route-map INTO-OSPF
```

```
R4(config)# router eigrp 100
R4(config-router)# redistribute ospf 10 route-map INTO-EIGRP
R4(config-router)# exit
R4(config)# router ospf 10
R4(config-router)# redistribute eigrp 100 subnets route-map INTO-OSPF
```



# Verifying Redistributed Routes

```
R3# show ip route ospf
```

```
<Output omitted>
```

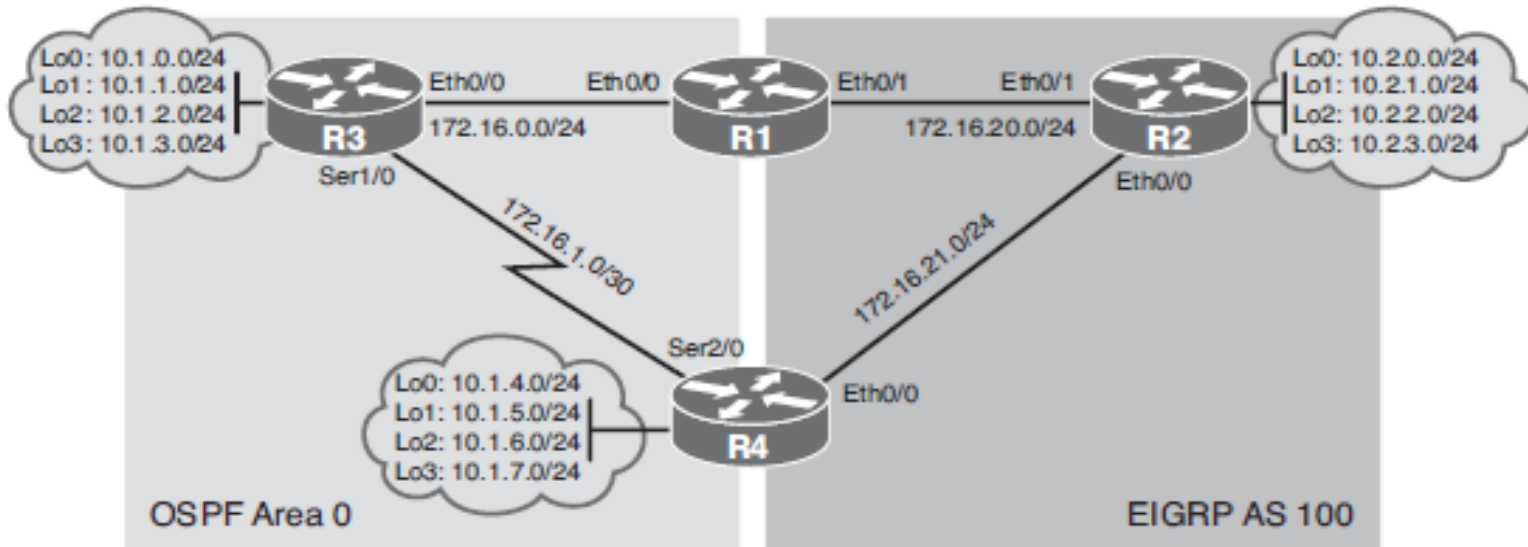
```
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 16 subnets, 2 masks
```

```
O 10.1.4.0/24 [110/65] via 172.16.1.2, 00:30:02, Serial1/0
O 10.1.5.0/24 [110/65] via 172.16.1.2, 00:33:48, Serial1/0
O 10.1.6.0/24 [110/65] via 172.16.1.2, 00:33:48, Serial1/0
O 10.1.7.0/24 [110/65] via 172.16.1.2, 00:33:38, Serial1/0
O E2 10.2.0.0/24 [110/20] via 172.16.0.2, 01:40:23, Ethernet0/0
O E2 10.2.1.0/24 [110/20] via 172.16.0.2, 01:40:23, Ethernet0/0
O E2 10.2.2.0/24 [110/20] via 172.16.0.2, 01:40:23, Ethernet0/0
O E2 10.2.3.0/24 [110/20] via 172.16.0.2, 01:40:23, Ethernet0/0
```



# Change Administrative Distance to Enable Optimal Routing



- R1 and R4 will be configured to support mutual redistribution without any filtering mechanism.
- R1 and R4 will be configured to support mutual redistribution using route maps.
- Change administrative distance for certain routes to enable optimal routing.





# Examine R1 Routing Table

```

R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
 ia - IS-IS inter area, * - candidate default, U - per-user static route
 o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
 + - replicated route, % - next hop override

Gateway of last resort is not set

 10.0.0.0/24 is subnetted, 12 subnets
O 10.1.0.0 [110/11] via 172.16.0.1, 03:47:09, Ethernet0/0
O 10.1.1.0 [110/11] via 172.16.0.1, 03:47:09, Ethernet0/0
O 10.1.2.0 [110/11] via 172.16.0.1, 03:47:09, Ethernet0/0
O 10.1.3.0 [110/11] via 172.16.0.1, 03:47:09, Ethernet0/0
O 10.1.4.0 [110/75] via 172.16.0.1, 00:32:22, Ethernet0/0
O 10.1.5.0 [110/75] via 172.16.0.1, 00:36:08, Ethernet0/0
O 10.1.6.0 [110/75] via 172.16.0.1, 00:36:08, Ethernet0/0
O 10.1.7.0 [110/75] via 172.16.0.1, 00:35:58, Ethernet0/0
D 10.2.0.0 [90/409600] via 172.16.20.2, 03:41:39, Ethernet0/1
D 10.2.1.0 [90/409600] via 172.16.20.2, 03:41:39, Ethernet0/1
D 10.2.2.0 [90/409600] via 172.16.20.2, 03:41:39, Ethernet0/1
D 10.2.3.0 [90/409600] via 172.16.20.2, 03:41:39, Ethernet0/1

 172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
C 172.16.0.0/24 is directly connected, Ethernet0/0
L 172.16.0.2/32 is directly connected, Ethernet0/0
O 172.16.1.0/30 [110/74] via 172.16.0.1, 01:04:30, Ethernet0/0
C 172.16.20.0/24 is directly connected, Ethernet0/1
L 172.16.20.1/32 is directly connected, Ethernet0/1
D 172.16.21.0/24 [90/281856] via 172.16.20.2, 03:41:39, Ethernet0/1

```



# Changing External Route Administrative Distance

```

R1(config)# router eigrp 100
R1(config-router)# distance eigrp 90 100
R1(config-router)# ^Z
R1#
R1#
*Jul 21 16:08:00.454: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.16.20.2
(Ethernet0/1) is down: route configuration changed
R1#
*Jul 21 16:08:03.705: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.16.20.2
(Ethernet0/1) is up: new adjacency

```

```

R1# show ip route
<Output omitted>

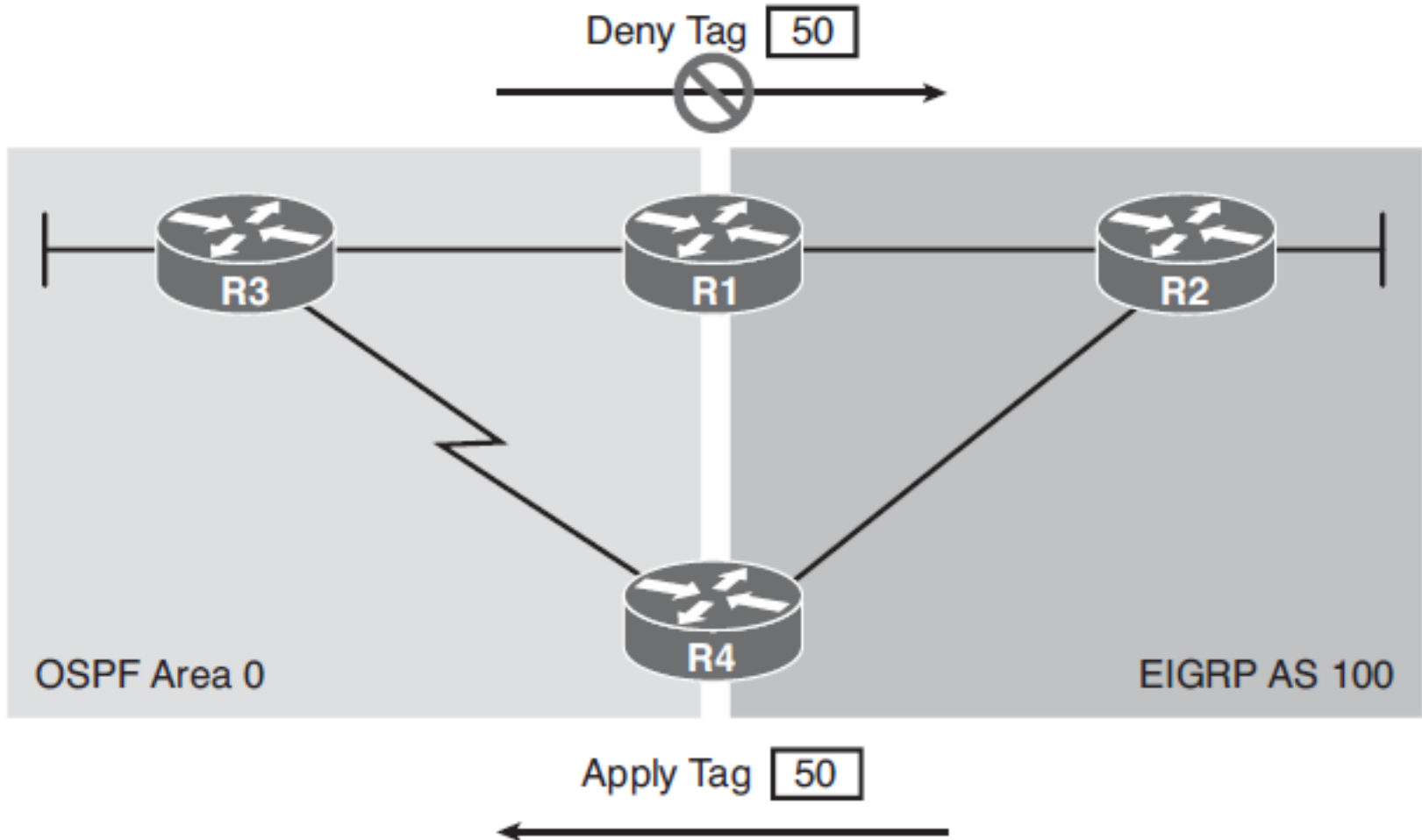
Gateway of last resort is not set

 10.0.0.0/24 is subnetted, 12 subnets
D EX 10.1.0.0 [100/284416] via 172.16.20.2, 00:00:26, Ethernet0/1
D EX 10.1.1.0 [100/284416] via 172.16.20.2, 00:00:26, Ethernet0/1
D EX 10.1.2.0 [100/284416] via 172.16.20.2, 00:00:26, Ethernet0/1
D EX 10.1.3.0 [100/284416] via 172.16.20.2, 00:00:26, Ethernet0/1
D EX 10.1.4.0 [100/284416] via 172.16.20.2, 00:00:26, Ethernet0/1

```



# Manipulating Redistribution Using Route Tagging





# Manipulating Redistribution Using Route Tagging

```
R4(config)# route-map EIGRP-TO-OSPF permit 10
R4(config-route-map)# set tag 50
R4(config-route-map)# exit
R4(config)# router ospf 10
R4(config-router)# redistribute eigrp 100 subnets route-map EIGRP-TO-OSPF
```

```
R1(config)# route-map OSPF-TO-EIGRP deny 10
R1(config-route-map)# match tag 50
R1(config-route-map)# exit
R1(config)# route-map OSPF-TO-EIGRP permit 20
R1(config-route-map)# exit
R1(config)# router eigrp 100
R1(config-router)# redistribute ospf 10 metric 1000 1 255 1 1500 route-map OSPF-TO-EIGRP
```



# Caveats of Redistribution

- Redistribution of routing information adds to the **complexity** of a network and **increases** the potential for **routing confusion**, so **you should use it only when necessary**.
- The key issues that arise when you are using redistribution are as follows:
  - **Routing loops**
  - **Incompatible routing information**
  - **Inconsistent convergence time**



# Chapter 4 Summary

This chapter covered how to support multiple routing protocols by using redistribution and route filtering techniques, through discussion of the following topics:

- Reasons for using more than one routing protocol (migration, host system needs, mixed-vendor environment, political and geographic borders, Multiprotocol Label Switching [MPLS] virtual private networks [VPNs]).
- Routing information can be exchanged between them (referred to as redistribution), and how Cisco routers operate in a multiple routing protocol environment.
- Route redistribution is always performed *outbound* . The router doing redistribution does not change its routing table.



# Chapter 4 Summary

- A router assigns a seed metric to redistributed routes using the **default-metric** router configuration command or the **redistribute** command with the **metric** parameter.
- The redistribution techniques, one-point and multipoint:
  - The two methods of one-point route redistribution are one-way and two-way. Suboptimal routing is a possible issue with these techniques.
  - The two methods of multipoint route redistribution are one-way and two-way. Multipoint redistribution is likely to introduce potential routing loops.



# Chapter 4 Summary

- To prevent routing issues, use one of the following options:
  - Redistribute a default route from the core autonomous system into the edge autonomous system, and redistribute routes from the edge routing protocols into the core routing protocol.
  - Redistribute multiple static routes about the core autonomous system networks into the edge autonomous system, and redistribute routes from the edge routing protocols into the core routing protocol.
  - Redistribute routes from the core autonomous system into the edge autonomous system with filtering to block out inappropriate routes.
  - Redistribute all routes from the core autonomous system into the edge autonomous system, and from the edge autonomous system into the core autonomous system, and then modify the administrative distance associated with redistributed routes so that they are not the selected routes when multiple routes exist for the same destination.





# Chapter 4 Summary

- Configuration of redistribution between various IP routing protocols:
  - To redistribute into EIGRP, use the **redistribute** *protocol* [ *process-id* ] [ **match** *route-type* ] [ **metric** *metric-value* ] [ **route-map** *map-tag* ] router configuration command.
  - To redistribute into OSPF, use the **redistribute** *protocol* [ *process-id* ] [ **metric** *metric-value* ] [ **metric-type** *type-value* ] [ **route-map** *map-tag* ] [ **subnets** ] [ **tag** *tagvalue* ] router configuration command.
- Using the **show ip route** [ *ip-address* ] and **traceroute** [ *ip-address* ] commands to verify route redistribution.



# Chapter 4 Summary

- Distribute lists, allowing an access list to be applied to routing updates:
  - The **distribute-list** { *access-list-number* | *name* } **out** [ *interface-name* ] router configuration command assigns the access list to filter outgoing routing updates. This command filters updates going out of the interface or routing protocol specified in the command, into the routing process under which it is configured.
  - The **distribute-list** { *access-list-number* | *name* } [ **route-map** *map-tag* ] **in** [ *interface-type* *interface-number* ] router configuration command assigns the access list to filter routing updates coming in through an interface. This command filters updates going into the interface specified in the command, into the routing process under which it is configured.



# Chapter 4 Summary

- Prefix lists can be used with distribute lists as an alternative to ACLs, with improvements in performance, support for incremental modifications, a more userfriendly command-line interface, and greater flexibility. Prefix lists are configured with the **ip prefix-list** { *list-name* | *list-number* } [ **seq** *seq-value* ] { **deny** | **permit** } *network* / *length* [ **ge** *ge-value* ] [ **le** *le-value* ] global configuration command.
- Whether a prefix in a prefix list is permitted or denied is based on the following rules:
  - An empty prefix list permits all prefixes.
  - If a prefix is permitted, the route is used. If a prefix is denied, the route is not used.
  - Prefix lists consist of statements with sequence numbers. The router begins the search for a match at the top of the prefix list, which is the statement with the lowest sequence number.



# Chapter 4 Summary

- When a match occurs, the router does not need to go through the rest of the prefix list. For efficiency, you might want to put the most common matches (permits or denies) near the top of the list by specifying a lower sequence number.
- An implicit **deny** is assumed if a given prefix does not match any entries in a prefix list.
- **Prefix list sequence numbers:**
  - Sequence numbers are generated automatically, unless you disable this automatic generation.
  - A prefix list is an ordered list. The sequence number is significant when a given prefix is matched by multiple entries of a prefix list, in which case the one with the smallest sequence number is considered the real match.



# Chapter 4 Summary

- The evaluation of a prefix list starts with the lowest sequence number and continues down the list until a match is found, in which case the **permit** or **deny** statement is applied to that network and the remainder of the list is not evaluated.
- Using route maps for route filtering during redistribution, PBR, and BGP.
- The characteristics of route maps, configured using the **route-map map-tag [ permit | deny ] [ sequence-number ]** global configuration command:
- Route maps allow some conditions to be tested against the packet or route in question using **match** commands. If the conditions match, some actions can be taken to modify attributes of the packet or route; these actions are specified by **set** commands.

Chapter 4   A collection of **route-map** statements that have the



# Chapter 4 Summary

- A collection of **route-map** statements that have the same route map name is considered one route map.
- Within a route map, each **route-map** statement is numbered and therefore can be edited individually.
- The default for the **route-map** command is **permit** , with a *sequence-number* of 10.
- Only one condition listed on the same **match** statement must match for the entire statement to be considered a match. However, all **match** statements within a **route-map** statement must match for the route map to be considered matched.



# Chapter 4 Summary

- When used with a **redistribute** command, a **route-map** statement with **permit** indicates that the matched route is to be redistributed, and a **route-map** statement with **deny** indicates that the matched route is not to be redistributed.



# Chapter 4 Labs

- **CCNPv7 ROUTE Lab 4.1-Redistribution EIGRP OSPF**
- **CCNPv7 ROUTE Lab 4.2-Controlling Routing Updates**
- **CCNPv7 ROUTE Lab 4.3-Redistribution EIGRP for IPv6 and OSPFv3**



# Cisco | Networking Academy<sup>®</sup>

Mind Wide Open<sup>™</sup>



# Acknowledgment

- Some of images and texts are from Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide by Diane Teare, Bob Vachon and Rick Graziani (1587204568)
- Copyright © 2015 – 2016 Cisco Systems, Inc.
- Special Thanks to *Bruno Silva*