Lab 6: Redistribution Between Cisco EIGRP RIP and OSPF

1. Basic Redistribution Between Cisco EIGRP and OSPF
2. Configure Redistribution between RIP and OSPF in Cisco IOS Router
3. **Why do we use the ip ospf network point-to-point command on loopback interfaces?**

Basic Redistribution Between Cisco EIGRP and OSPF

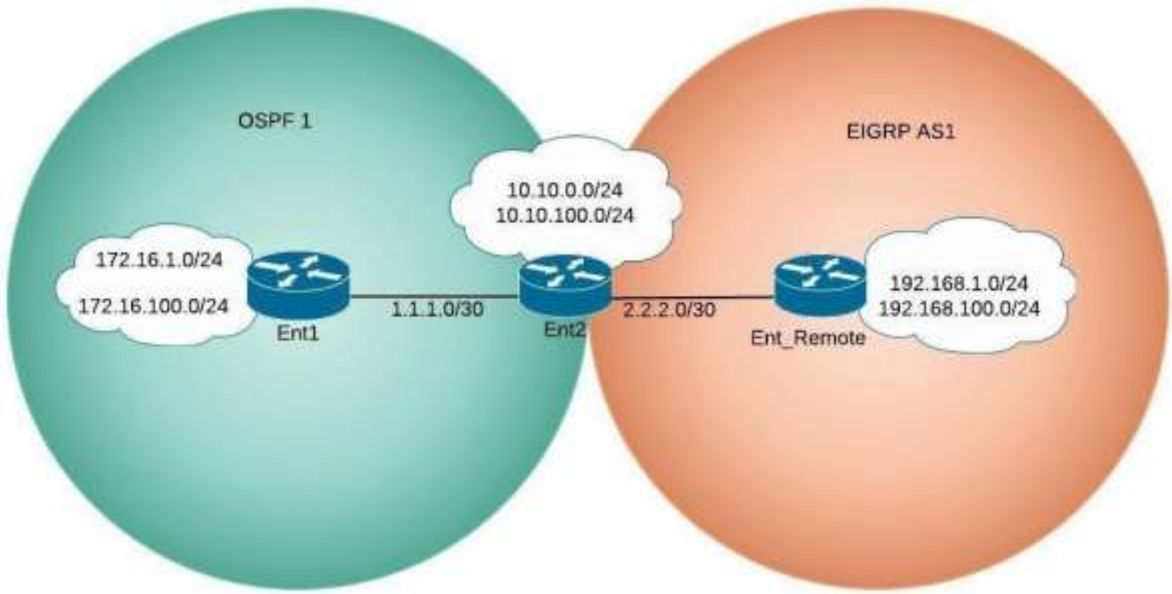
The most common scenario for big enterprise networks with multiple routers is to have a single IGP routing protocol (IGP = Interior Gateway Protocol) running between the routers in order to distribute all the routing information of the network.

The most common IGP routing protocols used by today's networks are OSPF, EIGRP (Cisco proprietary) and in some cases IS-IS. RIP also is an IGP but is not used anymore. It is found only in old legacy networks (or in lab environments for study purposes).

The above situation of having a single routing protocol is the most common case. However, there are networks that use two different IGP protocols for various reasons (such as merge of two companies that use different routing protocols, migration from one protocol to another, installation of router devices from a different vendor etc). If you have two IGP routing protocols in the same network, then you must perform **Routing Redistribution** in order to exchange routing information between the two protocols.

Note that you can configure and enable two routing protocols on a router (e.g both OSPF and EIGRP). However, this is not enough to exchange information between the two routing protocols. You must configure Redistribution as well.

The tutorial below will focus on Routing Redistribution between OSPF and EIGRP and vice- versa. As the title says, this is an introductory tutorial in order to learn the basics of router redistribution. The network topology shown below is very basic but it will help you to expand the concepts in bigger networks.



Network Scenario

From the topology above, assume we have a remote enterprise network (RED) running EIGRP only on router "**Ent_Remote**". Also, we have another enterprise network (BLUE) with two routers (ENT1 and ENT2). ENT1 is running OSPF only and ENT2 is running both OSPF and EIGRP.

Our goal is to have all router devices learn all the routes from both networks. That is, we want to redistribute the routes of "Ent_Remote" into the routing table of router "ENT1" (which is running only OSPF) and vice-versa.

This scenario is often seen in real life when two companies merge and they have to converge without causing downtime or modifications to the networks.

Company BLUE is running OSPF and has the following networks: 10.10.0.0/24 & 10.10.100.0/24 behind router ENT2 and 172.16.1.0/24 and 172.16.100.0/24 behind ENT1.

Company RED (running EIGRP) has merged with BLUE and has networks 192.168.1.0/24 and 192.168.100.0/24 behind router "ENT_remote"

(The networks are emulated using loopbacks.)

Configuration

NOTE: You can find the full router configurations at the end of this tutorial.

First, let's see the routing table on ENT1 before the redistribution:

ENT1#sh ip route

```
     1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       1.1.1.0/30 is directly connected, FastEthernet4
L       1.1.1.1/32 is directly connected, FastEthernet4
     10.0.0.0/32 is subnetted, 2 subnets
O 10.10.0.1 [110/2] via 1.1.1.2, 02:25:53, FastEthernet4 O 10.10.100.1
        [110/2] via 1.1.1.2, 02:25:53, FastEthernet4
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.1.0/24 is directly connected, Loopback1
L       172.16.1.1/32 is directly connected, Loopback1
C       172.16.100.0/24 is directly connected, Loopback100
L       172.16.100.1/32 is directly connected, Loopback100
```

From the routing table above (before redistribution), currently there are routes for the local networks emulated by loopbacks 1&100 and also two routes learned via OSPF for the networks behind ENT2. (The blue color routes above denoted by "O" are learned via OSPF).

Now, let's see the routing table of ENT2 before the redistribution

This router is the junction between the two networks and runs both OSPF and EIGRP. Therefore, its routing table contains the routes from both networks. However, we have not configured Redistribution yet.

ENT2#sh ip route

```
Codes: C – connected, D - EIGRP, O - OSPF,
    1.0.0.0/30 is subnetted, 1 subnets
C      1.1.1.0 is directly connected, FastEthernet1
```

2.0.0.0/30 is subnetted, 1 subnets

C   2.2.2.0 is directly connected, FastEthernet0

172.16.0.0/32 is subnetted, 2 subnets

O 172.16.1.1 [110/2] via 1.1.1.1, 02:38:35, FastEthernet1 **O**

172.16.100.1 [110/2] via 1.1.1.1, 02:40:11, FastEthernet1

10.0.0.0/24 is subnetted, 2 subnets

C 10.10.0.0 is directly connected, Loopback0

C 10.10.100.0 is directly connected, Loopback100

D 192.168.1.0/24 [90/156160] via 2.2.2.2, 00:00:06, FastEthernet0

D 192.168.100.0/24 [90/156160] via 2.2.2.2, 00:00:06, FastEthernet0

The blue color routes denoted by "**O**" are learned via OSPF (from the BLUE network) and are received from the neighbor router ENT1. The red color routes denoted by "**D**" are learned via EIGRP and are received from the RED network (from "Ent_Remote").

Let's examine the "ENT_Remote" routing table:

ENT_Remote#sh ip route

2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 2.2.2.0/30 is directly connected, GigabitEthernet0

L 2.2.2.2/32 is directly connected, GigabitEthernet0

10.0.0.0/24 is subnetted, 2 subnets

D 10.10.0.0 [90/156160] via 2.2.2.1, 00:06:11, GigabitEthernet0 **D 10.10.100.0**

[90/156160] via 2.2.2.1, 00:06:11, GigabitEthernet0

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.1.0/24 is directly connected, Loopback1

L 192.168.1.1/32 is directly connected, Loopback1

192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.100.0/24 is directly connected, Loopback100

L 192.168.100.1/32 is directly connected, Loopback100

The ENT_Remote device above has only EIGRP routes received from ENT2.

Our goal is to have a full converged network and every router needs to learn the routes to all the other networks. To achieve this, redistribution must be performed on ENT2 router that runs both protocols.

Redistribute OSPF into EIGRP

When redistributing routes from other IGP's into EIGRP you have to keep in mind that you **must specify the metric for the redistributed routes** which can be done in two ways:

1) Specify default metric for all the routes and issue redistribution command. All the routes that are redistributed will have that default metric.

ENT2 Router

router eigrp 1 < --- First go into EIGRP routing instance
redistribute ospf 1 < ----- Redistribute OSPF into EIGRP
network 2.2.2.0 0.0.0.3 network
10.10.0.0 0.0.0.255 network
10.10.100.0 0.0.0.255
default-metric 1000 33 255 1 1500 < -- Set the default metric for all redistributed routes no auto-summary

eigrp router-id 2.2.2.1

2) Set the metric per each redistribute command

ENT2 Router

router eigrp 1
redistribute ospf 1 metric 1000 33 255 1 1500 <- Set the metric for the specific OSPF 1 instance

network 2.2.2.0 0.0.0.3 network
10.10.0.0 0.0.0.255 network
10.10.100.0 0.0.0.255
no auto-summary
eigrp router-id 2.2.2.1

NOTE:
metric 1000 33 255 1 1500 - This command sets the K values that compose the metric

bandwidth to 1000 (Kbps), the delay to 33(tens-of-microseconds, or 330 microseconds), reliability to 255 (a value between 1–255 ,255 is best), load to 1 (a value between 1–255, 1 is best), and MTU of 1500.

The routing table on router "Ent_Remote" has the redistributed OSPF routes shown as **D EX** (EIGRP external) as shown below.

Ent_Remote#sh ip route eigrp

```
     1.0.0.0/30 is subnetted, 1 subnets
D EX    1.1.1.0 [170/2571008] via 2.2.2.1, 00:01:29, GigabitEthernet0
     10.0.0.0/24 is subnetted, 2 subnets
D       10.10.0.0 [90/156160] via 2.2.2.1, 00:01:29, GigabitEthernet0
D       10.10.100.0 [90/156160] via 2.2.2.1, 00:01:29, GigabitEthernet0
     172.16.0.0/32 is subnetted, 2 subnets
D EX    172.16.1.1 [170/2571008] via 2.2.2.1, 00:01:29, GigabitEthernet0
D EX    172.16.100.1 [170/2571008] via 2.2.2.1, 00:01:29, GigabitEthernet0
```

Redistribute EIGRP into OSPF

Basic redistribution into OSPF is pretty simple, just remember the word **subnets** because if you issue the redistribution command without the subnets statement you will only redistribute classful routes. Again the redistribution must be done on ENT2 router that has both protocols.

ENT2 Router

router ospf 1 < --- First go into OSPF routing instance router-id 1.1.1.2
log-adjacency-changes **redistribute eigrp 1 subnets** < ----- Redistribute EIGRP into OSPF

network 1.1.1.0 0.0 0.3 area 0
network 10.10.0.0 0.0 255.255 area 0

Now, the routing table on ENT1 shows the EIGRP redistributed routes as E2 with a metric of 20

ENT1#sh ip route ospf

O - OSPF, IA - OSPF inter area, E2 - OSPF external type 2

2.0.0.0/30 is subnetted, 1 subnets

O E2 2.2.2.0 [110/20] via 1.1.1.2, 00:00:08, FastEthernet4

10.0.0.0/32 is subnetted, 2 subnets

O 10.10.0.1 [110/2] via 1.1.1.2, 00:00:08, FastEthernet4

O 10.10.100.1 [110/2] via 1.1.1.2, 00:00:08, FastEthernet4

O E2 192.168.1.0/24 [110/20] via 1.1.1.2, 00:00:08, FastEthernet4

O E2 192.168.100.0/24 [110/20] via 1.1.1.2, 00:00:08, FastEthernet4

To verify that the remote EIGRP network is reachable you can issue a ping or a traceroute

ENT1#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4
ms ENT1#

ENT1#traceroute 192.168.1.1

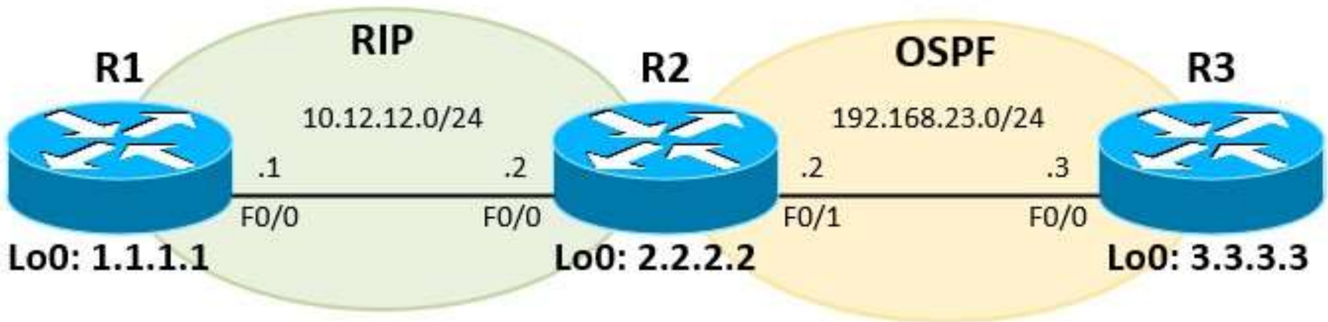
Type escape sequence to abort.
Tracing the route to
192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
1 1.1.1.2 0 msec 0 msec 0
msec
2 2.2.2.2 0 msec 0 msec *
192.168.1.1 0 msec 0 msec

ENT1#sh ip route 192.168.1.1

Routing entry for 192.168.1.0/24
Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric
1 Last update from 1.1.1.2 on FastEthernet4, 00:07:23 ago Routing
Descriptor Blocks:
* 1.1.1.2, from 1.1.1.2, 00:07:23 ago, via FastEthernet4 Route
metric is 20, traffic share count is 1

How to Configure Redistribution between RIP and OSPF in Cisco IOS Router

Redistribution can be performed **between RIP and OSPF**. We will use a scenario between three routers (**R1, R2, and R3**) to demonstrate how to Configure Redistribution between RIP and OSPF in Cisco IOS Router.



In the topology above, **RIP is used to connect R1-R2** and **OSPF is used to connect R2-R3**. In this scenario we have an issue where **R1 cannot communicate to R3 and vice versa**, despite the intermediate router (in this case is R2) knows exactly how to reach both network.

The following snippets are the routing configuration for each router in the topology above:

```
R1#sh run | s router

router rip

 version 2

 network 1.0.0.0

 network 10.0.0.0

 no auto-summary

R2#sh run | s router

router ospf 1

 network 2.2.2.2 0.0.0.0 area 0

 network 192.168.23.2 0.0.0.0 area 0

router rip

 version 2

 network 10.0.0.0

 no auto-summary

R3#sh run | s router
```



```
router ospf 1

 network 3.3.3.3 0.0.0.0 area 0

 network 192.168.23.3 0.0.0.0 area 0
```

To solve this kind of issue, a **2-way redistribution** must be done on the **intermediate router**, or in this case is R2. The command used to **redistribute RIP route into OSPF** is "**redistribute rip subnets**" and the implementation shown below:

```
R2(config)#router ospf 1

R2(config-router)#redistribute rip subnets
```

Now the command to **redistribute OSPF route into RIP** is "**redistribute ospf [PID] metric [hop count]**". Notice that when we redistribute any external route into RIP, we have to specify the **hop count** to reach that external network because RIP is a protocol that works based on the hop count. The hop count doesn't have to be exactly match the real condition but it could be adjusted according to our needs. In this example, we set the hop count to **12** for easy identification purpose:

```
R2(config)#router rip

R2(config-router)#redistribute ospf 1 metric 12
```

After adding the above command, notice the change in R1 and R3 routing table:

Before

```
R1#sh ip route | b Gate

Gateway of last resort is not set



     1.0.0.0/32 is subnetted, 1 subnets

C       1.1.1.1 is directly connected, Loopback0

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

C       10.12.12.0/24 is directly connected, FastEthernet0/0

L       10.12.12.1/32 is directly connected, FastEthernet0/0

R3#sh ip route | b Gate

Gateway of last resort is not set



     2.0.0.0/32 is subnetted, 1 subnets

O       2.2.2.2 [110/2] via 192.168.23.2, 00:02:14, FastEthernet0/0
```



```
     3.0.0.0/32 is subnetted, 1 subnets

C       3.3.3.3 is directly connected, Loopback0

     192.168.23.0/24 is variably subnetted, 2 subnets, 2 masks

C       192.168.23.0/24 is directly connected, FastEthernet0/0 L      192.168.23.3/32

is directly connected, FastEthernet0/0
```

After

```
R1#sh ip route | b Gate

Gateway of last resort is not set



     1.0.0.0/32 is subnetted, 1 subnets

C       1.1.1.1 is directly connected, Loopback0

     2.0.0.0/32 is subnetted, 1 subnets

R       2.2.2.2 [120/12] via 10.12.12.2, 00:00:06, FastEthernet0/0      3.0.0.0/32 is

subnetted, 1 subnets

R       3.3.3.3 [120/12] via 10.12.12.2, 00:00:06, FastEthernet0/0

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

C       10.12.12.0/24 is directly connected, FastEthernet0/0

L       10.12.12.1/32 is directly connected, FastEthernet0/0

R       192.168.23.0/24 [120/12] via 10.12.12.2, 00:00:06, FastEthernet0/0

R3#sh ip route | b Gate

Gateway of last resort is not set



     1.0.0.0/32 is subnetted, 1 subnets

O E2    1.1.1.1 [110/20] via 192.168.23.2, 00:00:17, FastEthernet0/0
```



```
     2.0.0.0/32 is subnetted, 1 subnets

O       2.2.2.2 [110/2] via 192.168.23.2, 00:02:47, FastEthernet0/0

     3.0.0.0/32 is subnetted, 1 subnets
```

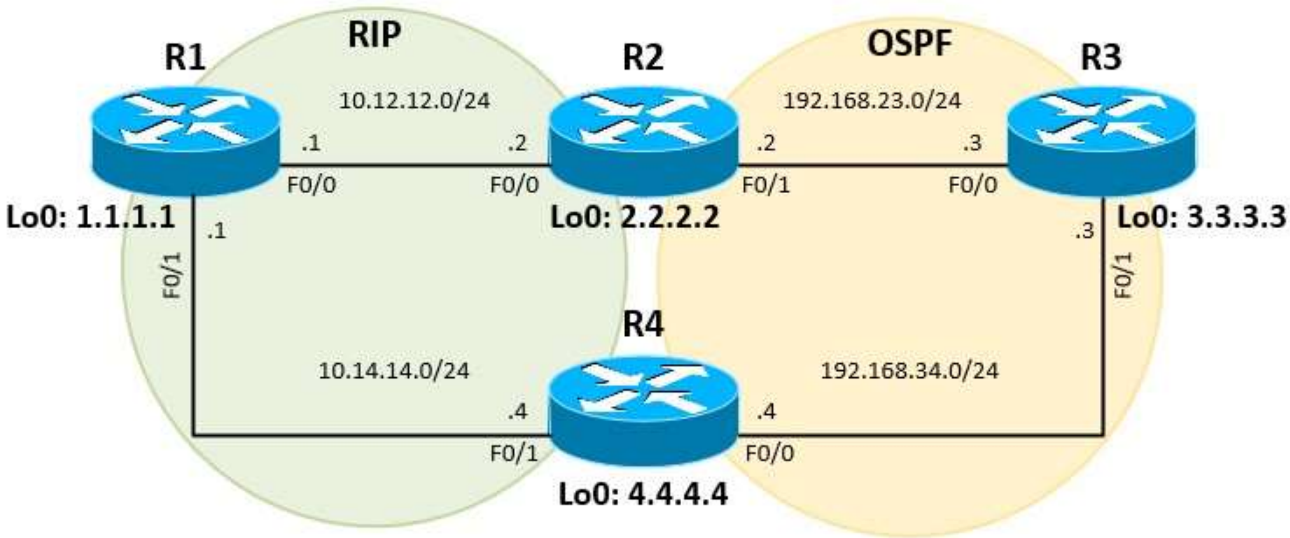
```
C      3.3.3.3 is directly connected, Loopback0

     10.0.0.0/24 is subnetted, 1 subnets

O E2   10.12.12.0 [110/20] via 192.168.23.2, 00:00:17, FastEthernet0/0

     192.168.23.0/24 is variably subnetted, 2 subnets, 2 masks

C      192.168.23.0/24 is directly connected, FastEthernet0/0

L      192.168.23.3/32 is directly connected, FastEthernet0/0
```

As you can see the route to **3.3.3.3 now has been inserted to R1's routing table** with hop count 12 as specified, and route to **1.1.1.1 has been inserted to R3's routing table** as an OSPF external route. With this, it means we have successfully configured redistribution between RIP and OSPF, and ping between R1 and R3 can now be established

```
R1#ping 3.3.3.3 sour 1.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/48/68 ms
```

Issue in Redistribution between RIP and OSPF

When we configure redistribution between RIP and OSPF in Cisco IOS router, the problem that commonly occurs is **suboptimal routing** a.k.a condition where router does not choose the shortest path to reach a destination. In a worse situation, this could also resulting in **infinite routing loop**. Suboptimal routing usually happens when there are **more than one router performing redistribution**, for example in a highavailability scenario. To demonstrate the problem, **R4** is added to the above same topology.



RIP is configured between R1-R4 and **OSPF configured between R3-R4**. Also, **2-way redistribution is performed on R4** as well with the configuration shown below:

```
R4#sh run | s router

router ospf 1

 redistribute rip subnets

 network 4.4.4.4 0.0.0.0 area 0

 network 192.168.34.4 0.0.0.0 area 0

router rip

 version 2

 redistribute ospf 1 metric 12

 network 10.0.0.0

 no auto-summary
```

In our case, **suboptimal routing happens on R4 for route to R1**. See R4 routing table and trace result below:

```
R4#show ip route 1.1.1.1

Routing entry for 1.1.1.1/32

 Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric 2
```

```
 Redistributing via rip

 Advertised by rip metric 12

 Last update from 192.168.34.3 on FastEthernet0/0, 00:08:53 ago  Routing Descriptor

Blocks:

 * 192.168.34.3, from 2.2.2.2, 00:08:53 ago, via FastEthernet0/0

   Route metric is 20, traffic share count is 1

R4#trace 1.1.1.1 sour 4.4.4.4

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

1 192.168.34.3 20 msec 12 msec 24 msec

2 192.168.23.2 40 msec 32 msec 52 msec

3 10.12.12.1 48 msec 56 msec 60 msec
```


It can be seen that **R4 choose the route to R1 via R3 then R2**, despite that **R4 can actually reach R1 directly**. This is because R4 receive route to R1 that has been redistributed through OSPF in R2-R3, and R4 choose to trusts this route more than the RIP route that it already knows because **OSPF route has lower AD value (110) than RIP (120)**.

Using Distribute-List to Prevent Redistribution Issue

To prevent the above issue from happening, we can use a configuration called **distribute-list**. A distributelist can be used to **control incoming route information on an interface** or control an **outgoing route information from an interface**. The command to use distribute-list is "**distribute-list [access-list_ID] [in/out] [interface_name]**", and as can be seen on this command that a distribute-list command needs an **access-list** as a reference.

In our example, what we are going to do is **preventing R4 to receive incoming route information to 1.1.1.1/32 on its OSPF-connected interface** so when RIP is redistributed to OSPF on R2 and the information forwarded to R4, R4 will not consider this information but will trust its own RIP routing table instead. In reverse, we also need to **do the same on R2** so that it will not consider the RIP route that is redistributed to OSPF on R4.

So we will first **create the access-list** on both R2 and R4:

```
R2(config)#access-list 10 permit any

R4(config)#access-list 10 deny 1.1.1.1 0.0.0.0

R4(config)#access-list 10 permit any
```

Then refer the above access-list on both R2 and R4 OSPF config using the **distribute-list** command:

```
R2(config)#access-list 10 deny 1.1.1.1 0.0.0.0
```



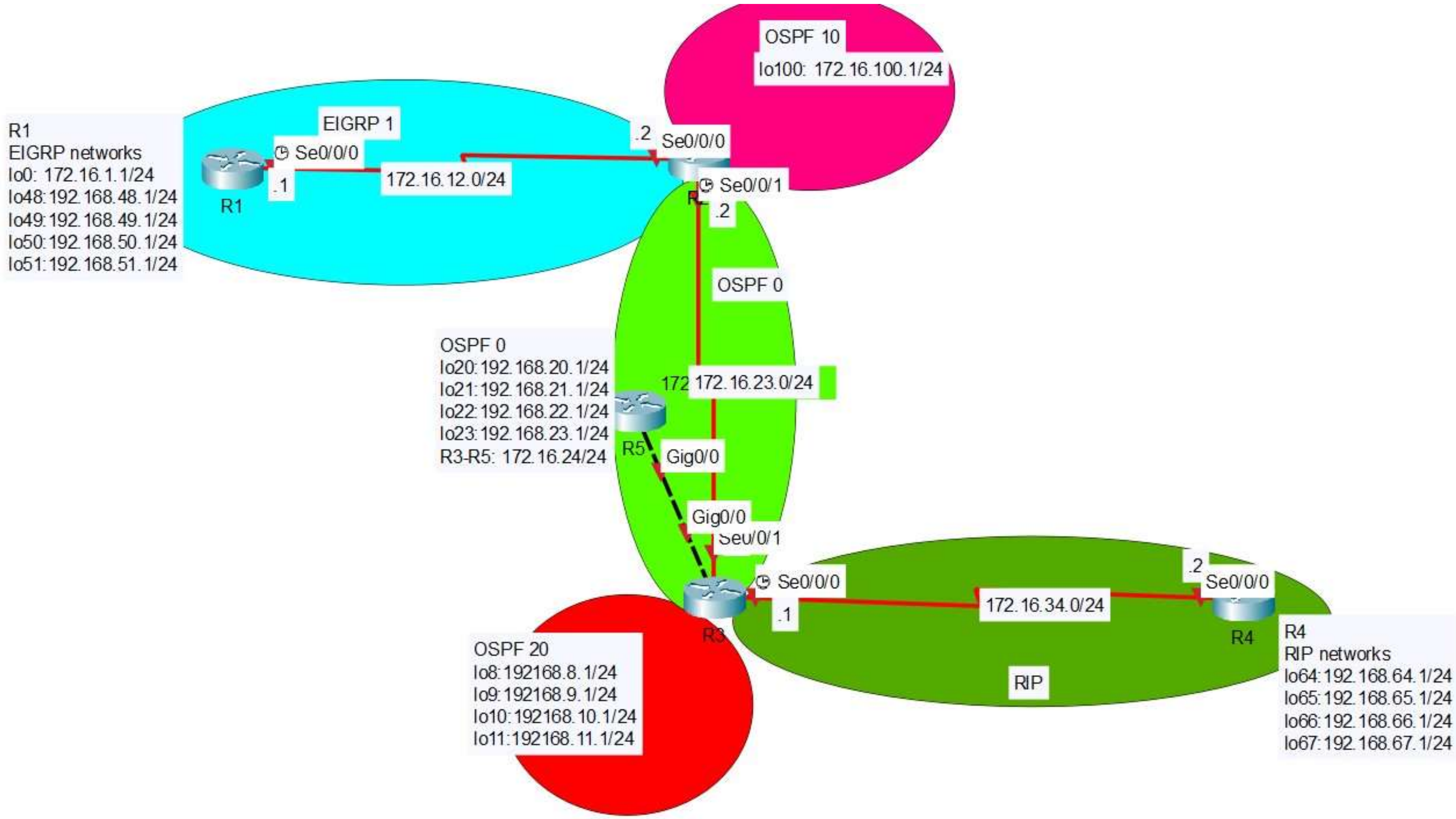
```
R2(config)#router ospf 1

R2(config-router)#distribute-list 10 in f0/1

R4(config)#router ospf 1

R4(config-router)#distribute-list 10 in f0/0
```

***f0/1 is the OSPF-connected interface on R2 while f0/0 is the OSPF-connected interface on R4.**

With this command we have successfully eliminate the suboptimal routing issue, as can be seen in the output below that R4 now choose the RIP route to reach R1, and this means **R4 can now reach R1 directly without having to go through R3-R2**.

```
R4(config)#do sh ip route 1.1.1.1

Routing entry for 1.1.1.1/32

 Known via "rip", distance 120, metric 1

 Redistributing via rip, ospf 1

 Advertised by ospf 1 subnets

 Last update from 10.14.14.1 on FastEthernet0/1, 00:00:14 ago

 Routing Descriptor Blocks:

 * 10.14.14.1, from 10.14.14.1, 00:00:14 ago, via FastEthernet0/1

   Route metric is 1, traffic share count is 1
```

And that's all the basic you need to know regarding how to configure redistribution between RIP and OSPF in Cisco IOS router.



OSPF 10
lo100: 172.16.100.1/24
R1
EIGRP networks
lo0: 172.16.1.1/24
lo48:192.168.48.1/24
lo49:192.168.49.1/24
lo50:192.168.50.1/24
lo51:192.168.51.1/24
EIGRP 1
Se0/0/0
.1
172.16.12.0/24
.2
Se0/0/0
Se0/0/1
.2
R1
OSPF 0
OSPF 0
lo20:192.168.20.1/24
lo21:192.168.21.1/24
lo22:192.168.22.1/24
lo23:192.168.23.1/24
R3-R5: 172.16.24/24
172
172.16.23.0/24
R5
Gig0/0
Gig0/0
Se0/0/1
Se0/0/0
.1
172.16.34.0/24
.2
Se0/0/0
R3
R4
OSPF 20
lo8:192168.8.1/24
lo9:192168.9.1/24
lo10:192168.10.1/24
lo11:192168.11.1/24
RIP
R4
RIP networks
lo64:192.168.64.1/24
lo65:192.168.65.1/24
lo66:192.168.66.1/24
lo67:192.168.67.1/24