Redistributing Routing Protocols

Document ID: 8606

Contents

Introduction

The use of a routing protocol to advertise routes that are learned by some other means, such as by another routing protocol, static routes, or directly connected routes, is called redistribution. While running a single routing protocol throughout your entire IP internetwork is desirable, multi−protocol routing is common for a number of reasons, such as company mergers, multiple departments managed by multiple network administrators, and multi−vendor environments. Running different routing protocols is often part of a network design. In any case, having a multiple protocol environment makes redistribution a necessity.

Differences in routing protocol characteristics, such as metrics, administrative distance, classful and classless capabilities can effect redistribution. Consideration must be given to these differences for redistribution to succeed.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco IOS® Software Release 12.2(10b)
- Cisco 2500 Series Routers

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Metrics

When you redistribute one protocol into another, remember that the metrics of each protocol play an important role in redistribution. Each protocol uses different metrics. For example, the Routing Information Protocol (RIP) metric is based on hop count, but Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) use a composite metric based on bandwidth, delay, reliability, load, and maximum transmission unit (MTU), where bandwidth and delay are the only parameters used by default. When routes are redistributed, you must define a metric that is understandable to the receiving protocol. There are two methods to define metrics when redistributing routes.

You can define the metric for that specific redistribution only:

```
router rip
redistribute static metric 1
redistribute ospf 1 metric 1
```

Or you can use the same metric as a default for all redistribution (Using the **default−metric** command saves work because it eliminates the need for defining the metric separately for each redistribution.):

```
router rip
redistribute static
redistribute ospf 1
default−metric 1
```

Administrative Distance

If a router is running more than one routing protocol and learns a route to the same destination using both routing protocols, then which route should be selected as the best route? Each protocol uses its own metric type to determine the best route. Comparing routes with different metric types cannot be done. Administrative distances take care of this problem. Administrative distances are assigned to route sources so that the route from the most preferred source will be chosen as the best path. Refer to Route Selection in Cisco Routers for more information about administrative distances and route selection.

Administrative distances help with route selection among different routing protocols, but they can cause problems for redistribution. These problems can be in the form of routing loops, convergence problems, or inefficient routing. See below for a topology and description of a possible problem.
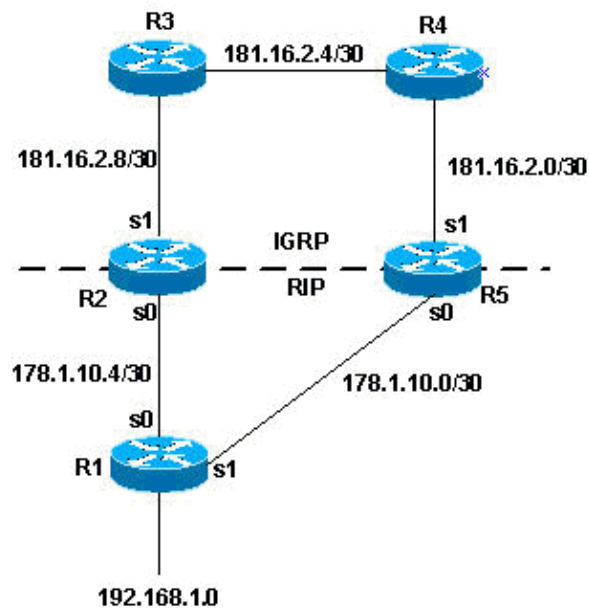
In the above topology, if R1 is running RIP, and R2 and R5 are running both RIP and IGRP and redistributing RIP into IGRP, then there is a potential problem. For example, R2 and R5 are both learning about network 192.168.1.0 from R1 using RIP. This knowledge is redistributed into IGRP. R2 learns about network 192.168.1.0 through R3, and R5 learns about it from R4 using IGRP. IGRP has a lower administrative distance than RIP (100 versus 120); therefore, the IGRP route is what is used in the routing table. Now there is a potential routing loop. Even if split horizon, or any other feature meant to help prevent routing loops comes into play, there is still a convergence problem.

If R2 and R5 are also redistributing IGRP into RIP (otherwise known as mutual redistribution) and the network, 192.168.1.0, is not directly connected to R1 (R1 is learning from another router upstream from it), then there is a potential problem that R1 will learn the network from R2 or R5 with a better metric than from the original source.

Note: The mechanics of route redistribution is proprietary on Cisco routers. The rules for redistribution on a Cisco router dictate that the redistributed route be present in the routing table. It is not sufficient that the route be present in the routing topology or database. Routes with a lower Administrative Distance (AD) are always installed in the routing table. For example, if a static route is redistributed into IGRP on R5, and then IGRP subsequently redistributed into RIP on the same router (R5), the static route is not redistributed into RIP because it never got entered into the IGRP routing table. This is due to the fact that static routes have an AD of 1 and IGRP routes have an AD of 100 and the static route is installed in the routing table. In order to redistribute the static route into IGRP on R5, you need to use the **redistribute static** command under the **router rip** command.

The default behavior for RIP, IGRP and EIGRP is to advertise directly connected routes when a **network** statement under the routing protocol includes the connected interface subnet. There are two methods to get a connected route:

- An interface is configured with an IP address and mask, this corresponding subnet is considered a connected route.
- A static route is configured with only an outgoing interface, and not an IP next–hop, this is also considered a connected route.

```
Router#conf t
   Router(config)#ip route 10.0.77.0 255.255.255.0 ethernet 0/0
   Router(config)#end
```

```
Router#show ip route static
10.0.0.0/24 is subnetted, 1 subnets
S 10.0.77.0 is directly connected, Ethernet0/0
```

A **network** command configured under EIGRP, RIP or IGRP that includes (or "covers") either of these types of connected routes includes that subnet for advertisement.

For example, if an interface has address 10.0.23.1 and mask 255.255.255.0, the subnet 10.0.23.0/24 is a connected route and will be advertised by these routing protocols when a **network** statement is configured as follows:

```
router rip | igrp # | eigrp #
    network 10.0.0.0
```

This static route, 10.0.77.0/24, is also advertised by these routing protocols, because it is a connected route and it is "covered" by the **network** statement.

See the Avoiding Problems Due to Redistribution section of this document for tips on how to avoid this problem.

Redistribution Configuration Syntax and Examples

IGRP and EIGRP

This output shows an IGRP/EIGRP router redistributing static, Open Shortest Path First (OSPF), RIP, and Intermediate System−to−Intermediate System (IS−IS) routes.

```
router igrp/eigrp 1
network 131.108.0.0
redistribute static
redistribute ospf 1
redistribute rip
redistribute isis
default−metric 10000 100 255 1 1500
```

IGRP and EIGRP need five metrics when redistributing other protocols: bandwidth, delay, reliability, load, and MTU, respectively. An example of IGRP metrics follows:

Metric	Value
bandwidth	In units of kilobits per second; 10000 for Ethernet
delay	In units of tens of microseconds; for Ethernet it is 100 x 10 microseconds = 1 ms
reliability	255 for 100 percent reliability
load	Effective load on the link expressed as a number from 0 to 255 (255 is 100 percent loading)
MTU	Minimum MTU of the path; usually equals that for the Ethernet interface, which is 1500 bytes

Multiple IGRP and EIGRP processes can run on the same router, with redistribution between them. For example, IGRP1 and IGRP2 can run on the same router. However, running two processes of the same protocol on the same router is rarely necessary, and can consume the router's memory and CPU.

The redistribution of IGRP/EIGRP into another IGRP/EIGRP process does not require any metric conversion, so there is no need to define metrics or use the **default−metric** command during redistribution.

A redistributed static route takes precedence over the summary route because the static route has an administrative distance of 1 whereas Eigrp summary route has an administrative distance of 5. This happens when a static route is redistributed with the use of **redistribute static** under the Eigrp process and the Eigrp process has a default route.

OSPF

This output shows an OSPF router redistributing static, RIP, IGRP, EIGRP, and IS−IS routes.

```
router ospf 1
network 131.108.0.0 0.0.255.255 area 0
redistribute static metric 200 subnets
redistribute rip metric 200 subnets
redistribute igrp 1 metric 100 subnets
redistribute eigrp 1 metric 100 subnets
redistribute isis metric 10 subnets
```

The OSPF metric is a cost value based on 10^8/ bandwidth of the link in bits/sec. For example, the OSPF cost of Ethernet is 10: $10^8/10^7 = 10$

Note: If a metric is not specified, OSPF puts a default value of 20 when redistributing routes from all protocols except Border Gateway Protocol (BGP) routes, which get a metric of 1.

When there is a major net that is subnetted, you need to use the keyword subnet to redistribute protocols into OSPF. Without this keyword, OSPF only redistributes major nets that are not subnetted.

It is possible to run more than one OSPF process on the same router. However, running more than one process of the same protocol is rarely needed, and consumes the router's memory and CPU.

You do not need to define metric or use the **default−metric** command when redistributing one OSPF process into another.

RIP

Note: The principles in this document apply to RIP versions I and II.

This output shows a RIP router redistributing static, IGRP, EIGRP, OSPF, and IS−IS routes.

```
router rip
network 131.108.0.0
redistribute static
redistribute igrp 1
redistribute eigrp 1
redistribute ospf 1
redistribute isis
default−metric 1
```

The RIP metric is composed of hop count, and the maximum valid metric is 15. Anything above 15 is considered infinite; you can use 16 to describe an infinite metric in RIP. When redistributing a protocol into RIP, Cisco recommends that you use a low metric, such as 1. A high metric, such as 10, limits RIP even further. If you define a metric of 10 for redistributed routes, these routes can only be advertised to routers up to 5 hops away, at which point the metric (hop count) exceeds 15. By defining a metric of 1, you enable a route to travel the maximum number of hops in a RIP domain. But, doing this increases the possibility of

routing loops if there are multiple redistribution points and a router learns about the network with a better metric from the redistribution point than from the original source, as explained in the Administrative Distance section of this document. Therefore, you have to make sure that the metric is neither too high, preventing it from being advertised to all the routers, or too low, leading to routing loops when there are multiple redistribution points.

Redistributing Static Routes Except Gateway of Last resort in RIP using Route Map

This configuration is an example of redistributing static routes except gateway of last gateway resort in RIP through routemap.

Initial Configuration for this example:

```
router rip

version 2

network 10.0.0.0

default-information originate

no auto-summary

!

ip forward-protocol nd

ip route 0.0.0.0 0.0.0.0 10.32.32.3

ip route 10.32.42.211 255.255.255.255 192.192.192.102

ip route 10.98.0.0 255.255.255.0 10.32.32.1

ip route 10.99.0.0 255.255.255.0 10.32.32.1

ip route 10.99.99.0 255.255.255.252 10.32.32.5

ip route 67.129.103.128 255.255.255.240 10.32.31.1

ip route 156.55.231.0 255.255.255.0 10.32.32.5

ip route 172.16.28.0 255.255.252.0 10.32.32.5

ip route 192.168.248.0 255.255.255.0 10.32.32.5

ip route 199.43.0.0 255.255.255.0 10.32.32.5

ip route 204.103.0.0 255.255.255.0 10.32.32.5
```

Complete these steps in order to configure this:

1. Create an access−list in order to match all networks that needs to be redistributed

```
Router#show access-lists 10

Standard IP access list 10

    10 permit 10.32.42.211

    20 permit 10.98.0.0, wildcard bits 0.0.0.255
```

```
30 permit 10.99.0.0, wildcard bits 0.0.0.255

40 permit 67.129.103.128, wildcard bits 0.0.0.15

50 permit 156.55.231.0, wildcard bits 0.0.0.255

60 permit 172.16.28.0, wildcard bits 0.0.3.255

70 permit 192.168.248.0, wildcard bits 0.0.0.255

80 permit 199.43.0.0, wildcard bits 0.0.0.255

90 permit 204.103.0.0, wildcard bits 0.0.0.255
```

2. Call this access–list in a route–map.

```
Route-map TEST

 Match ip address 10
```

3. Redistribute in RIP using route–map at and remove the **default information originate** command from the rip process.

```
Router RIP

version 2

network 10.0.0.0

redistribute static route-map TEST

no auto-summary
```

IS–IS

This output shows an IS–IS router redistributing static, RIP, IGRP, EIGRP, and OSPF routes.

```
router isis
network 49.1234.1111.1111.1111.00
redistribute static
redistribute rip metric 20
redistribute igrp 1 metric 20
redistribute eigrp 1 metric 20
redistribute ospf 1 metric 20
```

The IS–IS metric must be between 1 and 63. There is no default–metric option in IS–IS you should define a metric for each protocol, as shown in the example above. If no metric is specified for the routes being redistributed into IS–IS, a metric value of 0 is used by default.

Connected Routes

Redistributing directly connected networks into routing protocols is not a common practice and is not shown in any of the examples above for this reason. However, it is important to note that it can be done, both directly and indirectly. In order to directly redistribute connected routes, use the **redistribute connected** router configuration command. You should also define a metric in this case. You can also indirectly redistribute connected routes into routing protocols as shown in this example.
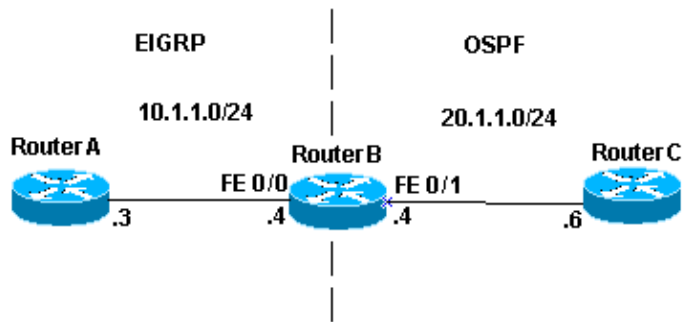
In this example, Router B has two Fast Ethernet interfaces. FastEthernet 0/0 is in network 10.1.1.0/24 and FastEthernet 0/1 is in network 20.1.1.0/24. Router B is running EIGRP with Router A, and OSPF with Router C. Router B is mutually redistributing between the EIGRP and OSPF processes. This is the pertinent configuration information for Router B:

Router B
| ```
interface FastEthernet0/0
 ip address 10.1.1.4 255.255.255.0

interface FastEthernet0/1
 ip address 20.1.1.4 255.255.255.0

router eigrp 7
 redistribute ospf 7 metric 10000 100 255 1 1500
 network 10.1.1.0 0.0.0.255
 auto-summary
 no eigrp log-neighbor-changes
!
router ospf 7
 log-adjacency-changes
 redistribute eigrp 7 subnets
 network 20.1.1.0 0.0.0.255 area 0
``` |

If you look at the routing table for Router B, you see the following:

```
routerB#show ip route
Codes: C – connected, S – static, I – IGRP, 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, E – EGP
       i – IS-IS, 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

Gateway of last resort is not set

     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
```

From the configuration and the routing table above there are three things to notice:

- The networks in question are in Router B routing table as directly connected networks.
- Network 10.1.1.0/24 is part of the EIGRP process and network 20.1.1.0/24 is part of the OSPF process.
- Router B is mutually redistributing between EIGRP and OSPF.

Below are the routing tables for Routers A and C.

```
routerA#show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0
     20.0.0.0/24 is subnetted, 1 subnets
D EX    20.1.1.0 [170/284160] via 10.1.1.4, 00:07:26, FastEthernet0


routerC#show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is not set

     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet1
O E2    10.1.1.0 [110/20] via 20.1.1.4, 00:07:32, FastEthernet1
```

Router A has learned about network 20.1.1.0/24 via EIGRP, which is shown as an external route, because it was redistributed from OSPF into EIGRP. Router C has learned about network 10.1.1.0/24 via OSPF as an external route, because it was redistributed from EIGRP into OSPF. Although Router B is not redistributing connected networks, it does advertise the network 10.1.1.0/24, which is part of the EIGRP process redistributed into OSPF. Similarly, Router B advertises network 20.1.1.0/24, which is part of the OSPF process redistributed into EIGRP.

Refer to Redistributing Connected Networks into OSPF for more information about connected routes being redistributed into OSPF.

Note: By default, only EBGP–learned information is candidate to be redistributed into IGP when the **redistibute bgp** command is issued. The IBGP routes is not redistributed into IGP until the **bgp redistribute–internal** command is configured under the **router bgp** command. But precautions must be taken in order to avoid loops within the Autonomous System when IBGP routes are redistirbuted into IGP.

Avoiding Problems Due to Redistribution

In the section on administrative distance you saw how redistribution can potentially cause problems such as below optimal routing, routing loops, or slow convergence. Avoiding these types of problems is really quite simple never announce the information originally received from routing process X back into routing process X.

Example 1

In the previous topology, R2 and R5 are doing mutual redistribution. RIP is being redistributed into IGRP and IGRP is being redistributing into RIP, as this configuration shows.

R2:

```
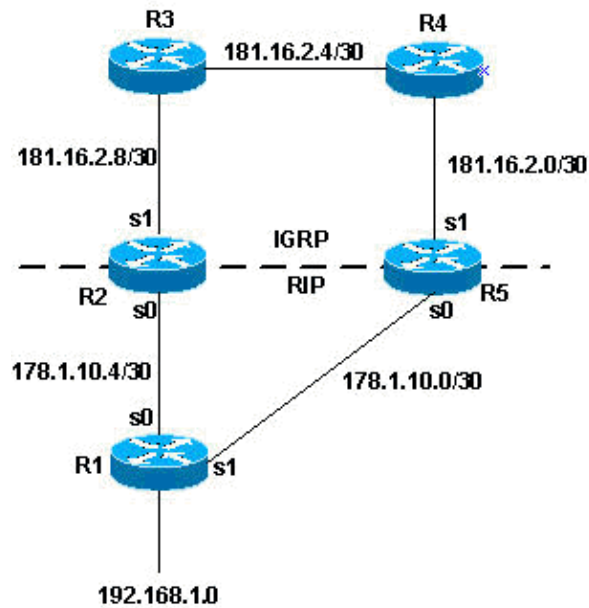router igrp 7
network 181.16.0.0

redistribute rip metric 1 1 1 1 1

router rip
network 178.1.0.0
redistribute igrp 7 metric 2
```

R5:

```
router igrp 7
network 181.16.0.0

redistribute rip metric 1 1 1 1 1

router rip
network 178.1.0.0
redistribute igrp 7 metric 2
```

With the previous configuration you have the potential for any the the problems previously described. In order to avoid them, you can filter routing updates as follows:

R2:

```
router igrp 7
network 181.16.0.0

redistribute rip metric 1 1 1 1 1
distribute-list 1 in s1

router rip
network 178.1.0.0
redistribute igrp 7 metric 2
```

```
        access-list 1 deny 192.168.1.0
        access-list 1 permit any
```

R5:

```
        router igrp 7
        network 181.16.0.0

        redistribute rip metric 1 1 1 1 1
        distribute-list 1 in s1

        router rip
        network 178.1.0.0

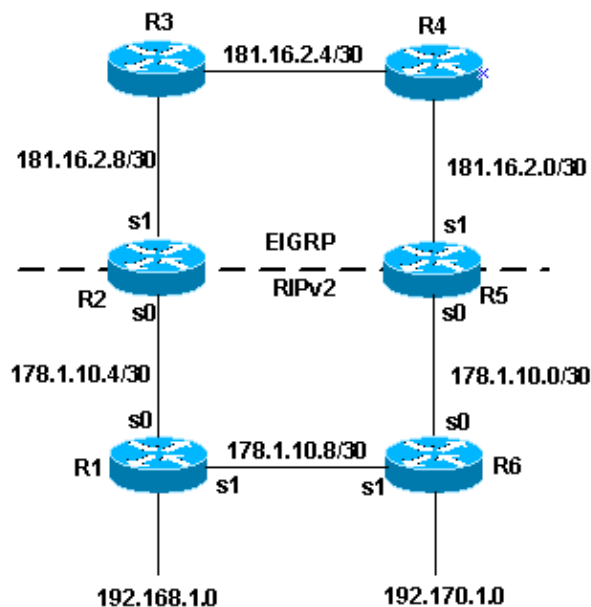        redistribute igrp 7 metric 2

        access-list 1 deny 192.168.1.0
        access-list 1 permit any
```

The distribute lists added to the configurations, as shown above, filter any IGRP updates that come into the serial 1 interface of the routers. If the routes in the updates are permitted by access list 1, the router accepts them in the update; otherwise it does not. In this example, the routers are being told that they should not learn network 192.168.1.0 through the IGRP updates they receive on their serial 1 interface. Therefore, the only knowledge these routers have for network 192.168.1.0 is through RIP from R1.

Also keep in mind that in this case it is not necessary to use the same filter strategy for the RIP process because RIP has a higher administrative distance than IGRP. If routes that originate in the IGRP domain were fed back to R2 and R5 through RIP, the IGRP routes would still take precedence.

Example 2

Using the topology as above, another method, which is sometimes more preferable, to avoid redistribution problems can be demonstrated. This method uses route–maps to set tags for various routes. Routing processes can then redistribute based on the tags. Note that redistribution based on tags do not work with RIP version 1 or IGRP.

One of the problems you can run into in the previous topology is as follows:

R1 advertises network 192.168.1.0 to R2. R2 then redistributes to EIGRP. R5 learns the network via EIGRP and redistributes it to RIPv2. Depending on the metric that R5 sets for the RIPv2 route, R6 might prefer the less desirable route through R5 instead of through R1 to reach the network. The following configuration helps to prevent this by setting tags and then redistributing based on the tags.

R2:

```
router eigrp 7
network 181.16.0.0
redistribute rip route-map rip_to_eigrp metric 1 1 1 1 1

!--- Redistributes RIP routes that are


!--- permitted by the route-map rip_to_eigrp




router rip
version 2
network 178.1.0.0
redistribute eigrp 7 route-map eigrp_to_rip metric 2

!--- Redistributes EIGRP routes and set the tags


!--- according to the eigrp_to_rip route-map



route-map rip_to_eigrp deny 10
match tag 88

!--- Route-map statement to deny any routes that have a tag of "88"


!--- from being redistributed into EIGRP


!--- Notice the routes tagged with "88" should be the EIGRP


!--- routes that are redistributed into RIPv2


route-map rip_to_eigrp permit 20
set tag 77

!--- Route-map statement to set the tag


!--- on RIPv2 routes redistributed into EIGRP to "77"


route-map eigrp_to_rip deny 10
match tag 77

!--- Route-map statement to deny any routes that have a


!--- tag of "77" from being redistributed into RIPv2


!--- Notice the routes tagged with "77" should be the RIPv2
```

```
route-map eigrp_to_rip permit 20
set tag 88
```

*!--- **Route-map** statement to set the tag on EIGRP*

!--- routes redistributed into RIPv2 to "88"

R5:

```
router eigrp 7
network 181.16.0.0
redistribute rip route-map rip_to_eigrp metric 1 1 1 1 1
```

!--- Redistributes RIPv2 routes that are permitted

*!--- by the **route-map rip_to_eigrp***

```
router rip
version 2
network 178.1.0.0
redistribute eigrp 7 route-map eigrp_to_rip metric 2
```

!--- Redistributes EIGRP routes and sets the tags

*!--- according to the **eigrp_to_rip route-map***

```
route-map rip_to_eigrp deny 10
match tag 88
```

*!--- **Route-map** statement to deny any routes that have a tag*

!--- of "88" from being redistributed into EIGRP

!--- Notice the routes tagged with "88" should be the EIGRP routes

!--- that are redistributed into RIPv2

```
route-map rip_to_eigrp permit 20
set tag 77
```

*!--- **Route-map** statement to set the tag on rip routes*

!--- redistributed into EIGRP to "77"

```
route-map eigrp_to_rip deny 10
match tag 77
```

!--- **Route-map** *statement to deny any routes that have a tag*

!--- *of "77" from being redistributed into RIPv2*

!--- *Notice the routes tagged with "77" should be the RIPv2 routes*

!--- *that are redistributed into EIGRP*

```
route-map eigrp_to_rip permit 20
set tag 88
```

!--- **Route-map** *statement to set the tag on EIGRP routes*

!--- *redistributed into RIPv2 to "88"*

With the above configuration performed, you can look at some specific routes in the routing table to see that the tags have been set. Below shows output from the **show ip route** command for specific routes on R3 and R1:

```
R3#show ip route 178.1.10.8
Routing entry for 178.1.10.8/30
  Known via "eigrp 7", distance 170, metric 2560512256
  Tag 77, type external
  Redistributing via eigrp 7
  Last update from 181.16.2.10 on Serial0, 00:07:22 ago
  Routing Descriptor Blocks:
  * 181.16.2.10, from 181.16.2.10, 00:07:22 ago, via Serial0
      Route metric is 2560512256, traffic share count is 1
      Total delay is 20010 microseconds, minimum bandwidth is 1 Kbit
      Reliability 1/255, minimum MTU 1 bytes
      Loading 1/255, Hops 1


R1#show ip route 181.16.2.4
Routing entry for 181.16.0.0/16
  Known via "rip", distance 120, metric 2
  Tag 88
  Redistributing via rip
  Last update from 178.1.10.5 on Serial0, 00:00:15 ago
  Routing Descriptor Blocks:
  * 178.1.10.5, from 178.1.10.5, 00:00:15 ago, via Serial0
      Route metric is 2, traffic share count is 1
```

EIGRP uses five different variables to calculate the metric. However, redistributed routes do not have these parameters, which causes routes to not be set uniformly. The best practice is to set a default–metric when redistributing routes. By setting the default metric, the performance of EIGRP can be improved. For EIGRP, the default values are entered with this command:

```
Router(config-router)#default-metric 10000 100 255 100 1500
```

Example 3

Redistribution can also take place among different processes of the same routing protocol. The following

configuration is an example of a redistribution policy used for redistributing two EIGRP process running on the same router or on multiple routers:

```
router eigrp 3
 redistribute eigrp 5 route-map to_eigrp_3
 default-metric 10000 100 255 1 1500

!--- Redistributes EIGRP 5 into EIGRP 3, setting the tags


!--- according to the route map "to_eigrp_3"



router eigrp 5
 redistribute eigrp 3 route-map to_eigrp_5
 default-metric 10000 100 255 1 1500

!--- Redistributes EIGRP 3 into EIGRP 5


!--- Routes with tag 33 will not be redistributed


!--- due to route map "to_eigrp_5"
!--- Though the default-metric command is not required
!--- when redistributing between different EIGRP processes,
!--- you can use it optionally as shown above to advertise
!--- the routes with specific values for calculating the metric.


route-map to_eigrp_3 deny 10
match tag 55

!--- Route-map statement used to deny any routes that have a tag


!--- of "55" from being redistributed into EIGRP 3


!--- Notice the routes tagged with "55" should be the EIGRP 3 routes


!--- that are redistributed into EIGRP 5


route-map to_eigrp_3 permit 20
 set tag 33

!--- Route-map statement used to set the tag on routes


!--- redistributed from EIGRP 5 to EIGRP 3 to "33"


route-map to_eigrp_5 deny 10
 match tag 33

!--- Route-map statement used to deny any routes that have a tag


!--- of "33" from being redistributed into EIGRP 5


!--- Notice the routes tagged with "33" should be the EIGRP 5 routes
```

```
                !--- that are redistributed into EIGRP 3


                route-map to_eigrp_5 permit 20
                set tag 55

                !--- Route-map statement used to set the tag on routes


                !--- redistributed from EIGRP 3 to EIGRP 5 to "55"
```

These are just a few examples of filtering strategies used for the intent of this document. However, there might be other valid strategies you can use. Refer to the section on Filtering Routing Information in Configuring IP Routing Protocol–Independent Features for more information.

Example 4

For example, you have two routers, one is a high end router running BGP protocol, and the other one is low end router running RIP protocol. When you redistribute BGP routes into RIP, it is possible that you see some packets become lost.

The redistribution of BGP into RIP protocol is generally not recommended and protocols like iBGP, OSPF, and EIGRP are scalable and have wide options available.

In case you encounter this scenario, which is the redistribution between BGP to RIP, and lose some packest, it possible that you have to configure this command on the RIP process:

Router(Config)#router rip

Router(Config–router)# input–queue 1024

Note: Consider the use of the **input–queue** command if you have a high–end router that sends at high speed to a low–speed router that might not be able to receive at the high speed. The configuration of this command help prevent the loss of information from the routing table.

Example 5

This example illustrates Redistributing Static Route into RIP routing protocol. As per the topology, we have three routers (R1, R2, and R3). R1 and R2 have RIP configured on interface Fast Ethernet 0/0. R1 has a static route to reach the Lo 0 interface (ip address 3.3.3.3/32) of Router R3. This static route is redistributed in RIP routing protocol. Router R3 is configured with a default route `R3# ip route 0.0.0.0 0.0.0.0 FastEthernet 0/0`.

```
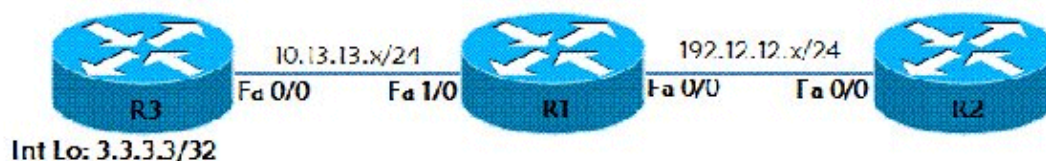        R1(config)# ip route 3.3.3.3 255.255.255.255 10.13.13.3
        R1(config)# router rip
        R1(config-router) redistribute static metric 10
```

On Router R2, route 3.3.3.3 can be seen via the **show ip route** command:

```
R2#show ip route
Codes: 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

Gateway of last resort is not set

C    192.12.12.0/24 is directly connected, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
R       3.3.3.3 [120/10] via 192.12.12.1, 00:00:07, FastEthernet0/0
```

How to Redistribute Single Static Route

In order to redistribute single static route, use **route−map** to select the static route that needs to be redistributed.

```
Router(config)#access-list 1 permit <network no> <mask>

Router(config)#route-map <route-map name> permit 10
Router(config-route-map)#match ip address access list number

Router(config)#router eigrp <As number>

Router(config-router)#redistribute static route-map <map-name> metric <value>
```

Related Information

- **RIP and OSPF Redistribution**
- **Redistribution Between Enhanced IGRP and RIP**
- **White Paper – Enhanced Interior Gateway Routing Protocol**
- **Redistributing Between Classful and Classless Protocols: EIGRP or OSPF into RIP or IGRP**
- **BGP Case Studies**
- **redistribute Command Reference**
- **RIP Support Page**
- **OSPF Support Page**
- **IGRP Support Page**
- **EIGRP Support Page**
- **Technical Support & Documentation – Cisco Systems**