

RIP configuration on Cisco routers

Lab aims

Helping students develop considerable skills of adjusting routers for operation with RIP dynamic routing protocol.

Lab progress

At first, let's find out what actually the dynamic routing protocol is, what peculiarities RIP has, and how it works. Then we will simulate a network and adjust a RIP in it. Finally, let's see what commands should be used to check the operability of the protocol.

Review of RIP

All dynamic routing protocols are meant to pursue the same objective, which is to build an effective network topology without loops.

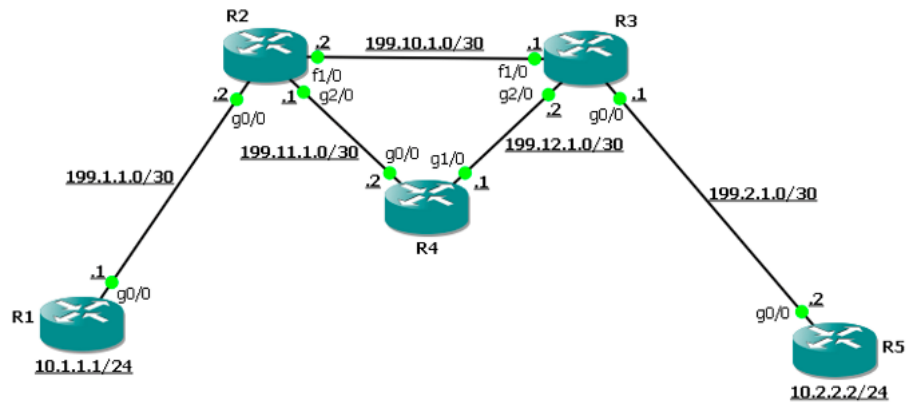
In this lab we will review RIPv2 (version 2). RIP of any version is related to the IGP routing protocols. It is a distance-vector protocol and it doesn't consider the channel states. A transit device counter is used as metrics in this protocol, calculating the number of routers, or any other L3 devices, that are used in the path to the network. RIP carries out slow convergence and possesses one of the highest standard administrative distances ever, 120, by default. Convergence is a process of network coincidence or, in other words, a process of detection of changes in the network as well as of searching for and building of the effective topology. Every routing protocol has a special number called an administrative distance, which is required for the IOS system to be able to pick out the preferred route in case if several routing protocols are used. It's also worth noticing that some routing protocols feature division of routes into internal and external ones. In this case, the AD is designated to various types of routes on a separate basis. Standard AD values for Cisco routers and other vendors can be different. Apart from it, RIP doesn't allow for combining routes with a mask shorter than the class one.

For its operation RIP uses four timers that are presented in the table below.

Name	Value by default	Intended use
Update timer	30 seconds	Responsible for the frequency of sending updates to the nearby devices.
Invalid timer	180 seconds	Responsible for the period of storage of the route information in the routing table without receiving the update. In case if the route update is not received over this period, it will be assigned metrics 16 (unreachable).
Flush timer	240 seconds	The time after which the route is excluded from the routing table.
Holddown timer	180 seconds	Functionality that solely Cisco equipment possesses. Is launched for inaccessible routes. Over this time the route is stored as an unavailable one and its status doesn't change even if the updates about the network are received.

Simulation and configuration

Let's simulate the following network.



At first we should build all necessary interfaces. The interface for R1 is loopback, which simulates the client network, as well as another interface towards the service provider.

```
R1(config)# int lo1
R1(config-if)#ip addr 10.1.1.1 255.255.255.0
R1(config-if)# int g0/0
R1(config-if)#ip addr 199.1.1.1 255.255.255.252
R1(config-if)# no shutdown
```

Similarly for R5.

```
R5(config)# int lo1
R5(config-if)#ip addr 10.2.2.2 255.255.255.0
R5(config-if)# int g0/0
R5(config-if)#ip addr 199.2.1.2 255.255.255.252
R5(config-if)# no shutdown
```

The interfaces for R2 are GigabitEthernet and one FastEthernet.

```
R2(config)# int g0/0
R2(config-if)#ip addr 199.1.1.2 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# int g2/0
R2(config-if)#ip addr 199.11.1.1 255.255.255.252
R2(config-if)# no shutdown
R2(config-if)# int f1/0
R2(config-if)#ip addr 199.10.1.2 255.255.255.252
R2(config-if)# no shutdown
```

Similarly for R3 and R4.

```
R3(config)# int g0/0
R3(config-if)#ip addr 199.2.1.1 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# int g2/0
R3(config-if)#ip addr 199.12.1.2 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# int f1/0
R3(config-if)#ip addr 199.10.1.1 255.255.255.252
R3(config-if)# no shutdown
R4(config)# int g0/0
R4(config-if)#ip addr 199.11.1.2 255.255.255.252
R4(config-if)# no shutdown
R4(config-if)# int g1/0
R4(config-if)#ip addr 199.12.1.1 255.255.255.252
R4(config-if)# no shutdown
```

Currently the local networks of 10.1.1.1 and 10.2.2.2 clients cannot detect one another since the corresponding route information is not available. Let's switch to the RIP configuration. One needs to enter the following commands in every router: **router rip** in the global configuration mode so as to switch for the protocol configuration mode; **network network_address** so as to switch the protocol on for the necessary interfaces. One should always remember that only the address of a class network must be specified as the network address in **network** command. Apart from it, we should also enter **no auto-summary** command in order to switch off the network summarization at the network edge.

An example of router R1 configuration.

```
R1(config)# router rip
R1(config-router)# network 199.1.1.0
R1(config-router)# redistribute connected
R1(config-router)# no auto-summary
```

Observation: if we had specified 10.1.1.0 in the second line as the network, the system wouldn't have issued any errors or warnings. Nonetheless, RIP would have been launched in all interfaces whose sub-networks are included in 10.0.0.0/8 network.

Important observation: we added networks of all connected interfaces (including the network of loopback interface) into the database of RIP – RIP DataBase (RDB) protocol using *redistribute connected* command, however we didn't switch on the protocol itself in these interfaces. Therefore, the information about this network is forwarded by the protocol through all interfaces where RIP functions (in our case through gi0/0), but the other interfaces do not send out RIP packets and listen to the network for incoming RIP messages.

An example of R2 configuration.

```
R2(config)# router rip
R2(config-router)# network 199.10.1.0
R2(config-router)# network 199.1.1.0
R2(config-router)# network 199.11.1.0
R2(config-router)# no auto-summary
```

Router R3 configuration.

```
R3(config)# router rip
R3(config-router)# network 199.10.1.0
R3(config-router)# network 199.2.1.0
R3(config-router)# network 199.12.1.0
R3(config-router)# no auto-summary
```

Configuration of R4 router.

```
R4(config)# router rip
R4(config-router)# network 199.12.1.0
R4(config-router)# network 199.11.1.0
R4(config-router)# no auto-summary
```

Router R5 settings.

```
R5(config)# router rip
R5(config-router)# network 199.2.1.0
R5(config-router)# redistribute connected
R5(config-router)# no auto-summary
```

In case if one attempts to enter *ping 10.2.2.2 source 10.1.1.1* command from R1 during this phase, the router will notify him/her about the unavailability of the node with 10.2.2.2 address. The thing is that RIPv1, which supports operation only with classful networks, will be launched by default. Thus, in our case, the protocol database will be added with only one network and not two: 10.0.0.0/8 instead of 10.1.1.0/24 and 10.2.2.0/24 since RIPv1 doesn't use the masks of these networks. One can make sure about this by reviewing the routing table and RDB on R2 and R3 which are passed the information solely about 10.0.0.0/8 network by R1 and R5. Fix this problem by entering *version 2* command in the routing protocol configuration mode in all devices.

That is where we bring the adjustment of the devices to a conclusion. Now let's pass directly on to testing it.

Testing

1. Make sure that the client local networks have access to one another using *ping 10.2.2.2 source 10.1.1.1* and *trace 10.2.2.2 source 10.1.1.1* commands that should be performed from R1 router.
2. Analyze the route which is used for transferring the packets between the two networks specified in the previous item.
3. Switch off the low-speed link between R2 and R3 routers. In what way would the packet movement route be changed?

4. Check RIP settings in every router using *show ip protocols* command.
5. Enter *show ip route rip* command and analyze its output.
6. Perform the traffic capturing between the routers and analyze RIP messages.
7. Demonstrate the operation of the split horizon method in RIP using the traffic interception from the previous item.

'Floating' static route

The traffic is transferred through FastEthernet channel in the scheme we built earlier. This may be ineffective due to the throughput of the interface, which is lower than that of GigabitEthernet. In this section we will adjust the so-called floating route, which will help us to deal with this problem.

At first one needs to switch off RIP in fa1/0 interface on R2 and R3.

```
R2(config)# router rip
R2(config-router)# no network 199.10.1.0
R3(config)# router rip
R3(config-router)# no network 199.10.1.0
```

Now let's adjust the static routes to the networks in loopback interfaces, however using the administrative distance of 130.

```
R2(config)# ip route 10.2.2.0 255.255.255.0 199.10.1.1 130
R3(config)# ip route 10.1.1.0 255.255.255.0 199.10.1.2 130
```

Let's add these routes in RIP (please provide the explanation yourself on why this needs to be done).

```
R2(config)# router rip
R2(config-router)# redistribute static
R3(config)# router rip
R3(config-router)# redistribute static
```

Let's make sure that the packets are transferred through R4 using *trace 10.2.2.2 source 10.1.1.1* command performed from R1 router. Apart from this, let's also review the routing table on R2 using *show ip route* command to make sure that there is no static route present there.

Now let's shutdown the interfaces towards R4.

```
R2(config)# int g2/0
R2(config-if)# shutdown
R3(config)# int g2/0
R3(config-if)# shutdown
```

Let's review the routing table once more to make sure that the static route is now present in it. And, finally, let's do this once more by entering *trace 10.2.2.2 source 10.1.1.1* command from R1 to make sure that the network connectivity is not disrupted.