Central America Workshop - Guatemala City

Guatemala

30 January -1 February '07

RIPng Configuration

Pedro Lorga (lorga@fccn.pt)

Simon Muyal (muyal@renater.pt)

Piers O'Hanlon (p.ohanlon@cs.ucl.ac.uk)

Laboratory Exercise: *RIPng configuration*

Objectives

In this laboratory exercise you will complete the following tasks:

- *Enable RIPng on a Cisco router*
- *Gather information regarding RIPng protocol*
- *Debug RIPng IPv6*

Visual Objective

The following figure shows the configuration of the RIPng laboratory:

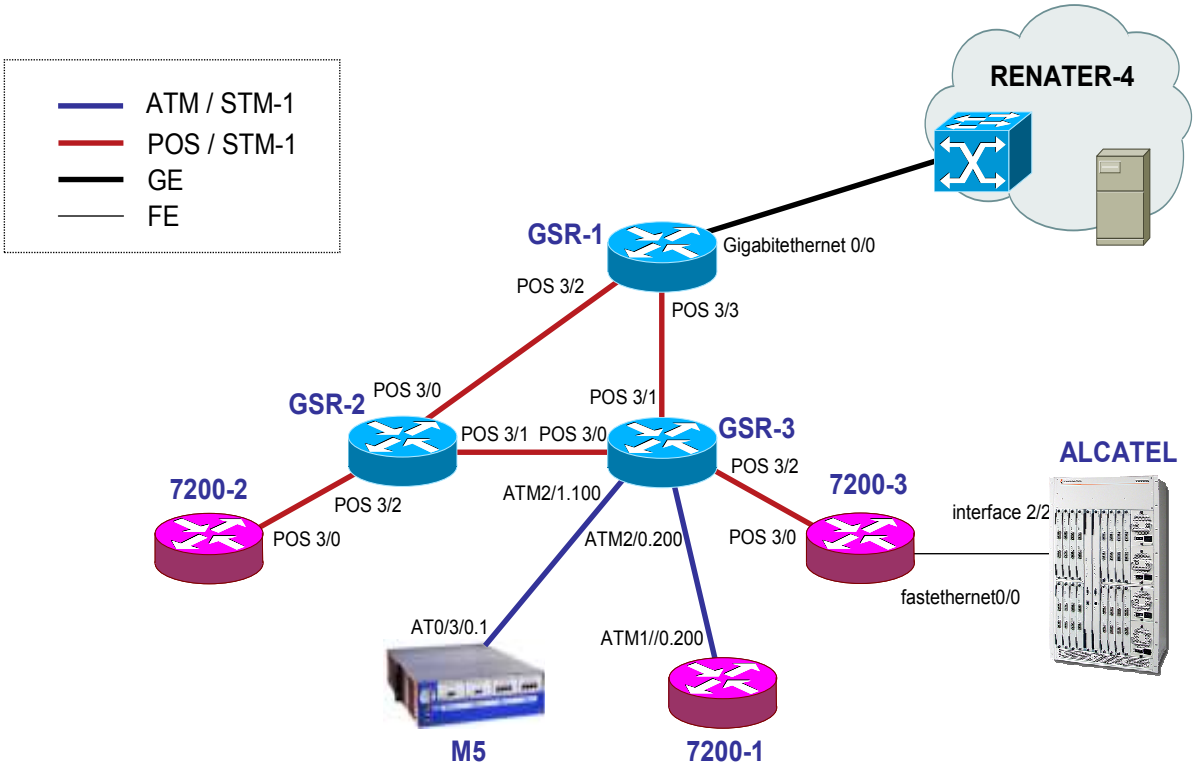

Figure 1: Scenario topology

Setup/Scenario

In this scenario there are two Cisco GSR routers and three 7200 that you will use. The routers are connected by ATM or POS ports to each other.

Preparing the LAB

There will be 2 groups per router.

Groups	Routers
Group 1	GSR 2
Group 2	
Group 3	GSR 3
Group 4	
Group 5	7200 – 1
Group 6	
Group 7	7200 – 2
Group 8	
Group 9	7200 – 3
Group 10	

Table 1: Groups per Routers

To connect to the router, you should use the fallowing IPv4 addresses:

Name	How to connect
GSR-2	194.254.101.5
GSR-3	194.254.101.6
7200-1	194.254.101.12
7200-2	194.254.101.8
7200-3	194.254.101.9

Table 2 - Addresses to connect

IPv4 Configured Interconnections:

Router #1 (IPv4 address)	Router #2 (IPv4 address)	Interconnection prefix
GSR-1 (194.254.101.73)	**GSR-2** (194.254.101.74)	194.254.101.72/30
GSR-1 (194.254.101.77)	**GSR-3** (194.254.101.78)	194.254.101.76/30
7200-2 (194.254.101.45)	**GSR-2** (194.254.101.46)	194.254.101.44/30
GSR-2 (194.254.101.49)	**GSR-3** (194.254.101.50)	194.254.101.48/30
GSR-3 (194.254.101.53)	**7200-3** (194.254.101.54)	194.254.101.52/30
GSR-3 (194.254.101.69)	**7200-1** (194.254.101.70)	194.254.101.68/30

Table 3 - IPv4 Interconnection addresses

Bellow you'll find the IPv6 addresses you should use on your routers.

Loopback addresses:

Name	Loopback address
GSR-2	2001:660:3007:8005::1/64
GSR-3	2001:660:3007:8006::1/64
7200-1	2001:660:3007:8012::1/64
7200-2	2001:660:3007:8008::1/64
7200-3	2001:660:3007:8009::1/64

Table 4 – Loopback addresses to use

IPv6 Interconnections:

Interconnecions (R1 - R2)	Prefix
GSR-1 - GSR-2	2001:660:3007:8101::/64
GSR-1 - GSR-3	2001:660:3007:8102::/64
7200-2 - GSR-2	2001:660:3007:8103::/64
GSR-2 - GSR-3	2001:660:3007:8104::/64
GSR-3 - 7200-3	2001:660:3007:8105::/64
GSR-3 - 7200-1	2001:660:3007:8108::/64

Table 5 - Interconnection addresses

R1 has address = prefix::1

R2 has address = prefix::2

Task 1: *Enabling RIPng*

Step 1: Testing connectivity

Connect to your router using the IPv4 address provided.

```
Login: 6diss
Password: 6diss
```

The first step is to check if your router has IPv6 routing enabled. The global **`ipv6 unicast-routing`** command should appear in the running configuration.

Try to ping another router that is not directly connected to yours. Did you succeed?

Step 2: Enable protocol on the interface

Now, configure the RIP protocol on the interfaces in which you want to enable IPv6.

(**Tip:** `routerX(config-if)# ipv6 …`)

- Check the interfaces in figure 1;

Step 3: Enabling RIPng process

Create a RIP process, named *riptest* on your router.

(**Tip:** `routerX(config)# ipv6 …`)

Step 4: Defining maximum number of paths

Enter into your RIPng process, and configure it so that two paths are available for each destination.

(**Tip1**: RouterX(config)# **`ipv6 router`**)

(**Tip2: `routerX(config-rtr)# maximum`**…)

Step 5: Redistributing routes

Now that the process is running, try again to ping another router not directly connected. Did you get a reply? Why?

On the RIPng process configuration, redistribute the connected and static routes.

(**Tip:** `routerX(config-rtr)# redistribute …`)

Step 6: Check your connectivity

Try to *ping* again the routers and PCs.

Step 7: Originate the default Route (only for router GSR-3)

Consider that the router is the gateway for your entire network. This router should originate the default gateway.

(**Tip:** `routerX(config-rtr)# ipv6 rip <name> default-information …`)

Task 2: *Verifying RIP configuration*

On Task 1 you've configured the RIPng protocol, now you will gather information in order to debug any problem.

Step 1: Using the *show* command

The RIPng process is now running on all routers, but are you receiving the information from all of them?

- Collect the information from the RIP process you are running and see if all routers are participating by looking at the RIP database. Also see the next-hop information.
- Check the routing table from RIP
- Now disconnect router 7200-2 by deleting the IPv6 address on the POS3/0 interface o the 7200-2 router. What changes do you see in the database?
- Look again at the RIP process information and routing table. What differences do you see?

  (**Tip:** `router`***x***`# show ipv6 rip …`)

Step 2: Debug the RIPng process

RIPng also has debugging facilities on the IOS software. This provides more detailed information than the delivered by the *show* command.

- Initiate the debugging process of RIPng process
- Debug RIPng on one interface in which you are running the protocol

(**Tip:** `router`***x***`# debug ipv6 rip …`)

What do you see?

Change the *maximum paths* value in your router. What do you see?

Summary

After completing these exercises, you should be able to:

- *Configure RIPng*
- *Debug and analyze information from the RIPng*

Appendix A

Task 1: *Enabling RIPng*

Step 2: Enable protocol on the interface

To configure RIPng on the interfaces you want to run the protocol, you can use the following command lines:

```
RouterX# enable
RouterX# configure terminal
RouterX(config)# interface POS or ATM[X]
RouterX(config-if)# ipv6 rip process_name enable
```

Where *process_name* is the specific name of the RIPng process you will configure.

Eg:

```
GSR-2# enable
GSR-2# configure terminal
GSR-2# (config)# interface pos3/2
GSR-2#config-if)# ipv6 rip riptest enable
```

Step 3: Enabling RIPng process

```
RouterX# configure terminal
RouterX(config)# ipv6 router rip riptest
Note: on some models the command line might be
      ipv6 rip riptest
```

Step 4: Defining maximum number of paths

Enter in the protocol configuration command line and type the appropriate commands:

Router*X*(config)# **ipv6 router rip riptest**

Router*X* (config-rtr)#**maximum-paths 2**

Step 5: Redistributing routes

To redistribute the connected and static routes enter into the RIPng process and type the appropriate commands:

Router*X*(config)# **ipv6 router rip riptest**

Router***X*** (config-rtr)# **redistribute connected**

Router***X*** (config-rtr)# **redistribute static**

To redistribute routes from another protocol use the same approach.

Step 7: Originate the default Route

To originate the default route, in the interface where you want to send this advertisement, you must type the commands:

```
GSR-3# configure terminal
GSR-3# (config)# interface pos3/2
GSR-2#(config-if)# ipv6 rip riptest default-information originate
```

Repeat the command on the other interfaces (ATM2/0.200, POS 3/0).

The other way to do this is using the following command:

```
Router***X***#(config-if)# ipv6 rip riptest default-information only
```

This will make the router to only announce the *default route*, and no other routes or updates. The *originate* option will announce the updates and routes, plus the default route. You can also have more them one default route and define a metric to choose between each other:

```
Router***X***#(config-if)# ipv6 rip riptest default-information [only|originate] [metric value]
```

Task 2: *Verifying RIP configuration*

Step 1: Using the *show* command

- Collect the information from the RIP process

Note: the fallowing outputs are only examples. They are not the output from these exercises. The sole purpose is to show how they look like.

RouterX# **show ipv6 rip database**

```
RIP process "riptest", local RIB
 2001:DB8:CAFE:4::1/128, metric 3, installed
   FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
 2001:DB8:CAFE:4::/64, metric 3, installed
   FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
 2001:DB8:CAFE:D::/64, metric 3, installed
   FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
 2001:DB8:CAFE:13::/64, metric 2
   FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
 2001:DB8:CAFE:34::/64, metric 2, installed
   FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
```

```
::/0, metric 2, installed
  FastEthernet1/FE80::216:C8FF:FE30:5FC4, expires in 170 secs
```

RouterX# **show ipv6 rip next-hops**

```
RIP process "riptest", Next Hops
 FE80::217:E0FF:FED6:7D3/FastEthernet0 [4 paths]
 FE80::218:19FF:FE18:964C/Vlan32 [4 paths]Check the routing table
```

RouterX# **show ipv6 route rip**

```
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS -
ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 -
OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
R   ::/0 [120/2]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet1
R   2001:DB8:CAFE:4::/64 [120/3]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet1
R   2001:DB8:CAFE:4::1/128 [120/3]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet1
R   2001:DB8:CAFE:D::/64 [120/3]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet1
R   2001:DB8:CAFE:34::/64 [120/2]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet1
```

Step 2: Debug the RIPng process

- Send the output from debug to your monitor:

  RouterX# **terminal monitor**

- Debug the RIPng

  RouterX# **debug ipv6 rip**

```
*Jul 12 08:39:36.479: RIPng: response received from
FE80::217:E0FF:FED6:7D3 on FastEthernet0 for quitorip
*Jul 12 08:39:36.479: src=FE80::217:E0FF:FED6:7D3 (FastEthernet0)
*Jul 12 08:39:36.479: dst=FF02::9
*Jul 12 08:39:36.479: sport=521, dport=521, length=92
*Jul 12 08:39:36.479: command=2, version=1, mbz=0, #rte=4
```

```
*Jul 12 08:39:36.479: tag=0, metric=1, prefix=2001:DB8:CAFE:1::1/128
*Jul 12 08:39:36.479: tag=0, metric=1, prefix=2001:DB8:CAFE:A::/64
*Jul 12 08:39:36.479: tag=0, metric=1, prefix=2001:DB8:CAFE:13::/64
(...)
```

- Debug RIPng on an interface

  gsr-2# **debug ipv6 rip pos3/1**