Central America Workshop - Guatemala City

Guatemala

30 January - 1 February '07

OSPFv3 Configuration

Pedro Lorga (lorga@fccn.pt)

Simon Muyal (muyal@renater.pt)

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

Laboratory Exercise: *OSPFv3 configuration*

Objectives

In this laboratory exercise you will complete the following tasks:

- *Configure OSPFv3 on a Cisco router*
- *Check the exchanged routing information and perform application-level tests towards addresses received from the other two routers*
- *Reset the OSPFv3 process*
- *Enable OSPFv3 on multiple areas*
- *Gather information regarding this protocol*

Visual Objective

The following figure shows the configuration of the laboratory for 5 routers (GSR-2, GSR-3, 7200-1, 7200-2, 7200-3). All the routers will be in AREA 0 for the scenario:

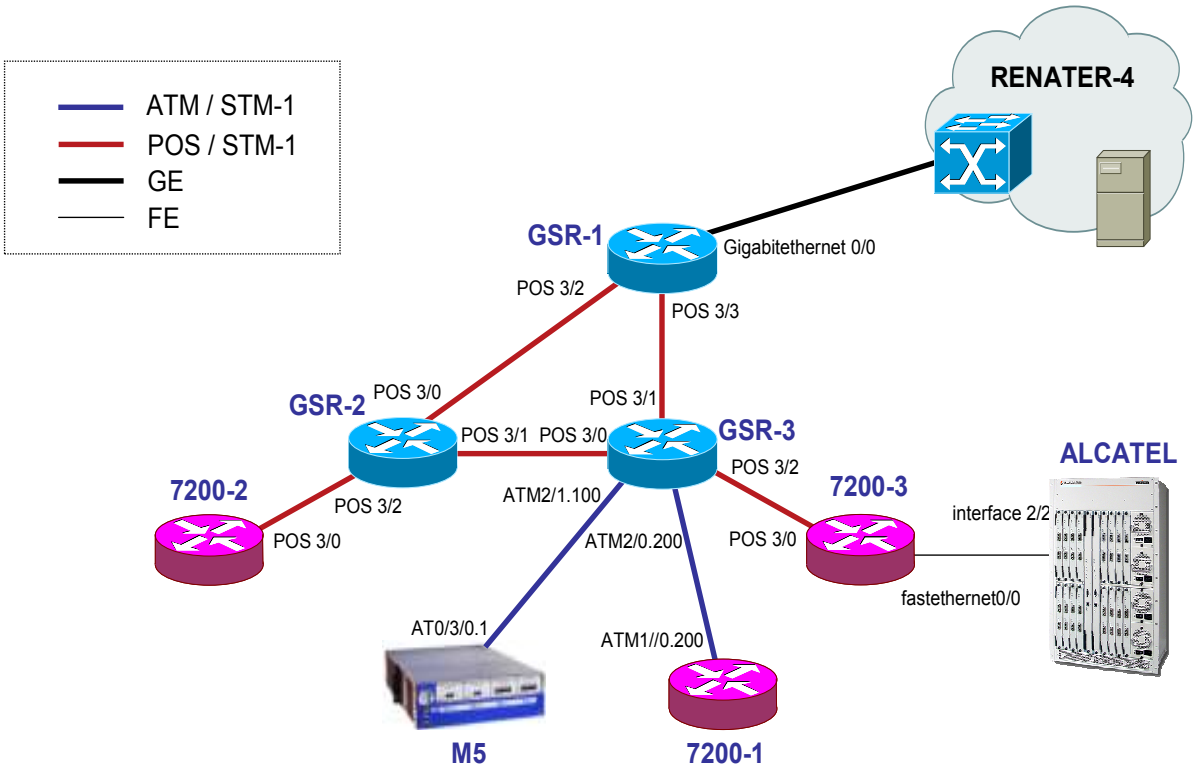

Figure 1: Area 0 Scenario

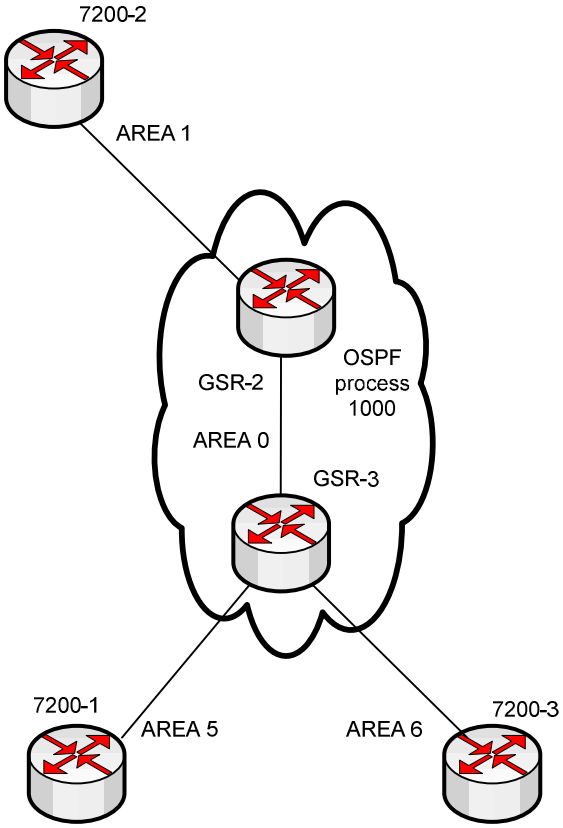

Figure 2 - Multiple Areas Scenario

Setup/Scenario

This is the same scenario used for the RIPng configuration.

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 the OSPFv3 protocol*

Complete the following steps for creating an OSPFv3 process on your router

Step 1: Testing connectivity

Connect to your router. Use the IPv6 *auto configuration* and plug your PC to any FastEthernet port on the router and *telnet* to it using the following authentication data:

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

The first step you must do is to see if your router has IPv6 routing active. The global **ipv6 unicast-routing** command should appear in the configuration.

Try to *ping* another router that is not directly connected to yours.

Step 2: Activate OSPF on the interface

Now, configure the OSPFv3 protocol on the interfaces in which you want to advertise IPv6. IPv6 must be already enabled on that interface. The *process ID* is 1000 and the Area is 0.

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

Step 3: Activate the routing process

Look at the configuration. Is the routing process already enabled?

Note that the neighboring routers are identified by their *route ID*.

Use the fallowing table to configure you router ID.

Router	Router ID
GSR-2	2.2.2.2
GSR-3	3.3.3.3
7200-1	4.4.4.4
7200-2	5.5.5.5
7200-3	6.6.6.6

Table 6 - Router ID

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

Step 4: Test connectivity again

Test your connectivity by *pinging* an interface on a router not directly connected (the IPv6 address of FastEthernet5). Did you succeed? Why?

Step 5: Redistributing routes

Redistribute the connected and static networks via OSPFv3

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

Step 6: Originating the default gateway (only for routers 3 and 4)

On router GSR-2, make it as the default gateway for the OSPF area and router GSR-3 as the backup default gateway. Note that you must use different metrics on the routers.

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

Is the router with higher metric the primary default gateway for the OSPF area?

Task 2: *Verifying OSPFv3 configuration*

Step 1: Check OSPFv3 interfaces

Use the show command to see if you are running OSPFv3. Also verify OSPFv3 interface related information.

(**Tip:** `routerX# show ipv6 ospf …`)

Analyze the output. In what area are you? What kind of router is yours?

Step 2: Check OSPFv3 neighbours

Which routers do you have as neighbours?

(**Tip:** `routerX# show ipv6 ospf …`)

What interfaces are running OSPFv3?

In what state are they?

Do you know the meaning of these values (DR, BDR, Full, INIT, 2WAY, DROTHER)?

Step 3: Check the OSPFv3 database

Are the routes you see from your neighbours correct?

(**Tip:** `routerX# show ipv6 ospf …`)

What type of link states do you see? Can you explain what they mean?

Step 4: Looking at the routes

Check the routing table.

(**Tip:** `routerX# show ipv6 routes…`)

Can you identify which ones you are receiving via OSPF?

There are different types of OSPF route codes. Explain what they mean and why you get different values.

Task 3: *Configuring OSPFv3 in multiple areas*

Step 1: Configure OSPFv3 in multiple areas

On your router, configure OSPFv3 according to Figure 2 **- Multiple Areas**

Step 2: Analysing the changes

Repeat the "show …" command several times.

What are the main differences you see?

Summary

After completing these exercises, you should be able to:

- *Configure OSPFv3 Process*
- *Debug and analyze information from the process*

Appendix

Task 1: *Enabling the OSPFv3 protocol*

Step 1: Testing connectivity

From your PC ping other routers or any PC not directly connected to your router.

Note: To check if the IPv6 commands are already in the router type:

```
RouterX# show running-config | inc ipv6
```

Step 2: Activate OSPF on the interface

```
RouterX# enable
RouterX# configure terminal
RouterX(config)# interface fastethernet[X]
RouterX#(config-if)# ipv6 ospf processID area area
```

Where *process_ID* is the specific name of the OSPFv3 process you will configure. Eg:

```
RouterX(config)# interface fastethernet0
RouterX#(config-if)# ipv6 ospf 1000 area 0
```

Step 3: Activate routing process

If you look, you can see that the routing process is already created:

```
Router1# show configuration | inc ospf
ipv6 ospf 1000 area 0
ipv6 router ospf 1000
```

There are two lines, the one you configured before and the routing process that was automatically created.

```
Router1(config)# ipv6 router ospf 1000
Router1 (config-rtr)#  router-id 1.1.1.1
```

Step 5: Redistributing routes

```
Router1(config)# ipv6 router ospf 1000
Router1 (config-rtr)#  redistribute connected
Router1 (config-rtr)#  redistribute static
```

Note: The routes from an interface will only be announced if that interface is up, or if you add its address to the routing table, for example by introducing a static route:

```
Router1(config)# ipv6 route 2001:DB8:CAFE:A::/64 null 0
```

Step 6: Originating the default gateway

Having two default gateways in your network is a good idea. The router with lower metric value will be the preferred gateway. Eg:

```
Router3(config)# ipv6 router ospf 1000
Router3 (config-rtr)#  default-information originate always metric 100
Router4(config)# ipv6 router ospf 1000
Router4 (config-rtr)# default-information originate always metric 200
```

Task 2: *Verifying OSPFv3 configuration*

Step 1: Check OSPFv3 interfaces

```
Router1# show ipv6 ospf
It is an autonomous system boundary router
Originate Default Route with metric 100 always
(...)
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Area BACKBONE(0)
        Number of interfaces in this area is 2
(…)
```

```
Router1# show ipv6 ospf interfaces
(…)
FastEthernet0 is up, line protocol is up
  Link Local Address FE80::216:C8FF:FE30:5FC4, Interface ID 2
  Area 0, Process ID 1000, Instance ID 0, Router ID 3.3.3.3
  Network Type BROADCAST, Cost: 1
(…)
 Designated Router (ID) 1.1.1.1, local address FE80::7D2
  Backup Designated router (ID) 3.3.3.3, local address FE80::FC4
(…)
```

Step 2: Check OSPFv3 neighbors

```
Router3# show ipv6 ospf neighbor
Neighbor ID    Pri    State    Dead Time Interface ID  Interface
```

```
4.4.4.4         1   FULL/BDR    00:00:30        2       Vlan32
1.1.1.1         1   FULL/DR     00:00:37        2    FastEthernet0
```

Step 3: Check the OSPFv3 database

```
Router1# show ipv6 ospf database

            OSPFv3 Router with ID (1.1.1.1) (Process ID 1000)

                    Router Link States (Area 0)

ADV Router      Age      Seq#         Fragment ID  Link count  Bits
  1.1.1.1       81       0x80000047   0            1           E
  3.3.3.3       76       0x80000040   0            1           E
  (...)

                    Net Link States (Area 0)

ADV Router      Age      Seq#         Link ID    Rtr count
  1.1.1.1       87       0x80000008   2          2
  (...)

        Link (Type-8) Link States (Area 0)

ADV Router      Age      Seq#         Link ID    Interface
  1.1.1.1       1320     0x80000028   2          Fa0
(...)

               Intra Area Prefix Link States (Area 0)

ADV Router      Age      Seq#         Link ID    Ref-lstype  Ref-LSID
  1.1.1.1       327          0x80000008   1002        0x2002   2
  (...)

                Type-5 AS External Link States

ADV Router      Age          Seq#         Prefix
  1.1.1.1       563          0x80000006   2001:DB8:CAFE:A::/64
  (...)
```

Step 4: Looking at the routes

```
Router1# show ipv6 route

IPv6 Routing Table - 5 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
OE2  2001:DB8:CAFE:A::/64 [110/20]
     via FE80::217:E0FF:FED6:7D2, FastEthernet0
C    2001:DB8:CAFE:13::/64 [0/0]
     via ::, FastEthernet0
L    2001:DB8:CAFE:13::3/128 [0/0]
     via ::, FastEthernet0
L    FE80::/10 [0/0]
```

```
          via ::, Null0
      L   FF00::/8 [0/0]
          via ::, Null0
```

Task 3: *Configuring OSPFv3 in multiple areas*

Step 1: Configure OSPFv3 in multiple areas

On your router, configure the interface to be connected to a specific area:

```
RouterX(config)# interface int X
RouterX(config-if)# ipv6 ospf 1000 area 1
```

Step 2: Analysing the changes

You can now see different output from some command lines. Eg on router1:

```
routerX# show ipv6 route ospf
(...)
     O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext
2
   (...)
OI  2001:DB8:CAFE:34::/64 [110/2]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet0

before:

O   2001:DB8:CAFE:34::/64 [110/2]
     via FE80::216:C8FF:FE30:5FC4, FastEthernet0
```

routerX# **show ipv6 ospf interface**

```
FastEthernet1 is up, line protocol is up
  (...)
  Area 1, Process ID 1000, Instance ID 0, Router ID 1.1.1.1
(...)
  Designated Router (ID) 3.3.3.3, local address
(...)
```

```
routerX# show ipv6 ospf database
            OSPFv3 Router with ID (1.1.1.1) (Process ID 1000)
                 Router Link States (Area 1)
ADV Router      Age         Seq#        Fragment ID  Link count  Bits
1.1.1.1         955         0x80000003  0            1           E
3.3.3.3         958         0x80000004  0            1           EB
                 Net Link States (Area 1)
ADV Router      Age         Seq#        Link ID    Rtr count
(...)
                 Inter Area Prefix Link States (Area 1)
ADV Router      Age         Seq#        Prefix
(...)
                Inter Area Router Link States (Area 1)
ADV Router      Age         Seq#        Link ID    Dest RtrID
```

```
(...)
                Link (Type-8) Link States (Area 1)
(...)
                Intra Area Prefix Link States (Area 1)
(...)
ADV Router      Age         Seq#        Link ID    Ref-lstype  Ref-
(...)
                Type-5 AS External Link States
ADV Router      Age         Seq#        Prefix
(...)
```

Note: The *Inter Area Prefix Link States (Area 1)* and *Inter Area Router Link States (Area 1)* now appear.

Some useful commands

- To see the number of routes by prefix

  `Router1#` **show ipv6 route summary**

- Forcing the SPF recalculation

  `Router1#` **clear ipv6 ospf process**

If you want to recalculate the SFP algorithm again, clear the OSPF database. If you type **clear ipv6 ospf force-spf** instead, the database will not be cleared before you run the SFP algorithm.

- Removing an interface from the OSPFv3 process

On OSPFv2, if you didn't want to run OSPF in all interfaces, you had to configure them as passive.

```
Router1(config)# ipv6 router ospf 1000
Router1 (config-rtr)#  passive-interface default
Router1 (config-rtr)#  no passive-interface FastEthernet0
```

All interfaces will not send OSPFv3 messages, except when you deny the command for any give interface.

- Authentication neighbors on an interface

Eg:

```
RouterX(config-if)# ipv6 ospf authentication ipsec spi 1000
md5 12345678900987654321ascdefedcba0
```

Where SPI value means Security Policy Index (a value between 256 and 4294967295) and the values after MD5 are the key in HEX format. One can also choose the SHA-1 algorithm instead of MD5.

- Authentication neighbors on an OSPF area

Eg:

```
RouterX(config-rtr)# area 0 authentication ipsec spi 1000 md5
12345678900987654321ascdefedcba0
```

Where SPI value means Security Policy Index (a value between 256 and 4294967295) and the values after MD5 are the key in HEX format. One can also choose the SHA-1 algorithm instead of MD5.

Debug commands

Try these commands and analyse their output.

- **debug ipv6 ospf packets**
- **debug ipv6 events**
- **debug ipv6 ospf adj**