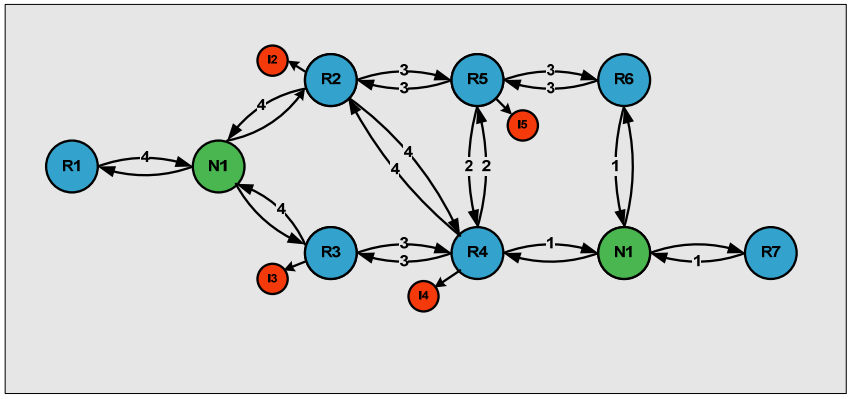
NETMASTERCLASS
ROUTING AND SWITCHING CCIE® TRACK

DOiT-200v6

VOLUME II



Scenario 2
ANSWER KEY

FOR

CCIE® CANDIDATES

☎ 1-888-677-2669 http://www.netmasterclass.net 13530 Dulles Technology Drive, Suite #150, Herndon, VA 20171

Disclaimer

NetMasterClass, LLC is an independent training and consulting company based in Herndon, Virginia. The terms “Cisco”, “Cisco Systems” and “CCIE” are the trademarks of Cisco Systems, Inc. NetMasterClass, LLC is Cisco Learning Partner.

Cisco Non-Disclosure Agreement Compliance

All products and services offered by NetMasterClass, LLC are in full compliance with the Cisco CCIE Lab non-disclosure agreement. The content of the NetMasterClass CCIE preparation materials is based upon the NetMasterClass “issue spotting and analysis” internetwork training methods.

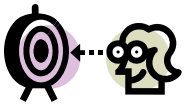
NOTE: To use this document to its maximum effectiveness, access the SHOWiT engine while reviewing each section of this Answer Key.

DOiT-V6 Scenario 2: Spot the Issue Answer Key

Table of Contents

2.1 Frame Relay....6
2.2 Catalyst Configuration....8
2.3 OSPF....16
2.4 RIP....18
2.5 EIGRP....21
2.6 ODR....23
2.7 BGP....25
2.8 IPv6 Addressing....28
2.9 IPv6 Routing....33
2.10 Router Maintenance....42
2.11 Security....43
2.12 QoS....45
2.13 Catalyst Specialties....45
2.14 Gateway Redundancy....47
2.15 Multicast....48

REGARDLESS OF ANY CONFIGURATION YOU PERFORM IN THIS EXAM, IT IS VERY IMPORTANT TO CONFORM TO THE GENERAL GUIDELINES PROVIDED BELOW. IF YOU DO NOT CONFORM TO THEM, THIS CAN RESULT IN A SIGNIFICANT DEDUCTION OF POINTS IN YOUR FINAL EXAM SCORE.



Goals and Restrictions

- IP subnets on the diagram belong to network 172.16.0.0/16.
- Do not use any static routes.
- Advertise Loopback interfaces with their original mask.
- Network 0.0.0.0/0 should not appear in any routing table (show ip route).
- Do not use the "ip default-network" command.
- All IP addresses involved in this scenario must be reachable, unless specified otherwise.
- Networks advertised in the BGP section must be reachable only in the BGP domain.
- Use conventional routing algorithms.

Explanation of Each of the Goals and Restrictions

IP subnets on the diagram belong to network 172.16.0.0/16 unless specified otherwise.

All IP addresses in this Exam belong to the 172.16.0.0/16 address space with the exception of a set of prefixes used in the BGP section.

Do not use any static routes.

Static routes can be used to solve a variety of reachability problems. However, you cannot have any route marked with an "S" in any of your unicast routing tables in this lab. You must rely on skillful configuration of all your unicast routing protocols.

Advertise loopback entries with their original masks.

By default OSPF advertises the loopback interfaces as host entries /32. Make sure you do not overlook it.

Do not use 0.0.0.0/0 anywhere in this scenario.

A 0.0.0.0/0 entry can also be used to solve a range of reachability problems. In this exercise, you cannot use any 0.0.0.0/0 entries.

Do not use "ip default-network" in this scenario.

You can solve the reachability issues by using the "ip default-network" command. This solution is not permitted.

All IP addresses involved in this scenario must be reachable unless specified otherwise.

This is a key goal to observe. This requires that all of your IGPs and your routing policy tasks are configured properly. The key elements of your routing policy include route redistribution and controlling routing updates using distribute-lists, route-maps and the distance command. Note that the term "redistribution" is never explicitly used in this exam. However, you must perform redistribution in order to

assure that all IP addresses are reachable, and you must do so without violating any of the other constraints listed above.
Some IP addresses are excluded from the reachability requirement. Please check the scenario ODR section for more information.

Networks advertised in the BGP section must be reachable only in the BGP domain.

In section 2.8, the BGP section, three non-172.16.0.0 prefixes will be advertised: 4.4.4.0/24, 5.5.5.0/24 and 7.7.7.0/24. This statement relaxes the full-reachability requirement. These prefixes need only be reachable among the routers specified in the BGP section. It is OK if they are in other unicast tables.

Use conventional routing algorithms.

This restriction prevents you from solving any problems by configuring policy routing. At the heart of this restriction is the interpretation of the "conventional routing algorithms". Although this phrase can be interpreted in a number of different ways, the interpretation applied in this workbook is:

> CONVENTIONAL ROUTING ALGORITHMS ARE ROUTING ALGORITHMS THAT APPLY DESTINATION BASED PREFIX LOOKUPS IN A ROUTING TABLE. CONVENTIONAL ROUTING ALGORITHMS DO NOT USE ANY OTHER TYPE OF INFORMAITON OTHER THAN THE DESTINATION ADDRESS TO MAKE A PACKET FORWARDING DECISION.

Due to this restrictive interpretation, no form of policy routing can be applied. Whenever you see this restriction, you will need to use dynamic routing protocols to fulfill all packet forwarding requirements

The following IOS versions were used on the devices:

Device	IOS version
R1	IOS (tm) 3600 Software (C3640-JK9O3S-M), Version 12.4(3)
R2	IOS (tm) 3600 Software (C3640-JK9O3S-M), Version 12.4(3)
R3	IOS (tm) C2600 Software (C2600-J1S3-M), Version 12.3(15a)
R4	IOS (tm) C2600 Software (C2600-J1S3-M), Version 12.3(15a)
R5	IOS (tm) 3600 Software (C3640-JK9O3S-M), Version 12.4(3)
R6	IOS (tm) 3600 Software (C3640-JK9O3S-M), Version 12.4(3)
FRS	IOS (tm) 2500 Software (C2500-JS-L), Version 12.2(27)
CAT1	IOS (tm) C3550 Software (C3550-I5Q3L2-M), Version 12.2(25)SEA
CAT2	IOS (tm) C3550 Software (C3550-I5Q3L2-M), Version 12.2(25)SEA

2.1 Frame Relay

HIDDEN ISSUES TO SPOT WITH THE FRAME-RELAY CONFIGURATION

Issue: Only the PVCs displayed in the diagram are to be used.

The Frame Relay switch is pre-configured for a full mesh of PVC's. When examining the Exam 2 diagram, you see two Frame-Relay clouds: the one in the center of the Exam 2 diagram is a hub and spoke topology with router R1 as the hub, and the second connects just routers R3 and R4.

Solution:

1. Disable inverse arp **no frame-relay inverse-arp** so that no undesirable dynamic inverse arp entries are found on any of the routers. Unneeded inverse arp entries could violate the constraint specified in frame relay section.

2. Provide static frame-relay mappings on each of the Frame-Relay attached routers. Make sure the frame-relay map statements reflect the topology in the Exam 2 diagram. Since the cloud at the center of the Exam 2 diagram is hub and spoke, you must make sure that one spoke of the Frame-Relay topology can ping the other spoke. In order to fulfill this requirement, routers R2 and R3 will need Frame-Relay map statements to router R1, as well as to one another.

Verification:

1. Issue the **show frame-relay pvc** command on each router with a multipoint interface and verify the number of DLCIs marked as *Local*, *Switched* and *Unused*.
    a. Routers R2, and R4 should indicate 3 Active with 1 Local and 2 Unused.
    b. R1 and R3 should indicate 3 Active, 2 Local, 1 unused.
    c. If you are seeing 3 Active and Local on the spokes and you have turned off inverse-arp, then try saving your configuration and reloading. If you see any DLCI's listed as *Deleted*, check you have not mistyped a DLCI in a map or interface-dlci command.

2. Issue the command **show frame-relay map**. Below is the result for R2. Note there are maps to its own IP address (so it can ping itself), to the hub, R1, and to the other spoke in the subnet. Also note the broadcast keyword on each map. The broadcast keyword is not strictly necessary when mapping to your own address or another spoke, but it is a good habit to get into.

```
R2#sh frame map
Serial0/0 (up): ip 172.16.123.1 dlci 201(0xC9,0x3090), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 172.16.123.2 dlci 201(0xC9,0x3090), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 172.16.123.3 dlci 201(0xC9,0x3090), static,
              broadcast,
              CISCO, status defined, active
```

Beware of any maps to 0.0.0.0! They indicate an inverse-arp mapping to an interface without a layer 3 address! To get rid of these, save your configuration and reload.

Issue: You are instructed to use logical interfaces on R1 and R3 and physical interfaces elsewhere.

Solution:

1. R1 will possess a single multipoint subinterface with two DLCI's in the same subnet.
2. R2 will have one DLCI mapped on its physical interface S0/0.
3. R3 will possess two p2p subinterfaces, one to R1 and the other to R4.
4. R4 will have one DLCI mapped to its physical interface S0/0.

Verification:

R1:

```
R1#show frame-relay map
Serial0/0.123 (up): ip 172.16.123.1 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0.123 (up): ip 172.16.123.2 dlci 102(0x66,0x1860), static,
              broadcast,
              CISCO, status defined, active
Serial0/0.123 (up): ip 172.16.123.3 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active
```

R2:

```
R2#show frame-relay map
Serial0/0 (up): ip 172.16.123.1 dlci 201(0xC9,0x3090), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 172.16.123.2 dlci 201(0xC9,0x3090), static,
              CISCO, status defined, active
Serial0/0 (up): ip 172.16.123.3 dlci 201(0xC9,0x3090), static,
              CISCO, status defined, active
```

R3:

```
R3#show frame-relay map
Serial0/0.123 (up): ip 172.16.123.1 dlci 301(0x12D,0x48D0), static,
              broadcast,
              CISCO, status defined, active
Serial0/0.123 (up): ip 172.16.123.2 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0.123 (up): ip 172.16.123.3 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0.34 (up): point-to-point dlci, dlci 304(0x130,0x4C00), broadcast
          status defined, active
```

R4:

```
R4#show frame-relay map
Serial0/0 (up): ip 172.16.34.3 dlci 403(0x193,0x6430), static,
              broadcast,
              CISCO, status defined, active
Serial0/0 (up): ip 172.16.34.4 dlci 403(0x193,0x6430), static,
              CISCO, status defined, active
```

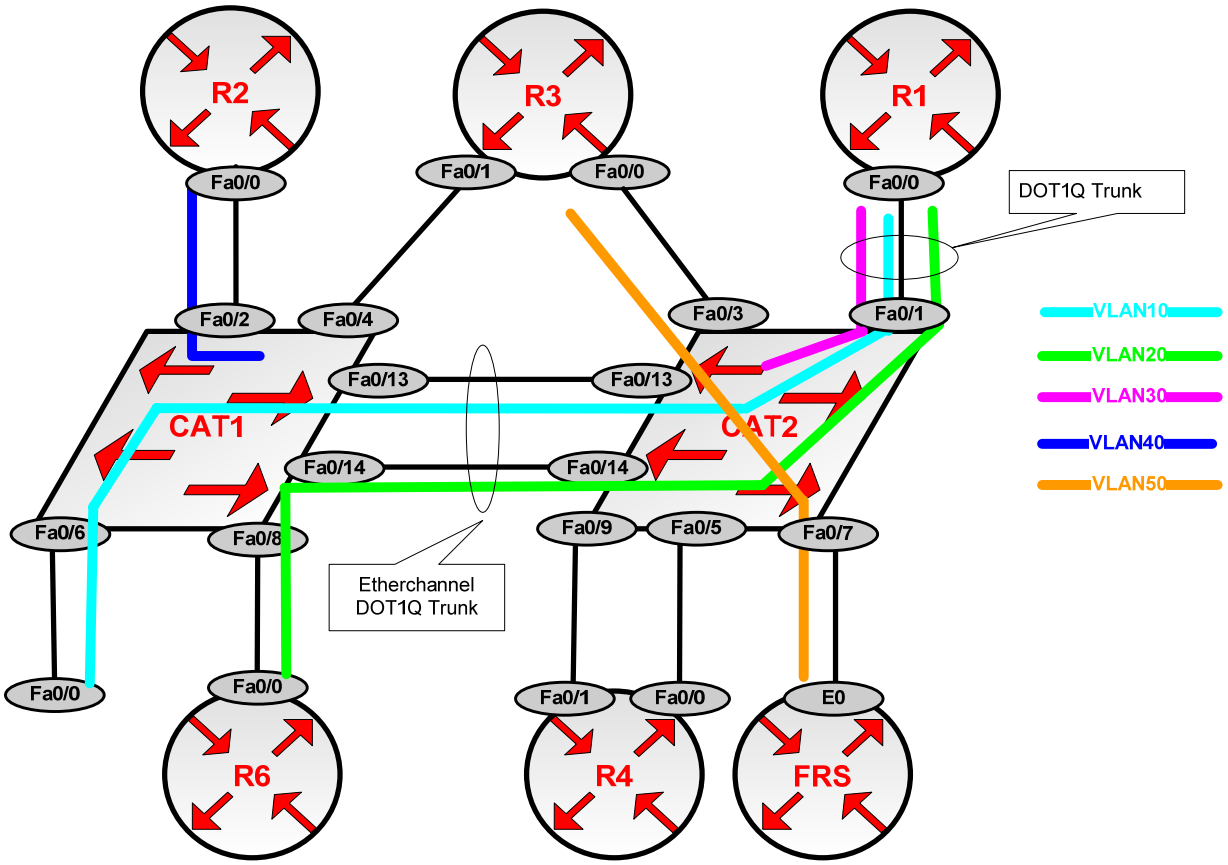
To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.2 Catalyst Configuration

HIDDEN ISSUES TO SPOT WITH THE CATALYST 3550 CONFIGURATION

Like any Catalyst 3550 configuration, you must address the following basic configuration requirements: setting the VTP mode, configuring trunk ports, statically assigning ports to VLAN's. For a good reference on basic Catalyst 3550 configuration tasks, download the following Tech-Note from the Technical Library on the NetMasterClass web-site: "Performing Basic Configuration Tasks on the Catalyst 3550"

VLAN Distribution Diagram



Issue: Configure VLANs according to the table and diagram.

Solution:

VLANs 10, 20, 30, 40, and 50 must be created on CAT1 and propagated to CAT2. These VLANs can be created in VLAN database mode or in global configuration mode.

Verification:

The following page shows the output of the **show VLAN brief** command. You can use this command to verify VLAN names and that all access ports are assigned to the proper VLANs. Note that ports that are in trunking mode do not appear in this output.

```
CAT1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/7, Fa0/9, Fa0/10, Fa0/15
                                                Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                                Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                                Gi0/1, Gi0/2
10   LANA                             active    Fa0/6
20   LANB                             active    Fa0/8
30   LANC                             active
40   LAND                             active    Fa0/2, Fa0/11, Fa0/12
50   LANE                             active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
CAT1#
```

Issue: Advertise VLANs from CAT1 to CAT2. Make sure CAT2 does not advertise any VLANs. Configure domain name as doitlab2.

Solution:

For this Scenario, you must configure CAT1 as a VTP server. Cat2 will learn VLANs from CAT1 whether it is configured as a VTP server or client, but a strict interpretation of the requirement would suggest that it should be a VTP Client, so that it cannot learn VLANs manually and does not allow configuration and advertisement of any VLANs. For example if we try to configure a new VLAN on CAT2 we would get a message similar to the following:

```
CAT2(config)#vlan 555
VTP VLAN configuration not allowed when device is in CLIENT mode.
CAT2(config)#
```

Both CAT1 and CAT2 must be assigned to the same VTP domain *doitlab2*. Note that the default VTP mode is Server, the default VTP domain name is null, and VTP is only advertised across trunk links. The IP address of CAT1 is shown as the update source.

Verification:

```
CAT2#show vtp status
VTP Version                     : 2
Configuration Revision          : 8
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 10
```

```
VTP Operating Mode              : Client
VTP Domain Name                 : doitlab2
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x75 0x28 0x89 0x56 0xD1 0x5B 0xB7 0xD4
Configuration last modified by 172.16.20.10 at 3-1-93 00:57:42
CAT2#
```

Issue: Make sure neither of the ports Fa0/13 or Fa0/14 is in a blocking state for any VLAN.

Solution:

Both of these ports are connected to the other Catalyst, as shown in the Exam 2 pod-cabling scheme. Since there are two parallel connections between the same two Catalysts, there is a looped topology, and Spanning Tree will put one of these ports into a blocking state on the non-root switch. Ports in an EtherChannel are treated as a single Spanning Tree instance – they all either block or forward as a group.

To configure Etherchannel you use the channel-group command and choose a mode. The “on”, “desirable” and “auto” modes are Cisco Port Aggregation Protocol (PAgP) modes. You can set both sides to mode “on”, but setting both sides to “desirable” is a Cisco best practice. The absence of any PAgP commands leaves the ports in the default “off” mode. “Active” and “passive” are modes of LACP, Link Aggregation Control Protocol, which is an IEEE standard method of aggregating EtherChannel ports. Issue the following command on both switches:

```
CAT1(config)#interface range f0/13 -14
CAT1(config-if-range)#channel-group 1 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected

CAT1(config-if-range)#channel-group 1 mode desirable
```

Verification:

The output below demonstrates that we have successfully negotiated an EtherChannel:

```
CAT1#show etherchannel summary

Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        u - unsuitable for bundling
        U - in use      f - failed to allocate aggregator
        d - default port

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         PAgP      Fa0/13(P)   Fa0/14(P)
```

We can verify that ports fa0/13 and fa0/14 are not blocked in any VLAN by issuing the command below on each switch:

```
CAT2#sh spanning-tree blocked

Name                 Blocked Interfaces List
-------------------- ------------------------------------
Number of blocked ports (segments) in the system : 0
```

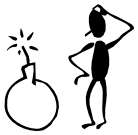
Issue: Use the dot1q protocol for all trunks (not to mention IRB!).

Solution:

1. The default 3550 trunk encapsulation is in "negotiate" mode, and it prefers ISL. The default 3550 trunk mode is dynamic desirable. You are therefore likely to find that ports fa0/13 and fa0/14 have automatically negotiated ISL trunks between CAT1 and CAT2. It is best practice to "nail down" trunk encapsulation and mode on both sides of the trunk. We suggest the following commands on both CAT1 and CAT2 in order to fulfill this requirement:

```
CAT2(config)#interface range f0/13 -14
CAT2(config-if-range)#switchport trunk encapsulation dot1q
CAT2(config-if-range)#switchport mode trunk
```

2. Typical of many CCIE-level requirements, the exercise demands the configuration of Integrated Routing and Bridging (IRB) without directly referring to it. The first thing to notice is that there are three VLANs that need connections to the single Fast Ethernet physical interface on R1, so immediately we see that three subinterfaces will be required, one per VLAN, encapsulated as a dot1q trunk. Remember to trunk interface fa0/1 on CAT2 with dot1q encapsulation.



We find that it is often necessary to turn off trunk negotiation on Catalyst trunk ports when they connect to bridged router ports. This can be done by using the command **switchport nonegotiate**.

3. We notice that VLANs 10 and 20 are in the same IP subnet, so we will have to bridge them together and create a BVI interface. Here are the steps involved:
   a. Create 3 subinterfaces on F0/0, and encapsulate each with a dot1q VLAN. Put an IP address on the VLAN 30 subinterface. See the SHOWiT configuration
   b. Issue the command **bridge 1 protocol IEEE** to create a "VLAN" on R1
   c. Enter **bridge IRB** to activate Integrated Routing and Bridging
   d. Create **Interface BVI1** and assign an IP address.
   e. Remember the command **bridge 1 route IP**. By default, all layer 3 protocols are simply bridged between the interfaces. This command connects the BVI to the bridge for IP traffic. Verify with **show interface IRB**.
   f. On the VLAN 10 and VLAN 20 subinterfaces enter **bridge-group 1**

Verification:

There are many commands that can help you confirm your trunk configuration. In the SHOWiT engine try **show interfaces trunk** on CAT1 and CAT2. Make sure you can ping all of the interfaces within the same Ethernet subnet before moving on.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

Route Redistribution

HIDDEN ISSUES TO SPOT WITH ROUTE REDISTRIBUTION

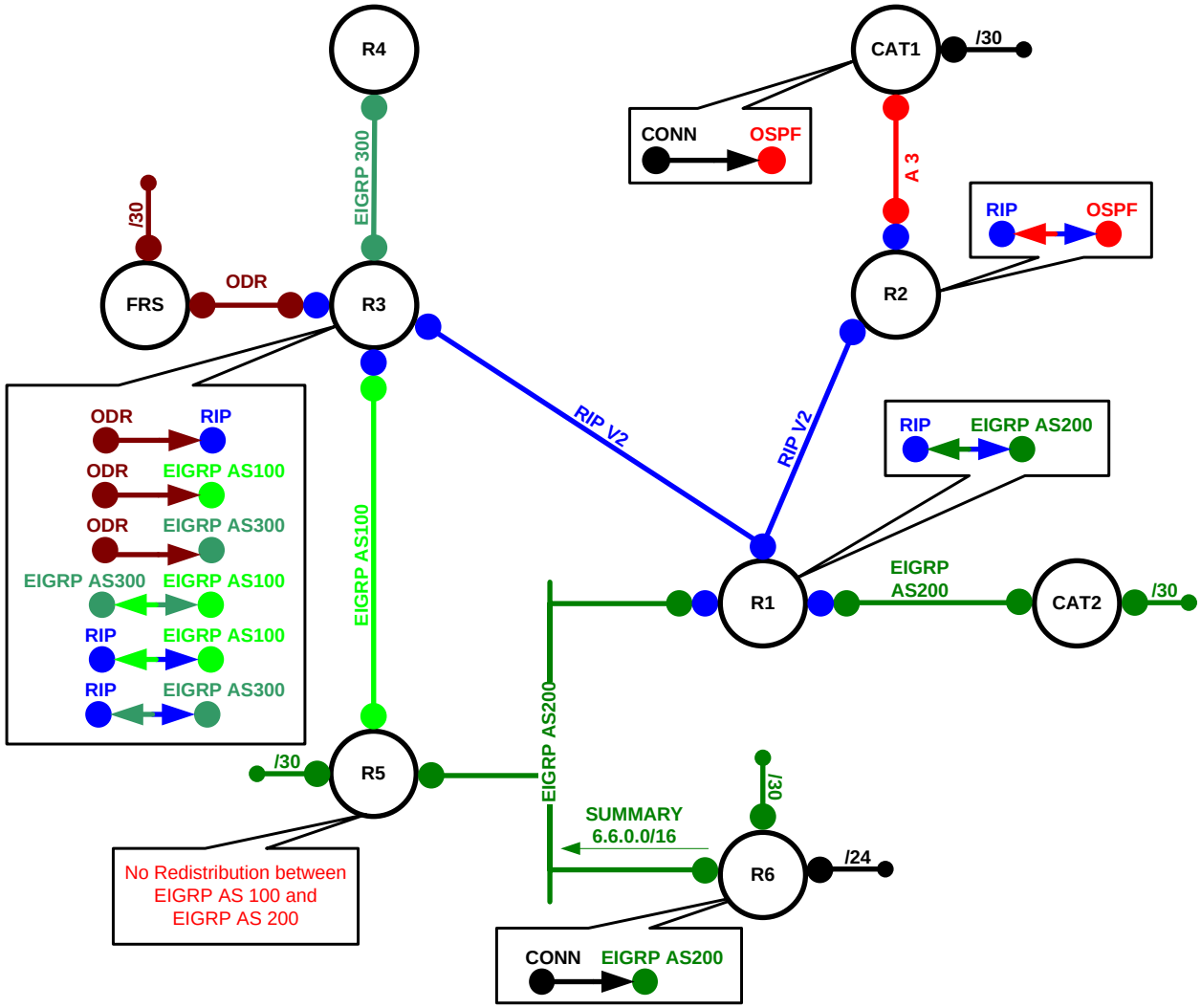
Before examining the specific issues related to configuring each of the IGPs involved in this Scenario, it is important to survey the entire topology and determine how the IGPs will interoperate. Performing such a survey will force us to consider the issues related to route redistribution.

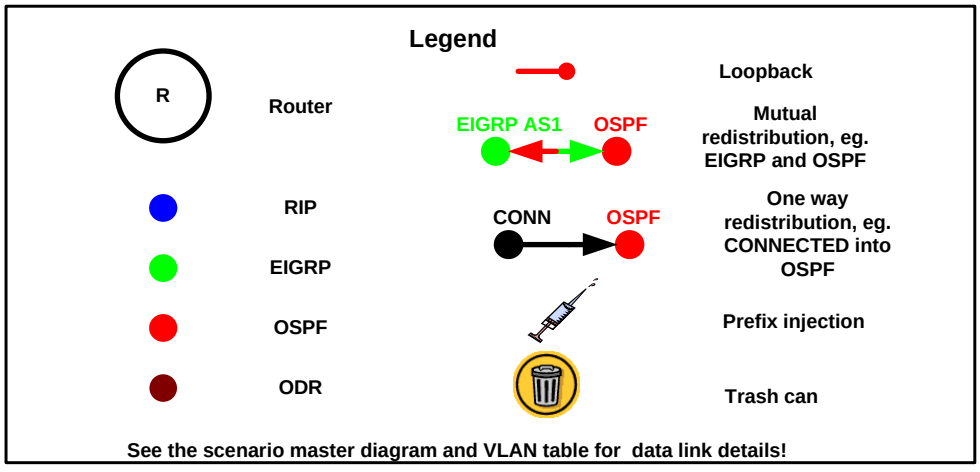
When evaluating a single internetwork topology that contains multiple routing protocols, a good starting point of analysis is to determine whether there is more than one direct or indirect connecting point between two routing protocols. If there is only one connecting point between two routing protocols, providing connectivity between them is relatively simple. If there are two or more connecting points, then providing connectivity between the two routing protocols can be complex. When two or more connecting points exist, you can use them to provide redundancy, load balancing and optimum path selection. However, when two or more connecting points exist, you must also assure, at the very least, that no routing loops exist and, whenever possible, no suboptimal paths are selected.

When evaluating this Scenario's internetwork topology and how the routing protocols have been assigned to it and the scenario redistribution requirements you will see that there exists only one forwarding path between the IP networks. The redistribution topology looks more like a start without the routing feedback and the IGP loops. This scenario is not concerned about the optimal forwarding paths and redundancy. Therefore the redistribution strategy is relatively simple.

The Scenario topology is represented in the following diagram:

NOTE: The colors used in this diagram greatly add to the understanding of redistribution applied to this Scenario. If you print this document, attempt to print this page with a color printer.





Redistribution Table

The following table provides a useful summary of which prefixes were imported into a given routing protocol. Pay special attention to the color-coding of the table. The colors exactly match the colors used in the diagram. Please make note of the following notation used in the redistribution table:

AD = A manually set "administrative distance" value.
M = A manually set "metric" value.
Protocol (Protocol-1, Protocol-2, etc.) When a protocol is followed by additional protocols enclosed within parentheses, this represents that the first protocol list is a transit routing domain for the protocols listed in the parentheses. When the protocol listed first gets redistributed into the target routing domain, it is also redistributes prefixes from the routing protocols listed inside the parenthesis.

Redist Point	Into RIP		Into OSPF		Into EIGRP AS100 and 300	
	PERMIT	DENY	PERMIT	DENY	PERMIT	DENY
R1	EIGRP200				RIP	
R2	OSPF		RIP			
R3	EIGRP AS 100 EIGRP AS 300 ODR				RIP ODR	EIGRP
R5						
R6					CONNECTED	

NOTE: The colors used in this table greatly add to the understanding of redistribution applied to this Scenario. If you print this document, attempt to print this page with a color printer.

Verification:

There are often many possible solutions to the redistribution challenge, but all successful solutions provide stable, optimal paths to all addresses from all addresses. How can you verify these results? Observing the output of the command **debug IP routing** on several of your routers can test the stability of your solution. If, over a period of a few minutes, you see routes being repeatedly deleted and installed, then you probably have a split-horizon or other route-feedback problem. To test the sanity of your paths, consider using the commands **show IP route <protocol>** and **show IP route | include <interface>**. You can use these commands to simplify the output and allow you to focus on what is important. Below you see such output on R1. Comparing this output to your diagram, does it make sense that the routes shown are in the RIP domain? Is Serial 0/0.123 the outgoing interface to these destinations?

```
R1# sh ip route rip
     172.16.0.0/16 is variably subnetted, 14 subnets, 3 masks
R       172.16.200.16/30 [171/3] via 172.16.123.3, 00:00:02, Serial0/0.123
R       172.16.200.0/30 [171/1] via 172.16.123.2, 00:00:04, Serial0/0.123
R       172.16.34.0/25 [171/1] via 172.16.123.3, 00:00:02, Serial0/0.123
```

```
R       172.16.35.0/25 [171/1] via 172.16.123.3, 00:00:02, Serial0/0.123
R       172.16.30.0/25 [171/1] via 172.16.123.3, 00:00:02, Serial0/0.123
R       172.16.20.0/25 [171/1] via 172.16.123.2, 00:00:04, Serial0/0.123
R       172.16.102.0/24 [171/1] via 172.16.123.2, 00:00:04, Serial0/0.123
R       172.16.103.0/24 [171/1] via 172.16.123.3, 00:00:02, Serial0/0.123
```

Perform the ip routing table analysis on the other routers using the SHOWiT engine.

Finally, can you reach all required interfaces from all routers? Have you achieved universal connectivity? The only way to know this for certain is to ping every address from every router. This can be a very tedious process if done manually. Consider running a simple TCL script on your router to automate this process. There is an excellent paper on this in the READiT section of the NetMasterClass.net web site. If you can learn to create a simple ping script and if you will make the effort to type each IP address in your pod once into Notepad, then you can truly know that you have satisfied this requirement. Here is a script that can be typed once and then pasted into each router to test connectivity:

```
tclsh
foreach address {
172.16.10.129
172.16.10.1
172.16.123.1
172.16.101.1
FEC0::101:1
FEC0::15:1
FEC0::123:1
FEC0::122:1
FEC0::16:1
172.16.20.2
172.16.123.2
172.16.102.1
FEC0::102:1
FEC0::123:2
172.16.35.3
172.16.34.3
172.16.30.3
172.16.123.3
172.16.103.1
FEC0::103:1
FEC0::123:3
FEC0::34:3
4.4.4.1
172.16.34.4
172.16.104.1
FEC0::104:1
FEC0::34:4
5.5.5.1
172.16.200.13
172.16.35.5
172.16.10.5
FEC0::105:1
FEC0::15:5
FEC0::156:5
7.7.7.1
172.16.200.9
6.6.127.1
```

6.6.130.1
6.6.145.1
172.16.10.6
FEC0::16:6
FEC0::106:1
172.16.200.17
172.16.30.10
172.16.200.1
172.16.20.10
172.16.10.130
172.16.200.5} {ping $address}

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.3 OSPF

HIDDEN ISSUES TO SPOT WITH THE OSPF CONFIGURATION

Issue: No OSPF Area 0 has been configured in the Exam:

Solution:

There is no Area 0 in this Exam. All OSPF routers in Exam 2 are configured in OSPF Area 3. Since there are no routers configured in any other areas, having all routers configured in a single non-area 0 area is acceptable. If a second non-area 0 area was configured and it was required to communicate with the already configured OSPF area 3, it would not be able to do so.

Issue: Devices on VLAN 40 can become adjacent only after they pass non-clear text authentication. Authenticate on a per-interface basis.

Solution:

Configure OSPF authentication for all OSPF routers on VLAN 40. Since it references non-clear text authentication, you must configure MD5 authentication for VLAN 40 on Router R2 and CAT1. Below you see the commands as issued on R2 F0/0 and CAT1 Interface VLAN 40. Note that the authentication type, message-digest in this case, can be configured directly on the interface or it can be inherited from the area configuration under the OSPF process. This scenario requires the interface configuration

```
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 nmc
```

Verification:

```
R2#show ip ospf int fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.16.20.2/25, Area 3
  Process ID 1, Router ID 172.16.200.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.200.2, Interface address 172.16.20.2
  Backup Designated router (ID) 172.16.200.1, Interface address 172.16.20.10
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 9
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.16.200.1  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
    Message digest authentication enabled
      Youngest key id is 1
R2#
```

Note that there is no area authentication:

```
R2#show ip ospf | begin Area 3
    Area 3
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm last executed 00:02:30.488 ago
        SPF algorithm executed 4 times
        Area ranges are
        Number of LSA 3. Checksum Sum 0x022A16
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
R2#
```

Issue: Advertise the loopback 172.16.200.1 into OSPF without any reference to an OSPF area.

Solution:

If it is going to be added to OSPF without being assigned to an OSPF area, it must be injected into OSPF as an External LSA. Therefore, **redistribute connected** will be configured on router CAT1. Also, either a distribute-list or route-map will be configured to assure that only the 172.16.200.0/30 prefix is injected into OSPF and no other CAT1 connected interface. The SHOWiT output demonstrates the use of the command **distribute-list 1 out connected**, specifying that only the prefix 172.16.200.0 be redistributed from source “connected” into OSPF.

Verification:

Basic verification would include the following commands:

1. Issue the command **show IP OSPF neighbor** on R2 or CAT1 to verify basic connectivity.

2. Use the command **show IP OSPF interface f0/0** on R2 to confirm required authentication.
3. Verify that redistribution of 172.16.200.0 was successful by issuing the command **show IP route OSPF** on R2. You should see the prefix listed as an OSPF E2 route with a /30 mask.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.4 RIP

HIDDEN ISSUES TO SPOT WITH THE RIP CONFIGURATION

Issue: Configure RIP version 2 between R1, R2 and R3. Unicast RIP updates on the Frame Relay network only between the router R1 and R3 and the routers R1 and R2.

Solution:

Note that we have configured RIP version 2 on these routers. RIP version 2 updates are sent the 224.0.0.9 multicast group. To solve the unicast update requirement perform the following two steps:

1. Issue the command passive-interface under the RIP processes on R1, R2 and R3 for each of the interfaces on network 172.16.123.0. This will keep these interfaces from sending the RIP v.2 multicast updates.

2. Configure neighbor statements on each of these routers specifying the 172.16.123.0/25 unicast address of a directly connected neighbor so that unicast updates are sent.

    a. On R1 configure neighbor statements to routers R2 and R3.
    b. On R2 configure a neighbor statement to R1.
    c. On R3 configure a neighbor statement to R1.

Issue: A Split-Horizon issue on router R1:

Solution:

Router R1 is configured with RIP on a multipoint Frame-Relay subinterface. Spilt-horizon is enabled on both point-to-point and multipoint Frame-Relay subinterfaces. Since R1 is the hub of an NBMA hub and spoke topology and the scenario does not allow to configure neighbor relationship between the spokes R2 and R3, spilt-horizon must be disabled on router R1, so that RIP updates from router R2 will be received from router R3 and vice versa. Use the command **no IP split-horizon** on S0/0.123 on R1.

Note: The RIP updates are originated with the TTL = 2, therefore if scenario permitted the neighbors between R2 and R3, the split horizon would not be an issue on R1. In the case of this scenario R2 and R3 should not be permitted to exchange the unicast RIP packets between each other.

Issue: A Split-Horizon issue on router R2:

Solution:

Router R2 is configured with RIP on a physical Frame-Relay subinterface. Spilt-horizon is disabled on physical Frame-Relay interfaces. Since you do not want R2 to send the RIP updates back out the frame relay serial interface you should enable split horizon on the RIP spoke router R2's frame relay interface.

Remember: The split horizon is enabled on the Frame Relay logical (both point-to-point and multipoint) interfaces. The split horizon is disabled on the physical Frame Relay interfaces.

Issue: Exchange RIP updates on Frame-Relay link only.

***Solution*:**

To solve this requirement, perform the following two steps: (1) Configure **passive-interface default** on all RIP configured routers. You want to keep the Frame-Relay interfaces on network 172.16.123.0 in a passive state in this Scenario because you are unicasting updates out these interfaces. For more details on unicasting RIP updates see the paragraph above.

Verification:

Basic verification consists of close examination of the RIP-learned routes. At a minimum, you would want to enter **show IP route rip** on each of the routers R1, R2, R3 and R4. In addition, you might want to issue the command **show IP protocols**, in order to confirm learned gateways and the passive status of the router interfaces. How can you best verify proper operation of split-horizon? The command **show IP interface** will confirm the current status of IP split-horizon. In addition you might want to turn on **debug IP rip** and watch a few update cycles to determine that the router is not relearning a route it advertised.

R1:

```
R1#show ip int serial 0/0.123 | inc horizon
  Split horizon is disabled
R1#
```

R2:

```
R2#show ip int serial 0/0 | inc horizon
  Split horizon is enabled
R2#
```

R3:

```
R3#show ip int serial 0/0.123 | inc horizon
  Split horizon is enabled
R3#

R3#show ip route rip
     6.0.0.0/16 is subnetted, 1 subnets
R       6.6.0.0 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
     172.16.0.0/16 is variably subnetted, 15 subnets, 3 masks
R       172.16.10.128/25 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
R       172.16.200.4/30 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
R       172.16.200.0/30 [120/2] via 172.16.123.2, 00:00:24, Serial0/0.123
R       172.16.200.12/30 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
```

```
R       172.16.200.8/30 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
R       172.16.20.0/25 [120/2] via 172.16.123.2, 00:00:24, Serial0/0.123
R       172.16.10.0/25 [171/1] via 172.16.123.1, 00:00:24, Serial0/0.123
R       172.16.102.0/24 [120/2] via 172.16.123.2, 00:00:24, Serial0/0.123
R3#


R3#show ip protocols | begin "rip"
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 22 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: eigrp 100, rip, odr
  Neighbor(s):
    172.16.123.1
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    Serial0/0.34           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
  Passive Interface(s):
    FastEthernet0/0
    Serial0/0
    Serial0/0.123
    FastEthernet0/1
    Serial0/1
    Loopback103
    Loopback603
  Passive Interface(s):
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.123.2         171      00:13:29
    172.16.123.1         171      00:00:17
  Distance: (default is 120)
    Address         Wild mask       Distance  List
    0.0.0.0         255.255.255.255      171  eigrp-p
```

You can generate a similar output on the other RIP speakers in the SHOWiT engine.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.5 EIGRP

HIDDEN ISSUES TO SPOT WITH THE EIGRP CONFIGURATION

Issue: Configure EIGRP 100 between R3 and R5.

Solution:

Use the EIGRP network/mask router configuration command to limit the number of interfaces that are participating in EIGRP on routers R3 and R5. If you configure EIGRP with the classful network command **network 172.16.0.0** all interfaces assigned with 172.16.0.0 will be included within EIGRP. As a general practice, use the mask option when configuring a network command under the EIGRP process to gain greater control over what interfaces are participating in EIGRP.

Issue: Configure EIGRP AS200 between R1, R5 and R6.

Solution:

Note that the R1 interface included in EIGRP AS 200 will be the BVI interface, discussed in the Catalyst section of this Answer Key, above.

Issue: Use unicast EIGRP packet exchange on the above-mentioned VLANs.

Solution:

Configure neighbor statements under the EIGRP 200 routing process on routers R1, R5, R6 and FRS for the 172.16.10.0/25 subnet. There are three items to be aware of here:

1. *Do not make the EIGRP interface passive*. This will keep the interfaces from forming EIGRP neighbors on these interfaces. This is unlike the configuration of RIP unicast updates.
2. The syntax of the EIGRP neighbor statement requires an outgoing interface.
3. You need to configure a full-mesh of neighbor statements among Routers R1, R5 and R6, which share the same subnet, or you need to disable a split horizon on the BVI interface of R1 if you form the neighbor relationship only between R1 and R5, and R1 and R6. We chose the full mesh neighbor relationship in this answer key

Issue: Advertise the specified loopbacks on R6 to AS200 without using a network statement.

Solution:

Issue the **redistribute connected** command on router R6 under the router EIGRP 200 to bring these loopbacks into the EIGRP process as “external”. Remember to add either a route-map or a distribute-list out configuration to limit only these loopbacks from being redistributed into EIGRP. Without the route-map

or distribute-list out configuration, other unwanted connected interfaces will get redistributed into EIGRP on router R6.

Verification:

```
R6#show ip eigrp topolog 6.6.127.0/24 | inc from Rconnected
  0.0.0.0, from Rconnected, Send flag is 0x0

R6#show ip eigrp topolog 6.6.130.0/24 | inc from Rconnected
  0.0.0.0, from Rconnected, Send flag is 0x0

R6#show ip eigrp topolog 6.6.145.0/24 | inc from Rconnected
  0.0.0.0, from Rconnected, Send flag is 0x0
R6#
```

Issue: Summarize these subnets with the optimal mask.

Solution:

The above-mentioned loopback interfaces will be summarized on the F0/0 interface of router R6 with the **ip summary-address EIGRP** command. The optimal mask will be 6.6.0.0/16. This is not a very optimal mask! However, it is the most "optimal" single summarization statement that can be configured given the three prefix addresses supplied. The difficulty in optimally summarizing the three supplied statements lies in the fact that the first subnet 6.6.127.0/24 is on the other side of the major bit boundary of 6.6.128.0/24 from the other two supplied subnets 6.6.130.0/24 and 6.6.145.0/24. As result, the optimal summarization statement must include the entire 3rd octet.

The summary for the external prefixes will be advertised as internal EIGRP prefix. The summarization will also create a local to R6 summary route to null interface.

Verification:

The first place to start might be by confirming that each router has the appropriate neighbor relationships. Below you see the result of the command **show IP eigrp neighbors** on R1. Note that R1 has neighbor relationships with R6 on both the BVI and ATM interfaces.

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.10.5             BV1               12 01:13:58   20   200  0  73
2   172.16.10.130           Fa0/0.30          14 4d02h       1   200  0  55
1   172.16.10.6             BV1               10 4d02h       5   200  0  76
```

Here is the same information on R5, showing neighbors for AS 100 and AS 200

```
R5#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.35.3             Se1/1             12 01:16:03   25   200  0  85
IP-EIGRP neighbors for process 200
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.10.1             Fa0/0             14 01:14:34   10   200  0  133
0   172.16.10.6             Fa0/0             11 01:14:35    6   200  0  76
```

To verify the summary on R6, check that the routing table on R1 has 6.6.0.0/16 and does not have the longer prefixes.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.6 ODR

HIDDEN ISSUES TO SPOT WITH THE ODR CONFIGURATION

Issue: Configure on demand routing between R3 and FRS.

Solution:

ODR is a feature that provides IP routing for stub sites, with minimum overhead. The overhead of a general, dynamic routing protocol is avoided without incurring the configuration and management overhead of static routing. FRS will be a stub router and R2 will play a role of the hub router.

ODR uses the Cisco Discovery Protocol (CDP) to carry minimal routing information between the hub and stub routers. The stub routers send IP prefixes to the hub router via CDP. ODR supports VLSM (Variable Subnet Length Mask).

In the earlier IOS releases (before 12.3T) the stub router must be configured with the **no ip routing.** In the newer IOS releases the stub router can be ip routing device, it will receive 0.0.0.0/0 network from the hub router and will set up the gateway of last resort to the ODR hub router (see the DOiT Volume II SAMPLE Lab Answer key and the SHOWiT Engine). This scenario requires the stub router FRS to be configured as "no ip routing". The FRS will ARP for the destination IP address and will use R3 as a proxy arp default gateway.

Do not forget to redistribute ODR routing protocol into RIP, EIGRP AS 100 and AS 300 on R3 to provide the reachability to the network advertised from the stub router FRS from the remote locations.

Issue: CDP packets exchange is local between two ports on the wire, therefore FRS and R3 will not see each other as intermediate CDP neighbors.

Solution:

Configure L2 protocol tunneling for the CDP. When protocol tunneling is enabled, edge switches on the inbound side encapsulate Layer 2 protocol packets with a special MAC address and send them across the network. The switches in the network do not process these packets but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP cross the network and are delivered to the device on the outbound interface:

Configuration and Verification:

On the port of the CAT2 connected to Fa0/0 interface of R3 configure:

```
CAT2#show run int fa 0/3
Building configuration...

Current configuration : 145 bytes
!
interface FastEthernet0/3
 description R3  Fa0/0
 switchport access vlan 50
 switchport mode access
 l2protocol-tunnel cdp
 no cdp enable
end
```

Disable CDP on the CAT2's port to stop the exchange between the CAT2 and R3

On the port of the CAT2 connected to E0 interface of FRS configure:

```
CAT2#show run int fa 0/7
Building configuration...

Current configuration : 143 bytes
!
interface FastEthernet0/7
 description FRS  E0
 switchport access vlan 50
 switchport mode access
 l2protocol-tunnel cdp
 no cdp enable
end
CAT2#
```

Disable CDP on the CAT2's port to stop the exchange between the CAT2 and R3

```
FRS#sho cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R3               Eth 0              178         R T       2621      Fas 0/0
FRS#
```

The IP routing is globally disabled on FRS:

```
FRS#show ip route
Default gateway is not set

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
FRS#
```

R3 is configured as ODR router for the network 172.16.0.0:

```
router odr
 network 172.16.0.0
```

```
R3#show ip route odr
     172.16.0.0/16 is variably subnetted, 15 subnets, 3 masks
o       172.16.200.16/30 [160/1] via 172.16.30.10, 00:00:53, FastEthernet0/0
R3#
```

FRS sends ARP requests for the destination IP address:

```
FRS#deb arp
ARP packet debugging is on

FRS#ping 172.16.200.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.200.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/260/1064 ms
FRS#
4d03h: IP ARP: creating incomplete entry for IP address: 172.16.200.5 interface Ethernet0
4d03h: IP ARP: sent req src 172.16.30.10 0010.7b3b.74de,
                  dst 172.16.200.5 0000.0000.0000 Ethernet0
4d03h: IP ARP: rcvd rep src 172.16.200.5 0002.4b62.2660, dst 172.16.30.10 Ethernet0
```

Issue: FRS should be able to ping all IP addresses from the 172.16.0.0/16 address space.

Solution:

This scenario does not require connectivity from FRS to BGP originated prefixes and R6's loopback networks. The FRS is a router with "no ip routing", therefore it will arp for the destination IP address. Some IOS releases will not ARP for the destination IP addresses out of the interface which does not belong to the same classful network as the destination IP address. For example, the Ethernet interface of FRS is part of 172.16.0.0/16 range and the destination is part of 6.0.0.0/8
This behavior may change between the different versions of IOS and the types of hardware.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.7 BGP

HIDDEN ISSUES TO SPOT WITH THE BGP CONFIGURATION

First, read the BGP section and draw a diagram according to AS numbering and peering requirements as an aid in configuration and issue spotting, like the one below.

Issue: Turn synchronization off.

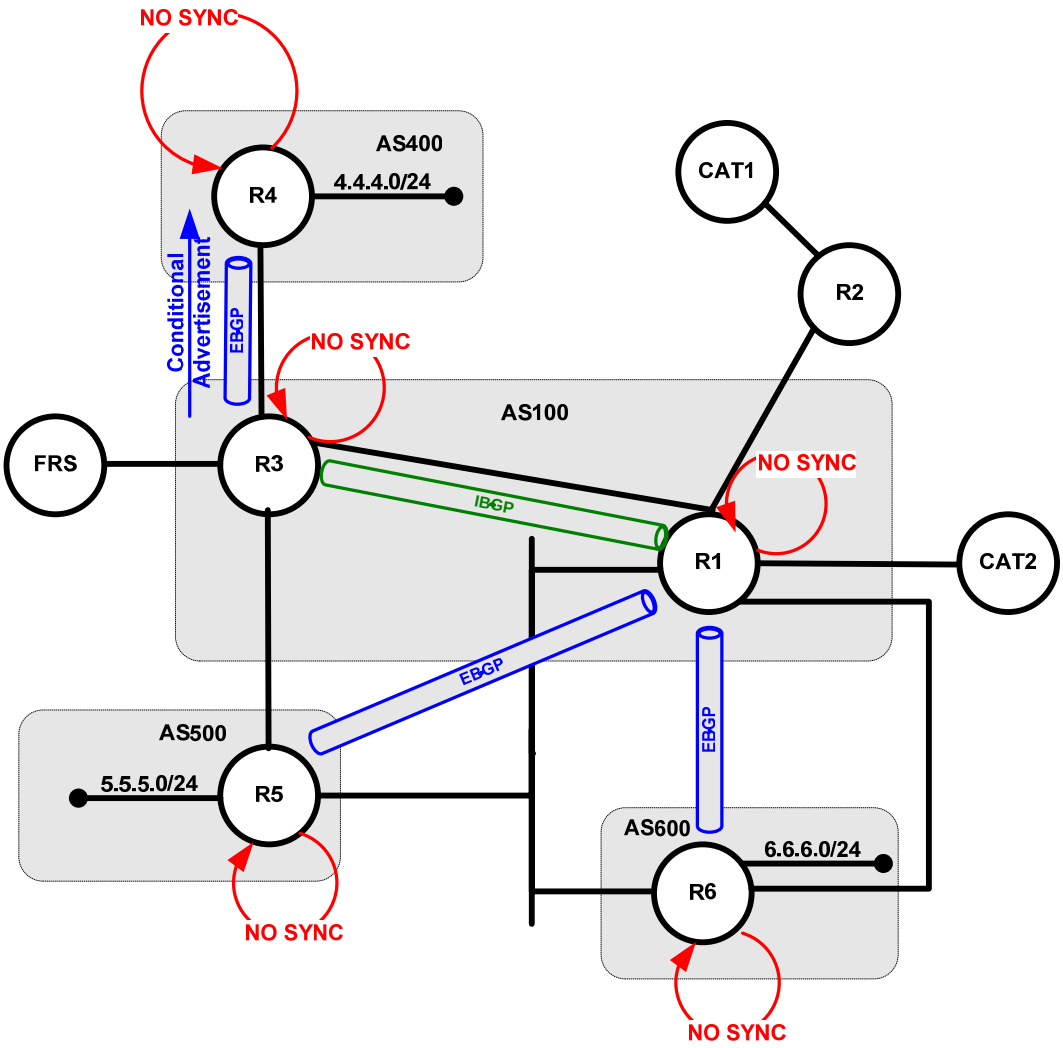
Solution:

Synchronization is purely an IBGP related issue, so it only needs to be disabled on routers R1 and R3. Is Synchronization on or off by default? It depends on the IOS version! You can use the command show ip protocols to determine the current setting.
This scenario does not require to provide a reachability to BGP originated prefixes from all the routers, therefore the redistribution into IGP is not necessary.

Issue: AS 100 should allow transit traffic from network 7.7.7.0/24 to 4.4.4.0/24 only if AS 100 loses reachability to network 5.5.5.0.

Solution:

BGP Topology



This requirement can be met with a BGP conditional advertisement configuration on router R3. In order for AS 100 to provide transit traffic for the 7.7.7.0/24 destination subnet to the source network 4.4.4.0/24 subnet, then AS 100 – specifically the BGP peer relationship between R3 and R4 - must advertise the 7.7.7.0.0/24 subnet to AS 400. To make this advertisement conditional the following BGP neighbor statement must be configured on router R3:

```
neighbor R4 advertise-map Advertise-if non-exist-map If-not-present
```

This neighbor statement references two route-maps: the one called “Advertise-if” matches on the subnet 7.7.7.0/24, and the other called “If-not-present” matches on subnet 5.5.5.0/24. If the 5.5.5.0/24 subnet does **not** exist in the BGP table of router R3, it **will** advertise the 7.7.7.0/24 subnet to router R4.

```
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 172.16.34.4 remote-as 400
 neighbor 172.16.34.4 advertise-map Advertise-if non-exist-map If-not-present
 neighbor 172.16.123.1 remote-as 100
 no auto-summary
!
route-map If-not-present permit 10
 match ip address 17
!
access-list 17 permit 5.5.5.0 0.0.0.255
!
route-map Advertise-if permit 10
 match ip address 18
!
access-list 18 permit 7.7.7.0 0.0.0.255
```

Verification

Basic BGP verification commands include **show IP BGP summary** and **show IP BGP**. To verify whether synchronization has been properly disabled (**sync, no sync**) you can use **show IP protocols**. To test that network 7.7.7.0/24 is only advertised to R4 when network 5.5.5.0/24 is NOT in the forwarding table of R3 try the following sequence:

1. On R3, do **show IP route 5.5.5.0** and confirm that the route is in the local forwarding table.

   ```
   R3#show ip route | inc 5.5.5.0
   B       5.5.5.0 [200/0] via 172.16.10.5, 02:03:16
   R3#
   ```

2. On R4, verify that the route 7.7.7.0 is NOT in the forwarding table.

   ```
   R4#show ip route | inc 7.7.7.0
   R4#
   ```

3. On R5, shut the loopback interface addressed as 5.5.5.1.

   ```
   R5#conf t
   Enter configuration commands, one per line.  End with CNTL/Z.
   ```

```
R5(config)#int lo 5551
R5(config-if)#shut
R5(config-if)#
```

4. After a minute or so, verify on R4 that network 7.7.7.0/24 is now in the local forwarding table.

```
R4#show ip route | inc 7.7.7.0
B       7.7.7.0 [20/0] via 172.16.34.3, 00:00:17
R4#

R4#ping 7.7.7.1 source 4.4.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.1, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 116/123/152 ms
R4#
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.8 IPv6 Addressing

HIDDEN ISSUES TO SPOT WITH THE IPv6 ADDRESSING

Issue: Configure subnet FEC0::123:0/125 on Frame-Relay links between R1, R2 and R3. Use the same interfaces that you have chosen for IPv4 task. Do not use any extra DLCIs above those used for IPv4 configuration. Also configure link local IPv6 addresses.

Solution:

Frame-Relay is already configured for IPv4 at this point, with a logical sub-interface on R3, a logical sub-interface on R1 and a physical interface on R2. R1 has more than one DLCI connected within the same subnet, so the interface is point-to-multipoint.

IPv6 configuration on frame-relay multipoint interfaces requires assigning IP addresses and mapping remote Global and link local IPv6 addresses to DLCIs.:

R1:

```
R1#show run int serial 0/0.123 | inc ipv6
 ipv6 address FEC0::123:1/125
 ipv6 address FE80::1 link-local
 ipv6 rip FRAME-123 enable
 frame-relay map ipv6 FE80::2 102 broadcast
 frame-relay map ipv6 FE80::3 103 broadcast
 frame-relay map ipv6 FEC0::123:2 102
 frame-relay map ipv6 FEC0::123:3 103
```

R2:

```
R2#show run int s0/0 | inc ipv6
 ipv6 address FEC0::123:2/125
 ipv6 address FE80::2 link-local
 ipv6 rip FRAME-123 enable
 frame-relay map ipv6 FEC0::123:1 201
 frame-relay map ipv6 FEC0::123:3 201
 frame-relay map ipv6 FE80::1 201 broadcast
```

R3:

```
R3#show run int s0/0.123 | inc ipv6
 ipv6 address FEC0::123:3/125
 ipv6 address FE80::3 link-local
 ipv6 rip FRAME-123 enable
 frame-relay map ipv6 FEC0::123:1 301
 frame-relay map ipv6 FEC0::123:2 301
 frame-relay map ipv6 FE80::1 301 broadcast
```

Verification:

First, verify that the Frame-Relay mapping is set up correctly:

```
R1#sh frame-relay map
Serial0/0.123 (up): ipv6 FE80::2 dlci 102(0x66,0x1860), static,
              broadcast,
              CISCO, status defined, active
Serial0/0.123 (up): ipv6 FE80::3 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active
Serial0/0.123 (up): ipv6 FEC0::123:2 dlci 102(0x66,0x1860), static,
              CISCO, status defined, active
Serial0/0.123 (up): ipv6 FEC0::123:3 dlci 103(0x67,0x1870), static,
```

Ping the remote global IPv6 addresses to verify that you have subnet level reachability:

```
R1#ping fec0::123:2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::123:2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R1#ping fec0::123:3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::123:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

Ping the remote link local IPv6 addresses to verify that you have subnet level reachability:

```
R1#ping fe80::2
Output Interface: serial0/0.123
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

```
R1#ping fe80::3
Output Interface: serial0/0.123
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::3, timeout is 2 seconds:
Packet sent with a source address of FE80::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

Issue: Configure subnet FEC0::34:0/125 on the Frame-Relay link between R3 and R4. Use the same interfaces that you chose for the IPv4 task. Do not use any extra DLCIs other than those used for the IPv4 configuration. Also configure link local IPv6 addresses.

Solution:

The Frame-Relay link between R3 and R4 is connected to a point-to-point interface on R3 and a physical interface on R4. R3's part will be configured with a IPv6 address, and the existing **frame-relay interface-dlci** command will take care of forwarding.

On R4, mapping needs to be configured for R3's IPv6 routable and link-local addresses.

R3:

```
interface Serial0/0.34 point-to-point
 ipv6 address FEC0::34:3/125
 ipv6 address FE80::3 link-local
 frame-relay interface-dlci 304
```

R4:

```
interface Serial0/0
 encapsulation frame-relay
 ipv6 address FEC0::34:4/125
 ipv6 address FE80::4 link-local
 frame-relay map ipv6 FE80::3 403 broadcast
 frame-relay map ipv6 FEC0::34:3 403
```

Verification:

First, verify that Frame-Relay mapping is set up correctly:

```
R4#show frame map
Serial0/0 (up): ipv6 FE80::3 dlci 403(0x193,0x6430), static,
             broadcast,
             CISCO, status defined, active
Serial0/0 (up): ipv6 FEC0::34:3 dlci 403(0x193,0x6430), static,
        CISCO, status defined, active
```

Ping the remote IPv6 addresses to verify that you have subnet level reachability:

```
R4#ping fec0::34:3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::34:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

```
R4#ping fe80::3
Output Interface: serial0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
R4#
```

Issue: Configure subnet FEC0::122:0/125 on the R1 interface that belongs to Vlan 30.

Solution:

To configure IPv6 on FastEthernet0/0.30, just add the IPv6 address command to the interface configuration:

```
interface FastEthernet0/0.30
 encapsulation dot1Q 30
 ipv6 address FEC0::122:1/125
```

Verification:

Issue: the show ipv6 interface FastEthernet0/0.30 command to see the IPv6 settings:

```
R1#sh ipv6 inte fa0/0.30
FastEthernet0/0.30 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2B0:64FF:FEEF:9BC1
  Global unicast address(es):
    FEC0::122:1, subnet is FEC0::122:0/125
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::1:FF22:1
    FF02::1:FFEF:9BC1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Issue: Configure interfaces Loopback60X with IPv6 address FEC0::10X:1/126, where X is the number of the router (1 for R1, 2 for R2, …, 6 for R6). Do not configure loopbacks on FRS and CAT switches.

Solution:

Configure loopback interface by issuing the following commands on all aforementioned routers (example shows R1):

R1:

```
interface Loopback601
 no ip address
 ipv6 address FEC0::101:1/126
```

These loopbacks will later be advertised via the IGPs and must be reachable from everywhere along with all other interfaces.

Issue: On R6 configure interfaces Loopback610 – Loopback614 using IPv6 addresses 2001:0:0:6::/64 - 2001:0:0:10::/64, using an EUI-64 interface ID in the low-order 64 bits of the address.

Solution:

Configure loopbacks with global aggregatable addresses on R6. They will be used later in an IGP task. Use the following command (shown for Loopback610 as an example):

R6:

```
interface Loopback610
 ipv6 address 2001:0:0:6::/64 eui-64
!
interface Loopback611
 ipv6 address 2001:0:0:7::/64 eui-64
!
interface Loopback612
 ipv6 address 2001:0:0:8::/64 eui-64
!
interface Loopback613
 ipv6 address 2001:0:0:9::/64 eui-64
!
interface Loopback614
 ipv6 address 2001:0:0:10::/64 eui-64
!
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.9 IPv6 Routing

HIDDEN ISSUES TO SPOT WITH THE IPv6 ROUTING CONFIGURATION

Issue: Configure IPv6 RIP FRAME-34 on subnet FEC0::34:0/125. Add loopback 603 and 604 IPv6 subnets to IPv6 RIP FRAME-34

Solution:

To set up RIP for IPv6 you just enable it on the interface. RIP will not take a subnet as a parameter, but will inherit it from the interface.

R3:

```
R3#show run int s0/0.34 | inc ipv6
 ipv6 address FEC0::34:3/125
 ipv6 address FE80::3 link-local
 ipv6 rip FRAME-34 enable

R3#show run int lo 603 | inc ipv6
 ipv6 address FEC0::103:1/126
 ipv6 rip FRAME-34 enable
R3#

R3#show run | inc ipv6 router rip FRAME-34
ipv6 router rip FRAME-34
R3#
```

R4:

```
R4#show run int s0/0 | inc ipv6
 ipv6 address FEC0::34:4/125
 ipv6 address FE80::4 link-local
 ipv6 rip FRAME-34 enable
 frame-relay map ipv6 FE80::3 403 broadcast
 frame-relay map ipv6 FEC0::34:3 403
R4#

R4#show run int lo 604 | inc ipv6
 ipv6 address FEC0::104:1/126
 ipv6 rip FRAME-34 enable
R4#

R4#show run | inc ipv6 router rip
ipv6 router rip FRAME-34
R4#
```

Note: the split-horizon is a property of the RIP process and not the interface as in IPv4.
IPv6 RIP split horizon is on by default:

```
R4#show ipv6 rip
RIP process "FRAME-34", port 521, multicast-group FF02::9, pid 105
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
```

```
    Periodic updates 19172, trigger updates 7
  Interfaces:
    Serial0/0
    Loopback604
  Redistribution:
    None
R4#
```

Split-horizon is controlled per IPv6 RIP instance (FRAME-34 for example).

Issue: Configure IPv6 RIP FRAME-123 on subnet FEC0::123:0/125. **Add loopback 602 IPv6 subnet to IPv6 RIP FRAME-123**

Solution:

Configure IPv6 RIP routing instance FRAME-123 on all Frame Relay interfaces connected to a subnet FECO::123/125. Make sure you disable split horizon fro RIPv6 RIP FRAME-123 on R1 to enable the routing updates exchange between two spokes R2 and R3

Configuration and Verification:

R1:

```
R1#show run int s0/0.123 | inc ipv6
 ipv6 address FEC0::123:1/125
 ipv6 address FE80::1 link-local
 ipv6 rip FRAME-123 enable

ipv6 router rip FRAME-123
  no split-horizon
```

R2:

```
R2#show run int s0/0 | inc ipv6
 ipv6 address FEC0::123:2/125
 ipv6 address FE80::2 link-local
 ipv6 rip FRAME-123 enable

R2#show run | inc ipv6 router rip
ipv6 router rip FRAME-123
R2#
```

R3:

```
R3#show run int s0/0.123 | inc ipv6
 ipv6 address FEC0::123:3/125
 ipv6 address FE80::3 link-local
 ipv6 rip FRAME-123 enable

R3#show run | inc ipv6 router rip FRAME-123
ipv6 router rip FRAME-123
R3#
```

Issue: Configure IPv6 OSPF area 0 on FastEthernet subinterface of R1 that belongs to Vlan 30, on R1s Loopback601, on R5s Loopback605. Configure IPv6 OSPF area 0 on the link between R1 and R5. Use IPv6 addresses FEC0::15:1/112 and FEC0::15:5/112 between R1 and R5; encapsulate IPv6 packets within IPv4 packets for transmission across the IPv4 link between R1 and R5.

Solution:

Implement IPv6 in an IPv4 tunnel between R1 and R5. Use Ethernet IPv4 addresses as tunnel endpoint addresses.

Configuration of OSPF for IPv6 includes routing process configuration and interface configuration. If you don't have any IPv4 interfaces on a router, you must also set up a router id for OSPF. As usual, loopbacks are host networks and will be advertised with a /128 mask, unless specifically configured for network type (usually, point-to-point).

R1:

```
interface Loopback601
 no ip address
 ipv6 address FEC0::101:1/126
 ipv6 ospf network point-to-point
 ipv6 ospf 100 area 0

interface FastEthernet0/0.30
 encapsulation dot1Q 30
 ip address 172.16.10.129 255.255.255.128
 ipv6 address FEC0::122:1/125
 ipv6 ospf 100 area 0

ipv6 router ospf 100
 log-adjacency-changes
```

R5:

```
interface Loopback605
 no ip address
 ipv6 address FEC0::105:1/126
 ipv6 ospf network point-to-point
 ipv6 ospf 100 area 0

ipv6 router ospf 100
 log-adjacency-changes
```

To configure the tunnel, create Tunnel interfaces on R1 and R5, using the IPv4 addresses of BVI1 on R1 and FastEthernet0/0 of R5 as source and destination:

R1:

```
interface Tunnel600
 no ip address
 ipv6 address FEC0::15:1/112
 ipv6 ospf 100 area 0
 tunnel source 172.16.10.1
 tunnel destination 172.16.10.5
 tunnel mode ipv6ip
```

R5:

```
interface Tunnel600
 no ip address
 ipv6 address FEC0::15:5/112
 ipv6 ospf 100 area 0
 tunnel source FastEthernet0/0
```

```
 tunnel destination 172.16.10.1
 tunnel mode ipv6ip
```

Verification:

Verify that R1 and R5 are adjacent and that you can ping both loopbacks from both routers:

```
R1#sh ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
172.16.200.13     1   FULL/  -        00:00:35    28              Tunnel600

R5#sh ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
172.16.123.1      1   FULL/  -        00:00:32    20              Tunnel600

R1#ping fec0::15:5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::15:5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

R5#ping fec0::15:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::15:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Make sure that you see both router IDs in OSPF database Router part for area 0:

```
R1#sh ipv6 ospf 100 0 database router

            OSPFv3 Router with ID (172.16.123.1) (Process ID 100)

                Router Link States (Area 0)

  LS age: 195
  Options: (V6-Bit E-Bit R-bit DC-Bit)
  LS Type: Router Links
  Link State ID: 0
  Advertising Router: 172.16.123.1
  LS Seq Number: 80000046
  Checksum: 0xE643
  Length: 40
  Area Border Router
  AS Boundary Router
  Number of Links: 1

    Link connected to: another Router (point-to-point)
      Link Metric: 11111
      Local Interface ID: 20
      Neighbor Interface ID: 28
      Neighbor Router ID: 172.16.200.13


  Routing Bit Set on this LSA
  LS age: 1913
  Options: (V6-Bit E-Bit R-bit DC-Bit)
  LS Type: Router Links
```

```
  Link State ID: 0
  Advertising Router: 172.16.200.13
  LS Seq Number: 8000003A
  Checksum: 0x1325
  Length: 40
  Area Border Router
  Number of Links: 1

    Link connected to: another Router (point-to-point)
      Link Metric: 11111
      Local Interface ID: 28
      Neighbor Interface ID: 20
      Neighbor Router ID: 172.16.123.1
```

Issue: Configure IPv6 OSPF area 56 on link between R1 and R6. Use IPv6 addresses FEC0::16:1/112 and FEC0::16:6/112 between R1 and R6; encapsulate IPv6 packets within IPv4 GRE packets for a transmission across the IPv4 link between R1 and R6. Do not allow Type 5 LSA's into area 56. Configure IPv6 OSPF area 56 on Fa0/0 interface of R5.

There is an issue is of 2 unconnected areas 56. From OSPF perspective when 2 areas are not connected the fact that the same number is assigned is nonessential. Note that only one area 56 needs to be NSSA, while for the other one no further restriction are present. For R1-R6 NSSA area 56 make sure that R1 introduces IPv6 default router into the area so R6 has reachability to external hosts.

Solution:

Configure a GRE tunnel between R1 and R6:

R1:

```
interface Tunnel601
 no ip address
 ipv6 address FEC0::16:1/112
 ipv6 ospf 100 area 56
 tunnel source 172.16.10.1
 tunnel destination 172.16.10.6
```

R6:

```
interface Tunnel601
 no ip address
 ipv6 address FEC0::16:6/112
 ipv6 ospf 100 area 56
 tunnel source 172.16.10.6
 tunnel destination 172.16.10.1
```

Note: The GRE mode is the default mode for the tunnel.

Configure OSPF NSSA area 56 on the tunnel link between R1 and R6. NSSA configuration is done under global **ipv6 router ospf** configuration:

```
ipv6 router ospf 100
 log-adjacency-changes
 area 56 nssa  default-information-originate
```

Issue: Loopbacks on R6 must be advertised via OSPF as external prefixes. Aggregatable global addresses must be summarized into the best single prefix. You must not see any of the aggregatable global networks on other IPv6 routers except for the summary.

Solution:

The current global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). Addresses with a prefix of 2000::/3 through E000::/3 are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. You must configure these prefixes using the command **ipv6 address 2001:0:0:N::/64** (where N is a number 6, 7, 8, 9 or 10, called the SLA – Site Level Aggregator):

```
interface Loopback610
 no ip address
 ipv6 address 2001:0:0:6::/64 eui-64
```

IOS will substitute the host part of the address with the interface ID, so when you do **show ipv6 interface loopback610** you will see the actual globally unique unicast address.

```
R6#sh ipv6 inte lo610
Loopback610 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2B0:64FF:FE81:E7A0
  Global unicast address(es):
    2001::6:2B0:64FF:FE81:E7A0, subnet is 2001:0:0:6::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF81:E7A0
  MTU is 1514 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

After you have configured the loopbacks with global addresses, you need to redistribute connected prefixes into OSPF on R6. This is done under ipv6 router ospf configuration. Since area 56 is NSSA, these networks will be flooded thru area 56 as Type-7 LSAs.

There is only one ABR here, so there will not be any race conditions in area 56 to determine which ABR will perform the Type-7 to Type-5 conversion:

```
R6#sh ipv6 ospf
 Routing Process "ospfv3 100" with ID 172.16.200.9
 It is an autonomous system boundary router

R1#sh ipv6 ospf
 Routing Process "ospfv3 100" with ID 172.16.123.1
 It is an area border and autonomous system boundary router
 [skipped]
    Area 56
        Number of interfaces in this area is 1
        It is a NSSA area
        Perform type-7/type-5 LSA translation
```

Otherwise, the ABR with the highest OSPF RID will perform the conversion.

Summarization must be done on R6 to prevent all other IPv6 routers from seeing the more specific globally aggregatable prefixes. By looking at these prefixes, it seems that they all fall into 2001::0/60, but that's not the case. If for IPv4 subnets 6,7,8,9 and 10 will fall in the range of 16 subnets, but IPv6 addresses use HEXADECIMAL digits, so 10 is actually 7 steps away from 9. The aggregate will be 2001::0/59, and you can configure it in **ipv6 router ospf** configuration mode:

```
ipv6 router ospf 100
 summary-prefix 2001::/59
```

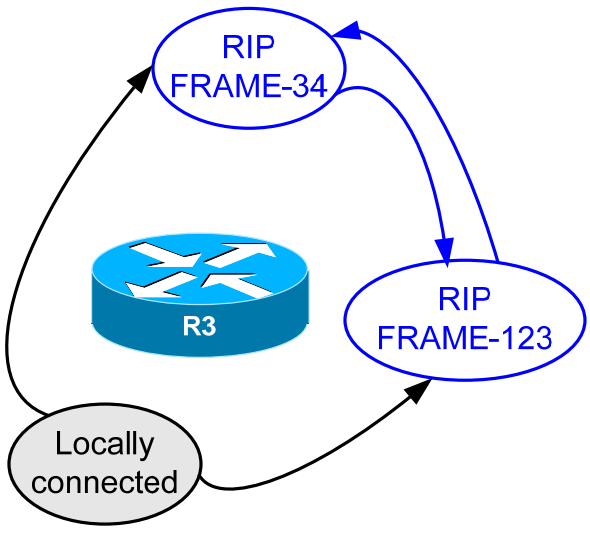
Verification:

To verify that the NSSA area is configured correctly and that summarization is working, issue the **show ipv6 route ospf** command on router R1:

```
R1#sh ipv6 route ospf
IPv6 Routing Table - 18 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
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON2   2001::/59 [110/20]
      via FE80::2B0:64FF:FE81:E7A0, ATM3/0.16
O    FEC0::105:0/126 [110/11112]
      via FE80::AC10:A05, Tunnel600
ON2   FEC0::106:0/126 [110/20]
      via FE80::2B0:64FF:FE81:E7A0, ATM3/0.16
```

We have two ON2 NSSA external prefixes from R6 (one summary, and one site-local loopback), and one OSPF prefix from R5 for R5's loopback.

Issue: Mutually redistribute RIP instances on R3.



Solution:

Redistribution for IPv6 is slightly different from IPv4. First of all, connected networks are not redistributed, even if they are part of the routing protocol.

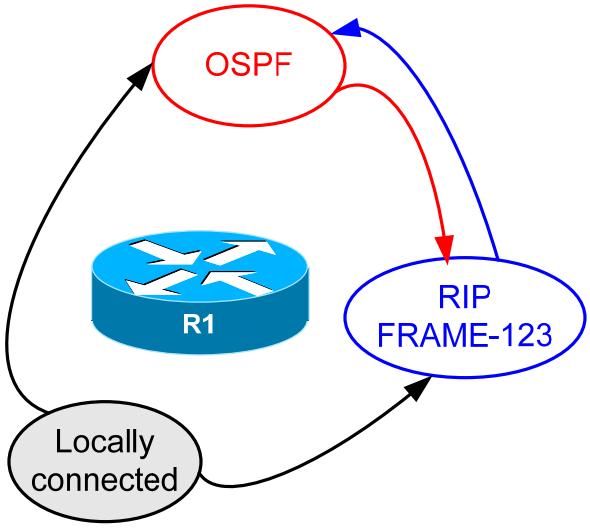
Redistribution between two instances of IPv6 RIP must be performed on R3 according to a diagram above.

```
ipv6 router rip FRAME-34
 redistribute connected metric 1
 redistribute rip FRAME-123 metric 1
!
ipv6 router rip FRAME-123
 redistribute connected metric 1
 redistribute rip FRAME-34 metric 1
!
```

IPv6 RIP table on R4:

```
R4#show ipv6 route rip
IPv6 Routing Table - 17 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
R   2001::/59 [120/2]
     via FE80::3, Serial0/0
R   FEC0::15:0/112 [120/2]
     via FE80::3, Serial0/0
R   FEC0::16:0/112 [120/2]
     via FE80::3, Serial0/0
R   FEC0::101:0/126 [120/2]
     via FE80::3, Serial0/0
R   FEC0::102:0/126 [120/2]
     via FE80::3, Serial0/0
R   FEC0::103:0/126 [120/2]
     via FE80::3, Serial0/0
R   FEC0::105:0/126 [120/2]
     via FE80::3, Serial0/0
R   FEC0::106:0/126 [120/2]
     via FE80::3, Serial0/0
R   FEC0::122:0/125 [120/2]
     via FE80::3, Serial0/0
R   FEC0::123:0/125 [120/2]
     via FE80::3, Serial0/0
R   FEC0::156:0/125 [120/2]
     via FE80::3, Serial0/0
R4#
```

Issue: Mutually redistribute RIP and OSPF on R1.



Solution:

Redistribution is performed according to the diagram above:

```
ipv6 router ospf 100
 log-adjacency-changes
 area 56 nssa default-information-originate
 redistribute connected metric 1 route-map Connected->OSPF
 redistribute rip FRAME-123
!
ipv6 router rip FRAME-123
 redistribute connected metric 1
 redistribute ospf 100 metric 1
  no split-horizon
!
route-map Connected->OSPF permit 10
 match ipv6 address Connected->OSPF
!
ipv6 access-list Connected->OSPF
 permit ipv6 FEC0::123:0/125 any
!
```

It is not essential to control the redistribution of the connected prefixes into OSPF. We just show how to use the route-map approach if you need to.

Verification:

To verify that redistribution was successful, you might check that all prefixes are listed in all of the routing tables. Also, all prefixes must be reachable from all IPv6 routers. One could also create a TCL script to ping each IPv6 address from each router.

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.10 Router Maintenance

HIDDEN ISSUES TO SPOT WITH THE ROUTER MAINTENANCE CONFIGURATION

Issue: Provide access for the Sun Diskless workstations on VLAN 40 attached to router R2 to the Sun Server attached to VLAN 50 on router R3. Provide a solution that does not require a DHCP server.

Solution:

An alternative to a DHCP server is a RARP server. Perhaps configuring a Cisco router as a RARP server can solve this task. Since the Sun workstations are diskless, they have no place to store their basic IP connectivity parameters such as an IP address and will generate RARP (Reverse ARP) requests. Router R2, which is directly connected to the same VLAN as the diskless workstations – VLAN 40, can act as a RARP server for the workstations.

To configure router R2 as an RARP server, complete the following steps:

1. Enter the interface configuration command **ip rarp-server X.X.X.X**, where X.X.X.X is the IP address of the Sun Server, 172.16.30.100.
2. Configure two static arp entries on R2 to map the IP addresses to the MAC addresses of the two diskless workstations. For example: **arp 172.16.40.100 0800.20ac.24b8 ARPA**
3. Once the Sun diskless workstations have received their IP address from the RARP server, they must communicate with the Sun Server via Sun Remote Procedure Call protocol on UDP port 111. To complete our configuration of this task, we need to include global configuration command **ip forward-protocol UDP 111** (the IOS may translate 111 to a keyword sunrpc) and interface-level command **ip helper-address 172.16.30.100.**

Check out the link below for more on configuring RARP.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt2/fcd206.htm

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.11 Security

HIDDEN ISSUES TO SPOT WITH THE SECURITY CONFIGURATION

Issue: Allow the Finger application to R4 from routers R1, R2 and R5. Disallow Finger from R4 to routers R1, R2 and R5. Do not use the "established" keyword.

Solution:

Finger is an application that uses TCP port 79. This task allows routers R1, R2 and R5 to establish a TCP connection on router R4 for the Finger application; however router R4 cannot establish a Finger connection to routers R1, R2 and R5. Since you cannot use the extended access-list "established" parameter, you should consider a reflexive access-list.

Routers R1, R2 and R5 are defined to be on the trusted side of Lab2 network, and Router R4 is defined to be on the untrusted side of the Exam 2 network. The reflexive access-list will be configured on router R3, the router at the boundary of the trusted and untrusted parts of the Exam 2 network, and will allow routers on the trusted side of the network to open TCP connection with untrusted devices without using the extended access-list "established" parameter.

Reflexive access lists generate temporary access-list entries, which are automatically created when a new IP session begins (for example, by an outbound packet). The temporary reflexive access-list entry will "reflect" the source and destination IP address as well as the source and destination TCP ports of the outbound Finger traffic. The word "reflect" means that the IP addresses and the TCP ports of the outbound Finger traffic will be reversed (or "reflected") and will be matched for the inbound, return-path, Finger traffic. When the Finger session ends, the temporary reflexive access-list entry will be removed automatically.

The access-list "FingerfromR4" is applied inbound on the R3 link facing R4. It does the following:

1. Contains an "evaluate" statement that will permit the return-path Finger traffic from R4.
2. Explicitly denies return Finger traffic from R4. This will be overridden as required by the evaluate statement above.
3. Denies direct finger traffic to any of the R1, R2 and R5 interfaces.
4. Permits all other traffic.

The access-list "FingertoR4" is applied outbound on the port facing R4. It does the following:

1. Permits Finger traffic from any interface on R1, R2 or R5 and calls the "reflect" keyword to dynamically permit return Finger traffic from R4.
2. Denies return Finger traffic to R4.
3. Permits all other traffic.

Verification:

You can test the access-list as follows:

1. On R4 issue the command **ip finger**.

2. From router R1, R2 or R5 telnet to R4 on port 79 and you will generate Finger output. The session will open for a few seconds, then close, displaying the output of **show users** on R4, for example:

```
R1#telnet 172.16.34.4 finger
Trying 172.16.34.4, 79 ... Open

    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:06:59
* 66 vty 0                 idle                 00:00:00 172.16.123.1

  Interface    User               Mode         Idle     Peer Address

[Connection to 172.16.34.4 closed by foreign host]
R1#
```

3. The temporary access list entry will be created on R3 to permit the return traffic from R4. This entry is removed as soon as the session closes, so it may be hard to capture. Here is a portion of **show access-list** output performed on R3 just as a session from R2 ended:

```
 Reflexive IP access list Finger
Extended IP access list FingertoR4
    10 permit tcp host 172.16.123.1 host 172.16.34.4 eq finger reflect Finger (19 matches)
```

The access list blocks the finger attempt the opposite direction from R4 to R1:

```
R4#telnet 172.16.123.1 finger
Trying 172.16.123.1, 79 ...
% Destination unreachable; gateway or host down

R4#

Reflexive IP access list Finger
Extended IP access list FingerfromR4
    10 evaluate Finger
    20 deny tcp host 172.16.34.4 eq finger any
    30 deny tcp host 4.4.4.1 eq finger any
    40 deny tcp host 172.16.34.4 host 172.16.123.1 eq finger (6 matches)
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.12 QoS

HIDDEN ISSUES TO SPOT WITH THE QOS CONFIGURATION

Issue: Assign DSCP value 45 to frames received by CAT1's fa0/11 and fa0/12 ports.

Solution:

This requirement cannot be fulfilled without considering the requirements in the next section. Incoming traffic will be tagged with CoS values 3, 4, and 5. Use the statement below to customize the default cos-dscp map so that these CoS values will be mapped to DSCP value 45.

```
CAT1(config)#mls qos map cos-dscp 0 8 16 45 45 45 48 56
```

Verification:

```
CAT1#sh mls qos map cos-dscp
   Cos-dscp map:
        cos:   0  1  2  3  4  5  6  7
      --------------------------------
        dscp:  0  8 16 45 45 45 48 56
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.13 Catalyst Specialties

HIDDEN ISSUES TO SPOT WITH THE CATALYST SPECIALTIES

Issue: Only one workstation and one IP phone are permitted to connect through each fa0/11 and fa0/12 on Cat1.

Solution:

The four permitted devices (two permitted on one port of CAT1) in the conference room are known by their MAC addresses. Use the port security feature. It entails enabling port security, setting the maximum number of permitted addresses, and specifying the permitted addresses. Static port security is not generally allowed on voice VLANs, but it is permitted when the voice VLAN is dot1p.

Note that in case when IP phone uses dot1q (which is not a case in this scenario) on Catalyst 3550 platform you must set the maximum allowed secure addresses on the port to two plus the maximum number of secure addresses allowed on the access VLAN. In other words with dot1q encapsulation the

phone will use 2 MAC addresses for purposes of port security, as switch will count phone's MAC address for data VLAN and then for voice VLAN.

```
CAT1#show run int fa0/11 | inc security
 switchport port-security
 switchport port-security maximum 2
 switchport port-security mac-address 0007.8595.d1a7
 switchport port-security mac-address 0800.20ac.24b8
CAT1#

CAT1#show run int fa0/12 | inc security
 switchport port-security
 switchport port-security maximum 2
 switchport port-security mac-address 0007.8595.d2b7
 switchport port-security mac-address 0080.20ac.24b9
CAT1#
```

Issue: Configure Priority-tagged frames to carry the voice traffic on the native VLAN

Solution:

Enable **MLS QOS** in global configuration mode, and then enter the interface-level command **switchport voice vlan dot1p**. This command will apply 802.1p tagging for voice traffic, which means that frames will come in with COS but no VLAN tag. The voice data traffic will arrive from the Cisco IP phone with the 802.1p COS value of 5, and the voice control traffic will arrive with COS 3.

Issue: Override the COS value in frames from the SUN workstations with a COS of 4.

The default COS setting for frames generated by the Sun workstations is 0. You can override the default COS setting to the Sun workstation traffic with the following Catalyst 3550 port command **switchport priority extend cos 4**. You can verify these settings with the following show command: **show interface fa0/X switchport**.

```
interface FastEthernet0/11
 description IP-Phone, Sun station
 switchport access vlan 40
 switchport mode access
 switchport voice vlan dot1p
 switchport priority extend cos 4
 mls qos trust cos

interface FastEthernet0/12
 description IP-Phone, Sun station
 switchport access vlan 40
 switchport mode access
 switchport voice vlan dot1p
 switchport priority extend cos 4
 mls qos trust cos
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".

2.14 Gateway Redundancy

HIDDEN ISSUES TO SPOT WITH THE GATEWAY REDUNDANCY CONFIGURATION

Issue: Make sure the mac address associated with the virtual gateway is set to 0000.oc07.ac1a.

Solution:

HSRP pre-sets the first ten hex digits of the mac address it uses to 0000.0c07.acXX. The last two hex-digits of the mac address are derived from the number assigned to the standby-group. For example, if the standby-group was assigned the number "10", the last two hex-digits of the mac address used by HSRP will be 0A. In this task the standby group should therefore be assigned the decimal value of 26. When the decimal value of 26 is translated into hex, it is converted to "1a".

Issue: Authenticate HSRP on the 172.16.10.0/25 subnet, and exchange the HSRP hello packets 3 times faster than by default.

Solution:

HSRP authentication is configured on the BVI interface of R1 and the F0/0 interface of R5 with the command **standby 26 authentication nmc**. The default HSRP hello timer is 3 seconds. Set the HSRP hello timer to 1 second on both router interfaces with the command **standby 26 timers 1 4.** Here we also change the holdtime to 4 seconds.

Issue: Select R5 as the preferred gateway if the FRS connection on R1 goes down. R1 should resume as primary when its Frame-Relay link is active again.

Solution:
Configure the track option on R1 to track the multipoint Serial subinterface on R1. R1 is set as the primary gateway with a priority of 150, so we must assure that we decrement R1's priority by at least 55 when the R1 multipoint subinterface goes down so that it will be less than the priority on R5. Configure R1 with the HSRP **preempt** option to assure that it will become the primary HSRP router when its Frame-Relay connection comes back up. To verify your work, try show standby.

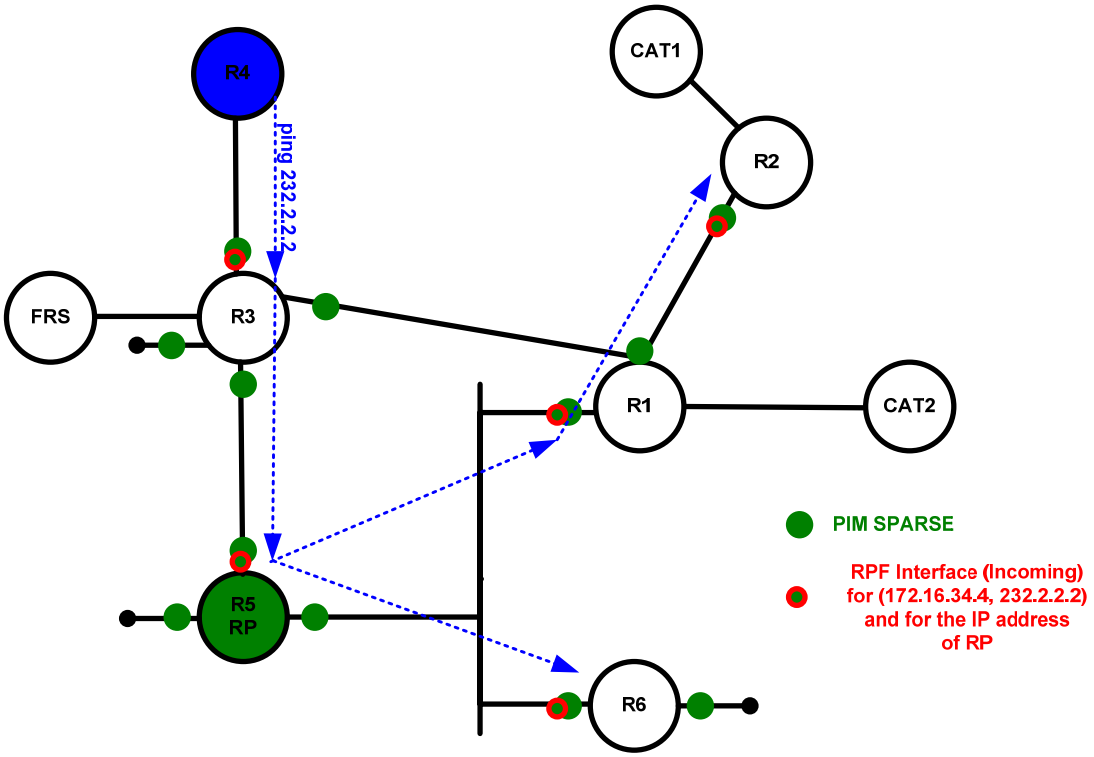
```
R1#show standby
BVI1 - Group 26
  State is Active
    5 state changes, last state change 2d03h
  Virtual IP address is 172.160.10.126
  Active virtual MAC address is 0000.0c07.ac1a
    Local virtual MAC address is 0000.0c07.ac1a (default)
  Hello time 1 sec, hold time 4 sec
    Next hello sent in 0.912 secs
  Authentication text "nmc"
  Preemption enabled
  Active router is local
  Standby router is 172.16.10.5, priority 100 (expires in 3.944 sec)
  Priority 150 (configured 150)
    Track interface Serial0/0.123 state Up decrement 60
```

2.15 Multicast

HIDDEN ISSUES TO SPOT WITH THE MULTICAST CONFIGURATION

Often the best place to start is with a good diagram. Below, the multicast source, Router R4, is represented in blue. The Rendezvous Point, router R5, is represented in green. The (S,G) is represented by the dotted blue line.

Multicast Topology



Issue: Statically configure a "shared" tree rooted from R5.

Solution:

Since a "shared" tree is required in this multicast configuration requirement, PIM Sparse Mode must be configured on all participating routers. PIM Sparse Mode uses two kinds of multicast distribution trees:

1. A shared tree that establishes the Rendezvous Point as the root of the tree. These trees are represented by (*,G) entries in the MROUTE table.
2. A source tree that establishes a specific multicast source as the root. These trees are represented by (S,G) entries in the MROUTE table. The shared tree is used to implement the explicit join model of PIM Sparse mode.

Unlike the "push" technology used in a PIM Dense Mode "flood and prune" model, PIM Sparse Mode uses a "pull" technology. Routers connected to active multicast clients send "explicit join" messages sent to the Rendezvous Point. Each router, therefore, needs to be informed of the address of the RP. There are three basic methods of accomplishing this: static, Auto-RP, and Bootstrap Router. This task specifies static configuration, so on each participating router you would enter the command **ip pim rp-address X.X.X.X** where X.X.X.X is the IP address of the Rendezvous Point. In our case, we are using a loopback address on R5, 172.16.200.13.

Issue: Join group 232.2.2.2 on every member of the Shared Tree.

Solution:

Configure **ip igmp join-group 232.2.2.2** on the specified routers. Whenever you configure this command on an interface, make sure the following three configuration requirements are also met:

1. The interface is configured with an IP address.
2. The IP address is propagated by a unicast routing protocol and
3. Multicast routing is also configured on the same interface with a command such as **ip pim sparse-mode.**
4. For purposes of stability, we recommend that the interface is a loopback interface, though this is not strictly required.

Issue: Make sure you receive replies from all routers listed for multicast pings generated by router R4.

Solution:

Start a long ping to the address 232.2.2.2 on R4. The outgoing interface of R4 will be the source address for the multicast stream. The interfaces on which you configured the **igmp join-group** commands will be the multicast receivers. Make this an extended ping, specifying a large number of pings and the source interface. In the end, you should get responses from 172.16.34.3 (R3), 172.16.35.5 (R5), 172.16.10.1 (R1), 172.16.10.6 (R6) and 172.16.123.2 (R2).

```
R4#ping ip
Target IP address: 232.2.2.2
Repeat count [1]: 9999
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: Serial0/0
Time to live [255]:
[accept defaults]
Type escape sequence to abort.
Sending 9999, 100-byte ICMP Echos to 232.2.2.2, timeout is 2 seconds:

Reply to request 0 from 172.16.34.3, 88 ms
Reply to request 0 from 172.16.123.2, 240 ms
Reply to request 0 from 172.16.10.6, 172 ms
Reply to request 0 from 172.16.10.1, 144 ms
Reply to request 0 from 172.16.35.5, 132 ms
```

Depending on the topology, you may not get all of these responses right away. That's what makes it interesting! Apart from basic errors in configuration, multicast troubleshooting usually focuses on two issues: the reverse-path forwarding check (RPF check) and the outgoing interface list (OILIST). You can use the commands **show IP mroute** and **mtrace** to diagnose multicast problems. Your primary tool to fix them will be a static mroute entry, such as **IP mroute 172.16.34.4 255.255.255.255 172.16.123.1**, where 172.16.34.4 is a source address and 172.16.123.1 is the preferred next hop toward that source. The basic troubleshooting technique we suggest is to start at the multicast source and follow the packet from hop to hop, verifying the RPF interfaces and the outgoing interface lists.

To demonstrate this process, let's assume that we are getting responses like the ones below:

```
R4#ping 232.2.2.2

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 232.2.2.2, timeout is 2 seconds:

Reply to request 0 from 172.16.34.3, 60 ms
Reply to request 0 from 172.16.123.1, 208 ms
Reply to request 0 from 172.16.10.6, 116 ms
Reply to request 0 from 172.16.35.5, 100 ms
```

This output shows that we are getting responses from all but R2. Why not? Here is a portion of the output from **show IP mroute** on R1.

```
(*, 232.2.2.2), 00:09:44/00:02:16, RP 172.16.200.13, flags: SJCL
  Incoming interface: BVI1, RPF nbr 172.16.10.5
  Outgoing interface list:
    Loopback100, Forward/Sparse-Dense, 00:09:44/00:02:19
    Serial0/0.123, Forward/Sparse, 00:09:30/00:02:52

(172.16.34.4, 232.2.2.2), 00:01:05/00:02:58, flags: LJT
  Incoming interface: Serial0/0.123, RPF nbr 172.16.123.3
  Outgoing interface list:
    Loopback100, Forward/Sparse-Dense, 00:01:05/00:02:19
    X                                                   X
```

The (*.G) entry verifies that R1 has learned an RP for this group and is doing a sparse-mode distribution. It also indicates that (*,G) joins have been received on Loopback 100 and S0/0.123, which is the path out toward R2. The problem can be seen in the (S,G) entry. The multicast traffic is arriving on S0/0.123. As a loop-prevention mechanism, PIM does not allow the incoming interface to be placed on the outgoing interface list. S0/0.123 is therefore not showing up on the (S,G) outgoing interface list. The traffic is being received on S0/0.123, but is not being sent back out. If network 123 were Ethernet, this would not be a problem, because the multicast stream could be sent directly from R3 to R2. But this is not possible on a hub and spoke Frame-Relay network. The traffic R2 needs is getting stopped at the hub, R1.

One solution to this problem is to engineer a path like the one pictured in the diagram above. We need the traffic to enter R1 on an interface other than S0/0.123 if we want to send traffic to R2. In fact, the traffic is arriving on the BVI interface of R1, but that traffic fails the RPF check – according to the unicast routing table it is not on the shortest path back to 172.16.34.4. The **mtrace** command can be used to verify the reverse path to a source address, as shown in the output on the next page.

```
R1#mtrace 172.16.34.4
Type escape sequence to abort.
Mtrace from 172.16.34.4 to 172.16.123.1 via RPF
From source (?) to destination (?)
Querying full reverse path...
```

```
 0  172.16.123.1
-1  172.16.123.1 PIM  [172.16.34.0/25]
-2  172.16.123.3 PIM  [172.16.34.0/25]
-3  172.16.34.4
```

We could get R1 to prefer the reverse path through BVI1 and R3 by changing the unicast route metrics, but a multicast-specific solution would be less disruptive. We prefer to use a static mroute command to override the unicast routing table for RPF checks. Below you see the required command and the resulting change in the **mtrace**.

```
R1(config)#ip mroute 172.16.34.4 255.255.255.255 172.16.10.5
R1
R1#clear ip mroute *
R1#

R1#mtrace 172.16.34.4
Type escape sequence to abort.
Mtrace from 172.16.34.4 to 172.16.123.1 via RPF
From source (?) to destination (?)
Querying full reverse path...
 0  172.16.123.1
-1  172.16.123.1 PIM/Static  [172.16.34.4/32]
-2  172.16.10.5 PIM/Static  [172.16.34.4/32]
-3  172.16.35.3 PIM  [172.16.34.0/25]
-4  172.16.34.4
```

In the static mroute command, we told R1 to use 172.16.10.5 as the RPF neighbor for traffic from 172.16.34.4. The **mtrace** command confirms it. So how come we are still not getting responses from R2? The output of **show IP mroute** on R1 shows the proper RPF path, but S0/0.123 is still not on the OILIST!

```
(*, 232.2.2.2), 00:06:35/stopped, RP 172.16.200.13, flags: SJCL
  Incoming interface: BVI1, RPF nbr 172.16.10.5
  Outgoing interface list:
    Loopback100, Forward/Sparse-Dense, 00:06:35/00:02:44
    Serial0/0.123, Forward/Sparse, 00:06:03/00:03:20

(172.16.34.4, 232.2.2.2), 00:06:35/00:02:58, flags: LJT
  Incoming interface: BVI1, RPF nbr 172.16.10.5, Mroute
  Outgoing interface list:
    Loopback100, Forward/Sparse-Dense, 00:06:35/00:02:44
    X                                                   X
```

In PIM Sparse Mode, interfaces are added to the Outgoing Interface List only when a client is detected downstream, that is, by the reception on that interface of a (*,G) join message. We need to go downstream, to R2, to further troubleshoot this problem. The changes we have made so far have not completely fixed the problem, but they were necessary steps. While our ping is still going on R4, we issue the command show IP mroute on R2, as shown on the next page.

The (*,G) entry on R2 shows that the shared tree to the RP was built through R1. The (S,G) entry tells us what happened at SPT cutover, when R2 pruned off the shared tree and tried to build a source tree. R2 sent a prune to R1 and sent a (S,G) join to R3 (172.16.123.3), which is on the unicast shortest path back to the source of the multicast stream, 172.16.34.4.

```
(*, 232.2.2.2), 00:34:00/stopped, RP 172.16.200.13, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 172.16.123.1
  Outgoing interface list:
    Loopback1, Forward/Sparse, 00:34:00/00:02:15
```

```
(172.16.34.4, 232.2.2.2), 00:06:23/00:00:13, flags: LJT
  Incoming interface: Serial0/0, RPF nbr 172.16.123.3
  Outgoing interface list:
    Loopback1, Forward/Sparse, 00:06:23/00:02:15
```

The problem, again, is that this source-based tree requires the multicast stream to go into and back out of the R1 hub multipoint interface. You may find that R2 responds momentarily, but then ceases to respond when the shared tree is pruned. In order to get consistent ping responses from R2, we need it to build both its shared tree and its source tree through R1, not R3. Again, we can accomplish this with a static mroute entry on R2: **ip mroute 172.16.34.4 255.255.255.255 172.16.123.1.** Here is what the resulting mroute entry looks like:

```
(*, 232.2.2.2), 00:03:30/stopped, RP 172.16.200.13, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 172.16.123.1
  Outgoing interface list:
    Loopback1, Forward/Sparse, 00:03:30/00:02:52

(172.16.34.4, 232.2.2.2), 00:03:28/00:02:59, flags: LJT
  Incoming interface: Serial0/0, RPF nbr 172.16.123.1, Mroute
  Outgoing interface list:
    Loopback1, Forward/Sparse, 00:03:28/00:02:52


R4#ping 232.2.2.2

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 232.2.2.2, timeout is 2 seconds:

Reply to request 0 from 172.16.34.3, 136 ms
Reply to request 0 from 172.16.123.2, 284 ms
Reply to request 0 from 172.16.10.6, 216 ms
Reply to request 0 from 172.16.10.1, 188 ms
Reply to request 0 from 172.16.35.5, 176 ms
R4#
```

To obtain a comprehensive view of the configuration tasks in this section, access the SHOWiT engine. With the SHOWiT engine, you can enter in over 1000 IOS commands as well a collection of NMC proprietary commands such as "show all".