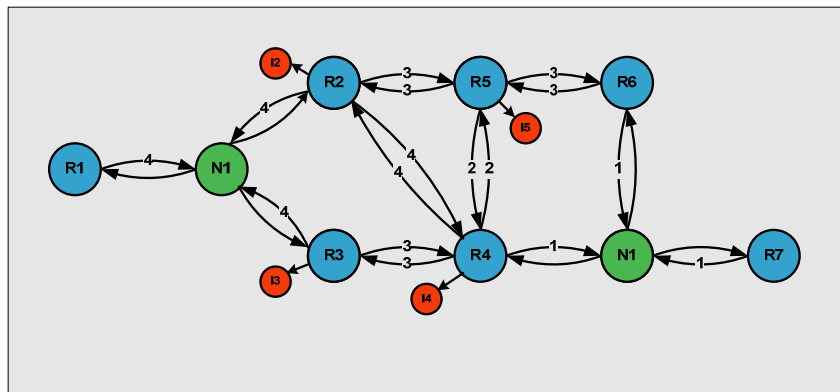


NETMASTERCLASS
ROUTING AND SWITCHING CCIE® TRACK

DOIT-200v6

VOLUME II



Scenario 8 ANSWER KEY

FOR

CCIE® CANDIDATES

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.

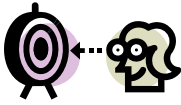
DOiT-200v6 Scenario 8: Spot the Issue Answer Key

Table of Contents

8.1 Frame Relay	6
8.2 Catalyst configuration	10
8.3 Frame Relay QoS	13
8.4 OSPF	19
8.5 RIP	23
8.6 EIGRP	25
8.7 BGP.....	26
8.8 VPN.....	29
8.9 Router Maintenance.....	30
8.10 IP Features	32
8.11 IPv6 IGP.....	32
8.12 IPv6 BGP	34
8.13 QOS	37
8.14 IOS Specialties	38
8.15 Catalyst specialties	39
8.16 Address Administration	40
8.17 Multicast.....	41



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 displayed in the Scenario diagram belong to network 172.16.0.0/16.
- Do not rely on dynamic Frame Relay inverse ARP.
- Do not use any static routes.
- Do not use “ip default-network”.
- Make sure all IPv4 and IPv6 loopback interfaces are advertised with their original mask, unless noted otherwise.
- Make sure all IPv4 interfaces in the diagram are **reachable** within this internetwork. **DO NOT FORGET THIS!**
- IP subnet 10.1.1.0/24 is excluded from the previous requirement.
- 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 in the Scenario diagram belong to network 172.16.0.0/16

The third and fourth octets of the IP addresses displayed on the diagram belong to 172.16.0.0/16.

Do not rely on dynamic Frame Relay Inverse ARP.

This requirement forces you to fulfill your Frame Relay inverse ARP requirements with Frame Relay map statements. Think of a Frame Relay map statement as the equivalent of a static inverse ARP entry.

Do not use any static routes.

Static routes can be used to solve a range of reachability problems. However, you cannot use them. You must rely on skillful configuration of all your unicast routing protocols.

Make sure all IPv4 and IPv6 loopback interfaces are advertised with their original mask, unless noted otherwise.

This requirement is primarily for the OSPF advertised loopbacks. Use “ip ospf network point-to-point” under the loopback interface. Otherwise, the loopback will be advertised as a /32 host entry by default.

Make sure all IP interfaces in the diagram are *reachable* within this internetwork. DO NOT FORGET THIS!

This is a key goal to observe. This requires that all of your IGP's are configured properly. Also, all of your routing policy tasks must be configured properly. The key elements of your routing policy include route redistribution and the controlling of routing updates using distribute-lists, route-maps and the distance command. A key point to remember about this exam is: 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 without the use of static routes.

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 INFORMATION 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

8.1 Frame Relay



HIDDEN ISSUES TO SPOT WITH FRAME RELAY

Issue: *Configure logical interfaces on the Frame Relay link between R3 and R5. This link does not have an explicit IP subnet.*

Solution:

Configure both sides of the Frame Relay the link as a point-to-point subinterfaces and apply IP unnumbered to the point-to-point subinterface. IP unnumbered can only be applied to point-to-point interfaces.

Implementation:

R3:

```
interface Loopback0
 ip address 172.16.103.1 255.255.255.0

interface Serial0/1.35 point-to-point
 ip unnumbered Loopback0
 frame-relay interface-dlci 503
```

R5:

```
interface Loopback0
 ip address 172.16.105.1 255.255.255.0

interface Serial1/1.35 point-to-point
 ip unnumbered Loopback0
 frame-relay interface-dlci 503
```

Issue: *None of the devices on the Frame Relay link between R3 and R5 should perform Frame Relay switching.*

Solution:

You have a back to back Frame Relay connection with no Frame Relay switch in between the two routers AND you cannot configure one router as a Frame Relay switch. One way to solve this requirement that would not violate any of the specified constraints is to disable the interface keepalive with the following interface configuration command: “no keepalive” on the main interface. When you enter this command on the Frame Relay interface, it disables LMI for that interface. With no LMI requirements, the Frame Relay interface set with keepalives disabled attains an up/up state when administratively enabled. To complete the configuration you must add command Frame Relay interface DLCI but you do not need any Frame Relay switching commands.

Implementation:

R3:

```
interface Serial0/1
no ip address
encapsulation frame-relay
no keepalive
no frame-relay inverse-arp
```

R5:

```
interface Serial1/1
no ip address
encapsulation frame-relay
no keepalive
clockrate 72000
no frame-relay inverse-arp
```

Verification:

To verify that you have correct interface to DLCI mapping issue the **show frame-relay map** command:

```
R5#sh frame map
Serial1/1.35 (up): point-to-point dlci, dlci 503(0x1F7,0x7C70), broadcast
R5#
```

To verify that you have configure PVC and it is in use issue the **show frame pvc** command:

```
R5#sh frame pvc

PVC Statistics for interface Serial1/1 (Frame-Relay DTE)

          Active      Inactive      Deleted      Static
Local            0             0             0             1
Switched         0             0             0             0
Unused           0             0             0             0

DLCI = 503, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial1/1.35

input pkts 61250          output pkts 41453          in bytes 30818585
out bytes 29994833        dropped pkts 0             in pkts dropped 0
out pkts dropped 0          out bytes dropped 0
in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
out BECN pkts 0          in DE pkts 0             out DE pkts 0
out bcast pkts 36084      out bcast bytes 29706486
5 minute input rate 2000 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 1d20h, last time pvc status changed 1d20h
R5#
```

After Frame Relay maps are created you can verify connectivity on link basis using **ping** command. All IP addresses connected to a segment shall be mutually reachable. There are few exceptions you need to be aware of:

- On multipoint Frame Relay interfaces (and also ATM and BRI interfaces) a router would not be able to ping his own IPv4 address unless a 'self pointing map' has been created
- For unnumbered interfaces you generally need routing assistance to reach the IP address on the other side of the unnumbered link. PPP peer route injection installs a host rout pointing to the

other side as a connected route; for non-PPP links you may need to wait until after a routing protocol is configured over the link to verify connectivity. You can also temporarily create a static route to verify connectivity before the routing protocol is configured, but ensure that the static route is removed immediately after the test is completed.

- For partial mesh segments which will have OSPF point-to-multipoint network type there is no need to configure maps on spokes pointing to other spokes as OSPF will fix cross-spoke connectivity. However this will result in spokes not able to ping each other until after OSPF is configured. You can temporarily add the spoke-to-spoke maps to test connectivity on a link basis without OSPF configuration, but ensure that the test maps are removed immediately after the test is completed.

To verify that you can ping across the link, issue the **ping 172.16.103.1** command:

```
R5#ping 172.16.103.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.103.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
R5#
```

Issue: Configure logical interfaces on the Frame Relay link between R3 and R4. Use CHAP authentication with the password “nmc”.

Solution:

A Frame Relay logical interface that uses CHAP authentication is not a point-to-point or multipoint subinterface. It is a virtual-template interface. In order to have a Frame Relay connection that uses CHAP authentication, you must have a Frame Relay connection that uses PPP. Regardless of the Frame Relay interface type – physical, point-to-point subinterface, multipoint subinterface, enter the following command: “frame relay interface-dlci 304 ppp Virtual-Template1”. This command applies PPP encapsulation to a specific Frame Relay DLCI. To complete the configuration, you must create and configure a virtual-template interface. It is on the virtual-template interface that you would apply the IP address as well as the command “ppp authentication chap”.

Implementation:

R3:

```
username R4 password 0 nmc

interface Serial0/0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay interface-dlci 304 ppp Virtual-Template1
no frame-relay inverse-arp
frame-relay lmi-type ansi

interface Virtual-Template1
ip address 172.16.34.3 255.255.255.0
no peer neighbor-route
ppp authentication chap
```


R4:

```
username R3 password 0 nmc

interface Serial0/0
 ip address 172.16.124.4 255.255.255.0
 encapsulation frame-relay
 no fair-queue
 frame-relay interface-dlci 403 ppp Virtual-Template1
 no frame-relay inverse-arp
 frame-relay lmi-type ansi

interface Virtual-Template1
 ip address 172.16.34.4 255.255.255.0
 no peer neighbor-route
 ppp authentication chap
```

Verification:

Verify that corresponding Virtual-Access interfaces is up, and that you can ping across the link:

```
R4#sho ip int brief
Interface                IP-Address      OK? Method Status      Protocol
FastEthernet0/0          172.16.46.4     YES NVRAM    up          up
Serial0/0                 172.16.124.4    YES NVRAM    up          up
FastEthernet0/1          unassigned      YES NVRAM    administratively down down
Serial0/1                 unassigned      YES NVRAM    administratively down down
Virtual-Access1          172.16.34.4     YES TFTP    up          up
Virtual-Template1        172.16.34.4     YES NVRAM    down       down
Virtual-Access2          unassigned      YES unset   down       down
Loopback104              172.16.104.1    YES NVRAM    up          up
```

```
R4#ping 172.16.34.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
```



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".

8.2 Catalyst configuration



HIDDEN ISSUES TO SPOT WITH CATALYST CONFIGURATION

Issue: *Configure VLANs according to the diagram and the VLAN configuration table.*

Solution:

Carefully examine the diagram. In particular, examine the two links between CAT1 and CAT2 at the bottom of the diagram. These two links correspond to ports Fa0/13 and Fa0/14 on CAT1 and CAT2. It is these two links that interconnect the two Catalysts involved in this scenario. Notice that only the top link has a VLAN associated with it. It is VLAN 40. This top link is associated with the Fa0/13 port on both CAT1 and CAT2. The link below VLAN 40 (port Fa0/14) has no VLAN associated with it. It only has the following IP addresses assigned to it: 172.16.2.10/24 on CAT1 and 172.16.1.20/24 on CAT2. These two IP addresses are directly assigned to port Fa0/14 on their respective Catalyst. In order to assign an IP address directly to a Catalyst port, you must enter the following interface configuration command: “no switchport”. Please re-examine the IP addresses supplied above. Then closely examine the IP addresses assigned to VLAN 40 on your diagram. You will notice that two IP subnets are involved: 172.16.10.0/24 and 172.16.20.0/24. When examining the IP addresses, they seem to be assigned incorrectly in a “criss-cross” manner. Even though this IP address scheme seems incorrect, you can get it to function by adding two static ARP entries on both CAT1 and CAT2. The static arp entries on CAT1 would reference the following two CAT2 assigned addresses: 172.16.2.20 and 172.16.1.20. The static arp entries on CAT2 would reference the following two CAT1 assigned addresses: 172.16.2.10 and 172.16.1.10. If you do not resolve the IP addressing issues between the two interconnecting links between CAT1 and CAT2, the RIP configuration in Section 8.5 will not work.

Implementation:

CAT1:

Configure Fa0/13 for trunking:

```
interface FastEthernet0/13
  description Trunk to CAT-2
  switchport trunk encapsulation isl
  switchport trunk allowed vlan 10,20,30,40
  switchport mode trunk
  speed 100
```

Configure Fa0/14 with no switchport as a router port:

```
interface FastEthernet0/14
  description L3 to CAT-2
  no switchport
  ip address 172.16.2.10 255.255.255.0

interface Vlan40
  ip address 172.16.1.10 255.255.255.0
```

Modify ARP table to provide reachability to both remote IP addresses:

```
arp 172.16.1.20 000a.b7f7.0700 ARPA
arp 172.16.2.20 000a.b7f7.0700 ARPA
```

Perform the same operations on CAT2:

CAT2:

```
interface FastEthernet0/13
  description Trunk to CAT-1
  switchport trunk encapsulation isl
  switchport trunk allowed vlan 10,20,30,40
  switchport mode trunk
  speed 100

interface FastEthernet0/14
  description L3 to CAT-1
  no switchport
  ip address 172.16.1.20 255.255.255.0

interface Vlan40
  ip address 172.16.2.20 255.255.255.0
  arp 172.16.2.10 0009.e8f9.4380 ARPA
  arp 172.16.1.10 0009.e8f9.4380 ARPA
```

MAC addresses in the arp command will be different on different switches.

Verification:

Make sure you can ping across the interconnect and both IP addresses are reachable:

```
CAT1#ping 172.16.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

CAT1#ping 172.16.2.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

CAT2#ping 172.16.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

CAT2#ping 172.16.2.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Issue: Make sure you allow only VLANs configured in this Scenario to pass through the trunks.

Solution:

First, identify what VLANs are REQUIRED to pass through any of the trunks in this Scenario. Once you have determined this, limit the VLANs that pass over a given trunk to only those that you have manually identified. You can limit what VLANs are transmitted on a trunk with the following Catalyst 3550 port command: “switchport trunk allowed vlan XX” where XX is a VLAN number. With this command, you can specify a range of VLANs with a hyphen and list individual VLANs with a comma. When you use this command, make sure that there are no spaces in the list of VLANs you supply, regardless of whether you use commas or hyphens to delimit the specific VLANs listed. Here is a sample of this command this is found in the Scenario 8 final configuration: “switchport trunk allowed vlan 10,12,16,20,30,40”.

Implementation:

```
interface FastEthernet0/13
description Trunk to CAT-2
switchport trunk encapsulation isl
switchport trunk allowed vlan 10,12,16,20,30,40
switchport mode trunk
```

Verification:

To verify that only necessary VLANs are allowed on the trunk, issue the **show interface trunk** command:

```
CAT1#show interface trunk

Port      Mode      Encapsulation  Status      Native vlan
Fa0/2     on        isl             trunking    1
Fa0/8     on        isl             trunking    1
Fa0/13    on        isl             trunking    1
Fa0/24    desirable n-isl          trunking    1

Port      Vlans allowed on trunk
Fa0/2     10,12
Fa0/8     16,30
Fa0/13    10,12,16,20,30,40
Fa0/24    1-4094

Port      Vlans allowed and active in management domain
Fa0/2     10,12
Fa0/8     16,30
Fa0/13    10,12,16,20,30,40
Fa0/24    1,10,12,16,20,30,40

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/2     10,12
Fa0/8     16,30
Fa0/13    10,12,16,20,30,40

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/24    1,10,12,16,20,30,40
```



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”.

8.3 Frame Relay QoS



HIDDEN ISSUES TO SPOT WITH FRAME RELAY QoS

Issue: On router R4 guarantee that future IPv6 transit traffic generated by IPv6 BGP task through router R4 will have guaranteed 25% of overall bandwidth outbound for PVCs that it is going to use. Apply only on necessary PVCs.

Solution:

You will need to read forward IPv6 BGP section to understand what the traffic will be. It is an IPv6-in-IP tunnel, which has IP protocol number 41. Since this protocol is not supported by class-map match command, an extended access-list will need to be created to match it.

Frame Relay can perform QoS on per-PVC basis, which needs to be enabled by using “Frame Relay traffic-shaping” command.

Create an access list to match ipv6inip traffic:

```
ip access-list extended IPv6inIP
 permit 41 any any
```

Create a class-map to match this pattern:

```
class-map match-all Match-IPv6inIP
 match access-group name IPv6inIP
```

Create a policy map to operate on this class map:

```
policy-map Guarantee-IPv6inIP
 class Match-IPv6inIP
 bandwidth percent 25
```

Create a Frame Relay map-class to attach policy map to it:

```
map-class frame-relay Forward
 frame-relay cir 115200
 service-policy output Guarantee-IPv6inIP
```

Attach map-class to necessary PVCs:

```
interface Serial0/0
 ip address 172.16.124.4 255.255.255.0
 ip pim nbma-mode
 ip pim sparse-dense-mode
 ip multicast boundary Multicast-filter
 encapsulation frame-relay
 ip ospf network broadcast
```

```

no fair-queue
frame-relay traffic-shaping
frame-relay map ip 172.16.124.1 401 broadcast
frame-relay map ip 172.16.124.2 402 broadcast
frame-relay map ip 172.16.124.4 401
frame-relay interface-dlci 401
  class Forward
frame-relay interface-dlci 402
  class Forward
frame-relay interface-dlci 403 ppp Virtual-Template1
no frame-relay inverse-arp
frame-relay lmi-type ansi
max-reserved-bandwidth 100

```

Verification:

You may verify this configuration on per PVC basis:

```
R4#sh frame pvc 401
```

```
PVC Statistics for interface Serial0/0 (Frame-Relay DTE)
```

```
DLCI = 401, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0
```

```

input pkts 54904          output pkts 52107          in bytes 4032769
out bytes 4149056        dropped pkts 0             in pkts dropped 0
out pkts dropped 0      out bytes dropped 0
in FECN pkts 0          in BECN pkts 0           out FECN pkts 0
out BECN pkts 0         in DE pkts 0             out DE pkts 0
out bcast pkts 45818    out bcast bytes 3605472
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 3d08h, last time pvc status changed 3d04h
cir 115200  bc 115200  be 0          byte limit 1800  interval 125
mincir 57600  byte increment 1800 Adaptive Shaping none
pkts 15056   bytes 1209222  pkts delayed 0   bytes delayed 0
shaping inactive
traffic shaping drops 0
service policy Guarantee-IPv6inIP
Serial0/0: DLCI 401 -

```

```
Service-policy output: Guarantee-IPv6inIP
```

```
Class-map: Match-IPv6inIP (match-all)
```

```
2607 packets, 243745 bytes
```

```
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: access-group name IPv6inIP
```

```
Queueing
```

```
Output Queue: Conversation 25
```

```
Bandwidth 25 (%)
```

```
Bandwidth 14 (kbps) Max Threshold 64 (packets)
```

```
(pkts matched/bytes matched) 0/0
```

```
(depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: class-default (match-any)
```

```
12348 packets, 957546 bytes
```

```
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: any
```

```
Output queue size 0/max total 600/drops 0
```

```
R4#
```



HIDDEN ISSUES TO SPOT WITH ROUTE REDISTRIBUTION

In this scenario, all protocols are in an equal position. Since there is no core and edge relationship, redistribution must be done based on what the requirements for reachability exist in the preamble of this Scenario. There is only two specific requirements for reachability: one asks that traffic destined to 10.10.10.0/24 is forwarded to R5 from R2, and another states that prefix 10.10.10.0/24 is in the routing tables of R1 and R4 with the next hop pointing to R2.

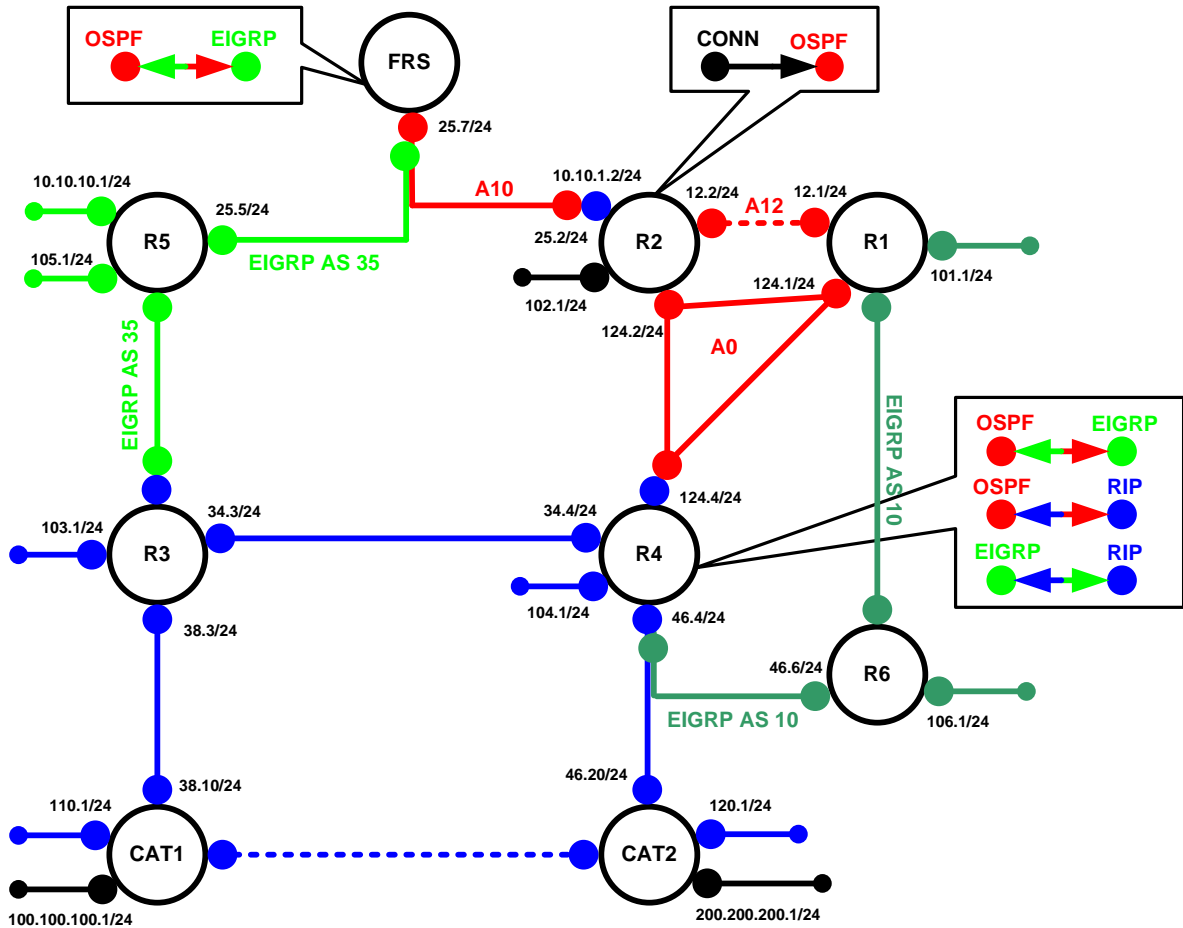
As the following redistribution diagram reflects, redistribution points are R4 and FRS.

On router FRS, OSPF and EIGRP are fully mutually redistributed.

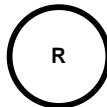




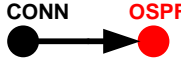



On router R4, OSPF, RIP and EIGRP are fully mutually redistributed.

You also want to protect intra-RIP routing by assigning administrative distance of less than other IGPs, however it is not necessary due to the nature of this exercise. It is, however, an option to check. For example, if EIGRP 35 was an OSPF domain instead, this precaution would have been necessary.

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.



Legend

	Router		Loopback
	RIP		Mutual redistribution, eg. EIGRP and OSPF
	EIGRP		One way redistribution, eg. CONNECTED into OSPF
	OSPF		Prefix injection
			Trash can

See the scenario master diagram and VLAN table for data link details!

Additional discussions of redistribution are provided under specific routing protocol Spot the Issues sections provided in this Answer Key.

One way to test that your redistribution satisfies the goal of universal connectivity is to run a TCL script like the one below on each router. TCL scripting support is available in the IOS versions used here on routers R1, R2, R5 and R6 (the 3600 models). The simple script below lists all of the IP addresses in our pod. It can be built once in notepad, and then pasted into each router to automate pings. There is a paper on TCL scripting available in the READiT section of the Netmasterclass website. Some addresses are used in later tasks and may not be reachable at this point. Run tclsh in privileged mode, paste the script below, and then issue the command tclq.

```
foreach address {  
10.10.10.1  
10.10.11.2  
10.10.11.3  
10.10.1.2  
10.10.2.3  
172.16.101.1  
172.16.102.1  
172.16.103.1  
172.16.104.1  
172.16.105.1  
172.16.106.1  
172.16.1.10  
172.16.110.1  
172.16.1.20  
172.16.120.1  
172.16.12.1  
172.16.12.2  
172.16.124.1  
172.16.124.2  
172.16.124.4  
172.16.2.10  
172.16.2.20  
172.16.25.2  
172.16.25.5  
172.16.25.7  
172.16.34.3  
172.16.34.4  
172.16.38.10  
172.16.38.3  
172.16.46.20  
172.16.46.4  
172.16.46.6  
} {ping $address}
```

We also need to make sure that our solution is a stable one. If we have split-horizon or other route feedback problems routes may continually be inserted and removed from our routing tables. We can test stability by observing the output of debug IP routing. Finally, we need to make sure that our routes are optimal: that native prefixes are routed by native protocols and that we are using the shortest paths. This requires close examination of each routing table.

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. Whenever a permit column for a given routing protocol is completely empty, it reflects that no prefixes were redistributed into the routing protocol. This represents that the routing protocol is involved in one-way redistribution.

Redist point	Into RIP		Into OSPF		Into EIGRP	
	PERMIT	DENY	PERMIT	DENY	PERMIT	DENY
R4	All routes from OSPF All routes from EIGRP 10		All routes from RIP All routes from EIGRP 10		All routes from RIP All routes from OSPF	
FRS			All routes from EIGRP 35		All routes from OSPF	

8.4 OSPF



HIDDEN ISSUES TO SPOT WITH OSPF

Issue: Configure Area 10 between R2 and FRS.

Solution:

By using the non-broadcast network type on the Ethernet segment VLAN 10, you are going to generate a single prefix for the VLAN subnet, 172.16.25.0/24. When you configure this connection as an OSPF non-broadcast network type, remember to configure OSPF neighbor statements. It does not matter which router is elected DR or BDR. This configuration step is setting the stage for an OSPF LSA Type 5 forwarding address problem. This will be discussed in the following section.

Implementation:

R2:

```
interface FastEthernet0/0
  description VLAN 10
  ip address 172.16.25.2 255.255.255.0
  ip ospf network non-broadcast

router ospf 1
  log-adjacency-changes
  network 172.16.25.0 0.0.0.255 area 10
  neighbor 172.16.25.7
```

FRS:

```
interface Ethernet0
  description VLAN 10
  ip address 172.16.25.7 255.255.255.0
  ip ospf network non-broadcast
  ip ospf priority 0

router ospf 1
  log-adjacency-changes
  network 172.16.25.0 0.0.0.255 area 10
```

Verification:

To verify the OSPF adjacency between R2 and FRS issue the **show ip ospf neighbor** command:

```
FRS#sh ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.102.1	1	FULL/DR	00:01:45	172.16.25.2	Ethernet0

```
FRS#
```

Issue: Make sure packets destined to network 10.10.10.0/24 are forwarded to R5 from R2.

Solution:

When you examine the Scenario diagram, you will see that the 10.10.10.0/24 prefix is originated by ISIS on router R5 and it is redistributed into OSPF on FRS. Both R5 and FRS reside on the common subnet of 172.16.25.0/24. When router R2, which also resides on the 172.16.25.0/24 subnet, receives packets that are destined to the 10.10.10.0/24 network, it is most efficient for R2 to forward packets directly to router R5 instead of to the OSPF ASBR, FRS. To forward packets destined to the 10.10.10.0/24 network to the OSPF ASBR (router FRS) instead of the actual source of this prefix (router R5) would result in forwarding packets via an unnecessary next-hop to FRS. OSPF External LSA's possess a feature to avoid such unnecessary next-hops. This OSPF feature is called the Type 5 LSA "forwarding address" feature. The OSPF Type 5 LSA "forwarding address" feature allows the original next-hop forwarding address to be preserved within the External LSA. This is useful on a multi-access network to avoid packets from taking unnecessary next-hops. In our Scenario, the forwarding address for the 10.10.10.0/24 subnet will be set to 172.16.25.5. With this forwarding address feature, router R2 will forward packets destined to the 10.10.10.0/24 subnets directly to R5, and consequently bypassing the ASBR FRS. It is possible to accidentally eliminate this useful forwarding address feature. The forwarding address can accidentally be set to a null address of 0.0.0.0. In this Scenario, this can occur if you configure FRS with an OSPF network command of "network 172.16.25.7 0.0.0.0 area 10". When you configure the OSPF network command for the subnet on the ASBR that is used in the forwarding address with a wildcard mask of 0.0.0.0, it results in setting the forwarding address for all Type 5 LSA's generated by the ASBR to 0.0.0.0. As an example, the subnet prefix used in the forwarding address of the Type 5 LSA for the 10.10.10.0/24 prefix is 172.16.25.0/24. If this prefix is configured under the OSPF process with the following network command: "network 172.16.25.5 0.0.0.0 area 10", this will result in setting the forwarding address to 0.0.0.0. To avoid this from happening, set the network statement for the prefix used in the forwarding address to a statement that uses something other than a 0.0.0.0 wildcard mask. To preserve the forwarding address in this Scenario, configure the following network command for the 172.16.25.0/24 subnet under the OSPF routing process: "network 172.16.25.0 0.0.0.255 area 10".

Implementation:

As it is also noted in the previous implementation, OSPF area 10 shall be configured using any mask, different than 0.0.0.0:

```
router ospf 1
  log-adjacency-changes
  redistribute connected subnets route-map Connected-->OSPF
  network 172.16.25.0 0.0.0.255 area 10
```

Verification:

Issue the **show ip route** command on R2 and verify the next hop for network 10.10.10.0/24:

```
R2#sh ip route | inc 10.10.10.0
O E2    10.10.10.0 [110/1] via 172.16.25.5, 1d11h, FastEthernet0/0.10
```

Issue: Make sure prefix 10.10.10.0/24 is in the routing table of R1 and R4 with the next-hop pointing to R2.

Solution:

The 10.10.10.0/24 prefix is originated by ISIS on router R5. It can be learned by routers R1 and R4 via multiple sources. Consider the following 3 sources: (1) The 10.10.10.0/24 prefix can be learned by routers R1 and R4 via R2; (2) it can be learned via R4 if R4 receives routes from RIP and EIGRP and (3) it can be learned from R1 via EIGRP. Of these three possible options, both R1 and R4 will select R2 due to OSPF's lower administrative distance (Administrative Distance 110) when compared to RIP (Administrative Distance 120) and external EIGRP (Administrative Distance 170). This assures that R2 is the preferred source for the 10.10.10.0/24 prefix for routers R1 and R4. However, we must go one step further. We must also assure that both R1 and R4 select R2 as the "next-hop" for the 10.10.10.0/24 prefix. To assure that R1 and R4 select R2 as the "next-hop" for the 10.10.10.0/24 prefix, you must also configure either the OSPF network-type "broadcast" or "non-broadcast" for the 172.16.124.0/24 subnet. These two network types will retain R2 as the next-hop on the 172.16.124.0/24 segment. By doing so, these two OSPF network types make the subnet appear as a multi-access network. See Appendix D at the end of this workbook for more details on the behavior of OSPF network types. This Appendix provides a detailed explanation of the next-hop behavior of the different OSPF network types.

Implementation:

Configure OSPF network type Broadcast on the subnet 124.0/24:

R1:

```
interface Serial0/0
 ip address 172.16.124.1 255.255.255.0
 encapsulation frame-relay
 ip ospf network broadcast
 ip ospf priority 0
 no fair-queue
 frame-relay map ip 172.16.124.1 104
 frame-relay map ip 172.16.124.2 104
 frame-relay map ip 172.16.124.4 104 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

R2:

```
interface Serial0/0
 ip address 172.16.124.2 255.255.255.0
 encapsulation frame-relay
 ip ospf network broadcast
 ip ospf priority 0
 no fair-queue
 frame-relay map ip 172.16.124.1 204
 frame-relay map ip 172.16.124.2 204
 frame-relay map ip 172.16.124.4 204 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

R4:

```
interface Serial0/0
 ip address 172.16.124.4 255.255.255.0
 encapsulation frame-relay
 ip ospf network broadcast
 no fair-queue
 frame-relay map ip 172.16.124.1 401 broadcast
 frame-relay map ip 172.16.124.2 402 broadcast
 frame-relay map ip 172.16.124.4 401
 frame-relay interface-dlci 403 ppp Virtual-Template1
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

Verification:

To make sure that network type is BROADCAST on the subnet 124.0/24, issue the **show ip ospf interface X** command, where X is the name of the interface:

```
R1#sh ip ospf inte s0/0 | inc Type
 Process ID 1, Router ID 172.16.101.1, Network Type BROADCAST, Cost: 64
```

To verify that OSPF adjacencies are established issue the **show ip ospf neighbor** command:

```
R4#sh ip ospf nei
Neighbor ID      Pri   State           Dead Time   Address        Interface
172.16.102.1     0    FULL/DROTHER    00:00:34   172.16.124.2   Serial0/0
172.16.101.1     0    FULL/DROTHER    00:00:33   172.16.124.1   Serial0/0
```

To verify that the next hop for 10.10.10.0/24 is set correctly to R2, issue the **show ip route ospf | inc 10.10.10** command:

```
R4#sh ip route ospf | inc 10.10.10
O E2   10.10.10.0 [110/1] via 172.16.124.2, 3d17h, Serial0/0

R1#sh ip route ospf | inc 10.10.10
O E2   10.10.10.0 [110/1] via 172.16.124.2, 3d17h, Serial0/0
```

To make sure that the next hop is reachable, issue the **ping 172.16.124.2** command on R1 and R4:

```
R1#ping 172.16.124.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/63/72 ms

R4#ping 172.16.124.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```



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".

8.5 RIP



HIDDEN ISSUES TO SPOT WITH RIP

Issue: Do not use secondary addresses in the RIP domain

Solution:

This task is related to the mismatching IP addresses associated with the 172.16.10.0/24 and 172.16.20.0/24 subnets interconnecting CAT1 and CAT2. See the Spot the Issues Section for the Catalyst Configuration in Task 8.2 above. In order to successfully exchange RIP updates between CAT1 and CAT2, enter the following command under the “router rip” configuration mode: “no validate update-source”.

Implementation:

Configure RIP to not to validate the update source:

```
router rip
no validate-update-source
```

Verification:

Verify that RIP receives and accepts updates over the interconnect:

```
CAT1#sh ip route rip
172.16.0.0/24 is subnetted, 16 subnets
R    172.16.46.0 [120/1] via 172.16.2.20, 00:00:21
    [120/1] via 172.16.1.20, 00:00:21
R    172.16.34.0 [120/1] via 172.16.38.3, 00:00:04, Vlan20
R    172.16.25.0 [120/1] via 172.16.38.3, 00:00:04, Vlan20
R    172.16.12.0 [120/2] via 172.16.2.20, 00:00:21
    [120/2] via 172.16.1.20, 00:00:21
R    172.16.124.0 [120/2] via 172.16.2.20, 00:00:21
[skipped]
CAT1#
```

Issue: CAT1 and CAT2 should forward traffic directed to 172.16.103.1 via R4. R4 should forward traffic directed to 172.16.103.1 across subnet 172.16.34.0/24.

Solution:

Both of these requirements can be fulfilled by configuring an inbound “offset-list” command under the RIP routing process on CAT1. With the offset-list command, you can manually increase the metric for one or all updates received. Without the offset-list command, the RIP process on both CAT1 and CAT2 would pick router R3 as the next-hop for the 172.16.103.0/24 prefix instead of R4. R3 shares a common subnet with CAT1. By setting the offset-list command, we can make R3 appear to be several hops away from CAT1.

Implementation:

```
router rip
  offset-list RIP-offset-VLAN20 in 3 Vlan20

ip access-list standard RIP-offset-VLAN20
  permit 172.16.103.0
```

Verification:

Verify that CAT1 will send traffic to 103.1 via R4:

```
CAT1#sh ip route | inc 103.0
R      172.16.103.0 [120/3] via 172.16.2.20, 00:00:10
CAT1#
```

Verify that R4 will send traffic to 103.1 over subnet 34.0/24:

```
R4#sh ip route | inc 103.0
R      172.16.103.0 [80/1] via 172.16.34.3, 00:00:13, Virtual-Access1
R4#
```

Issue: R4 will not advertise the EIGRP redistributed prefixes to CAT2.

Solution:

The R4's FastEthernet interface on the VLAN30 belongs to two routing protocols RIP and EIGRP. The router R4 performs the mutual redistribution between RIP and EIGRP, please look at the redistribution diagram. R4 will not advertise the prefixes learned from the FastEthernet back out the same interface, this rule is applied to the redistributed prefixes as well. In order to advertise such prefixes you need to disable the split horizon on the FastEthernet on R4 (Reminder: by default it is enabled on the Ethernet interfaces) Please look at the example given below:

```
R4# show ip rip data
...
172.16.106.0/24 redistributed
  [2] via 172.16.46.6,

CAT2# show ip rip data
...
172.16.106.0/24
  [2] via 172.16.46.4, 00:00:26, Vlan30
```

Implementation:

```
interface FastEthernet0/0
  description VLAN 30
  ip address 172.16.46.4 255.255.255.0
  no ip split-horizon
```



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".

8.6 EIGRP



HIDDEN ISSUES TO SPOT WITH EIGRP

This scenario contains two separated EIGRP domains, AS 10 and AS 35.

Issue: *Configure EIGRP AS 35 between FRS and R5 on VLAN10. R2 is not part of this AS. Extend this AS to the link between R5 and R3.*

Solution:

There are no IP addresses on the link between R3 and R5. On this link, "ip unnumbered" must be used with addresses of loopback interfaces. You will not be able to ping across the link until EIGRP has exchanged routing information.

Verification:

```
R3#sh ip route eigrp
  172.16.0.0/24 is subnetted, 17 subnets
D    172.16.25.0 [90/2172416] via 172.16.105.1, 2d07h, Serial0/1.35
D    172.16.105.0 [90/2297856] via 172.16.105.1, 2d07h, Serial0/1.35
  10.0.0.0/24 is subnetted, 4 subnets
D    10.10.10.0 [90/2297856] via 172.16.105.1, 2d00h, Serial0/1.35

R3#ping 172.16.105.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.105.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

Issue: *Do not allow EIGRP routing updates to go beyond necessary devices on VLAN10.*

Solution:

Configure EIGRP neighbor statement pointing to FastEthernet interfaces on FRS and R5. EIGRP will not send multicast on these interfaces since then:

```
router eigrp 35
 network 10.10.10.0 0.0.0.255
 network 172.16.25.0 0.0.0.255
 network 172.16.105.0 0.0.0.255
 no auto-summary
 neighbor 172.16.25.7 FastEthernet0/0
```



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".

8.7 BGP



HIDDEN ISSUES TO SPOT WITH BGP

Issue: You must analyze the last three tasks of this BGP section together: (1) Task 8.8.7: R2 must not accept any NLRI transiting through R5. R5 must not accept any NLRI transiting through R2. Do not use any filtering technique. (2) Task 8.8.8: Make sure you provide the transit connectivity between 100.100.100.0/24 and 200.200.200.0/24 across AS 100 even if one of the IBGP peers is lost. (3) Task 8.8.9: Configure the minimum number of IBGP peer relationships to fulfill the criteria above. Do not introduce new AS numbers.

Solution:

This entire BGP section creates a complex route-reflector configuration requirement. The primary indication that it is a route-reflector task can be found in Task 8.8.9, “add the minimum number of IBGP peer relationships to fulfill the criteria”. This eliminates the configuration requirement of a full mesh in AS 100. Alternatives to a full mesh are route-reflectors and confederations. The recommended general practice is to attempt to use a route-reflector configuration instead of a confederation to fulfill a “no full mesh of IBGP speakers requirement” or a “minimum number IBGP peering sessions in a given AS requirement”. Route-reflectors are recommended over confederations to fulfill these requirements because they are simpler and faster to configure and troubleshoot. Even when you exclude the **“Do not introduce new AS numbers” requirement in Task 8.8.9**, you still cannot configure a confederation because of the filtering requirement specified in task 8.8.7. Therefore, to solve all of the configuration requirements of this BGP Section, make each and every router in AS 100 a route-reflector. Assign a unique cluster-id to each and every one of the AS 100 BGP speakers with the exception of two: routers R2 and R5. Make sure that these two IBGP speakers have the same BGP cluster-id. By configuring R2 and R5 in this manner, you will fulfill Task 8.8.7. Remember that if you configure a cluster-id on a router, enter the cluster-id before the route-reflector-client commands. If you enter the route-reflector-client commands before the bgp cluster-id, you will get an IOS error message immediately after you attempted to enter the bgp cluster-id.

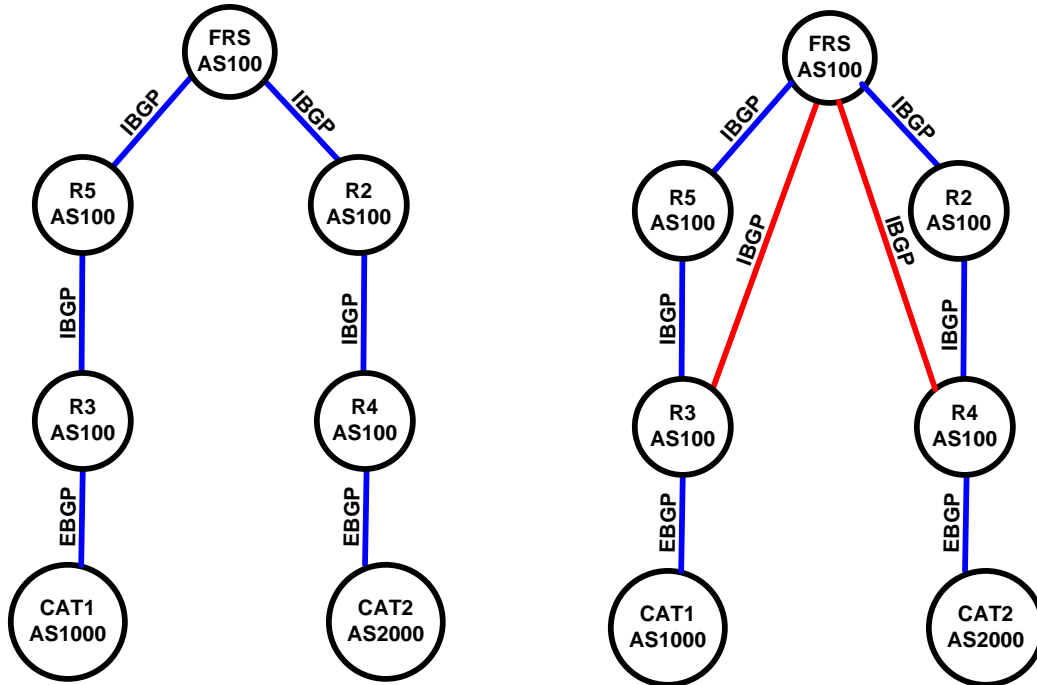
By configuring the same BGP cluster-id on routers R2 and R5, you will fulfill the requirements of Task 8.8.7. When the identical BGP cluster-id is assigned to two route-reflectors, they will not accept updates from each other. To illustrate this, examine the following “debug ip bgp updates” output:

```
BGP(0): 172.16.25.7 rcv UPDATE about 100.100.100.0/24 -- DENIED due to: CLUSTERLIST contains our own cluster ID;
BGP: 172.16.104.1 Route Reflector cluster loop; Received cluster-id 0.0.0.25
BGP(0): 172.16.104.1 rcv UPDATE w/ attr: nexthop 172.16.38.10, origin i, localpref 100, metric 0, originator 172.16.103.1, clusterlist 172.16.104.1 172.16.25.7 0.0.0.0, path 1000, community , extended community
BGP(0): 172.16.104.1 rcv UPDATE about 100.100.100.0/24 -- DENIED due to: CLUSTERLIST contains our own cluster ID;
```

This is the type of output generated when a route-reflector receives a BGP update that is referencing the same cluster-id as its own.

An alternative configuration to consider is to fulfill the configuration requirements of this Section is to make both routers R2 and R5 basic IBGP speakers with no router-reflector configuration. If both R2 and R5

were configured as non-route-reflector IBGP speakers, they will never advertise IBGP routes. The Scenario topology would be constructed in the manner reflected in the following diagram:



Instead of having only the IBGP neighbor relationships displayed in the diagram on the left, two more IBGP neighbor relationships are added between R3 and FRS as well as R4 and FRS (See the diagram on the right.) With these additional neighbor relationships, updates from AS 1000 can get to AS 2000 and updates from AS 2000 can get to AS 1000. Therefore, the “transit” connectivity stated in the configuration tasks is fulfilled. If R2 and R5 are configured as non-route-reflector IBGP speakers, NOTHING will transit through either R2 or R5. Initially, this seems to fulfill the configuration requirements of this task. However, even though this configuration alternative seems like a solution, it violates one of the configuration requirements of this Section: “each BGP speaker in AS 100 must possess both prefixes (one from AS 1000 and the second from AS 2000) **even if one of the given IBGP peers is lost.**”

In order to fulfill this requirement, both R2 and R5 must be route-reflectors AND the two additional IBGP neighbor relationships displayed on the righthand side of the diagram above must be configured. You cannot fulfill this requirement with either R2 or R5 configured as simply non-route-reflector IBGP speakers.

A common point of confusion with this scenario is reconciling the language of Task 8.8.4 with the language of Task 8.8.9. Task 8.8.4 instructs you to restrict your peering relationships within AS 100 to a specific list. Task 8.8.9 directs you to use the minimum number of IBGP peers to fulfill the configuration requirements. Think of Task 8.8.4 as providing you with the initial number of IBGP peers to act as a “starting point” for completing the overall BGP configuration task in this Scenario. Task 8.8.9 is instructing you to add any additional IBGP neighbor relationships to fulfill the requirements of Task 8.8.8. In order to fulfill Task 8.8.8 and 8.8.9, you need to configure the two additional IBGP neighbor relationships displayed on the righthand side of the diagram above (R3-FRS and R4-FRS).

To test to make sure that the prefix from AS 1000 as well as the prefix from AS 2000 are in the BGP table of all IBGP speakers even when one of the IBGP neighbor relationships fails within AS 100, perform an administrative shutdown on one of the IBGP neighbor relationships. After doing this, you should still see the two prefixes in all IBGP tables. As an example, if the IBGP neighbor relationship between R3 and R5 is shutdown, the following BGP updates will occur:

For the 100.100.100.0/24 prefix, the following updates will be propagated:

R3 will send an update to FRS
FRS will send an update to R5
FRS will send an update to R2
FRS will send an update to R4

For the 200.200.200.0/24 prefix, the following updates will be propagated:

R4 will send an update to FRS and R2
R2 will send an update to FRS
FRS will send an update to R5 and R3

To further increase your understanding of this BGP Scenario, consider other breaks in IBGP neighbor relationships within AS 100. Determine how the non-AS 100 updates get propagated to all IBGP speakers even when there is a lost neighbor relationship within AS 100.

A final note to remember is: This Scenario is based upon the loss of an IBGP neighbor relationship within AS 100 and NOT the loss of an actual Data-Link connection. If a Data-Link was lost in this BGP Scenario, it would disrupt the forwarding of both BGP Control Plane messages as well as actual IP traffic. Again, this BGP Scenario is based upon the loss of an IBGP neighbor relationship within AS 100 and not an underlying Data-Link connection.

Implementation:

Configure R2 and R5 with the same cluster id:

```
router bgp 100
  bgp cluster-id 25
```

Configure R3 and R4 as a route-reflector client of FRS for IBGP redundancy:

```
router bgp 100
  no synchronization
  bgp log-neighbor-changes
  neighbor 172.16.103.1 remote-as 100
  neighbor 172.16.103.1 route-reflector-client
  neighbor 172.16.104.1 remote-as 100
  neighbor 172.16.104.1 route-reflector-client
```



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".

8.8 VPN



HIDDEN ISSUES TO SPOT WITH VPN

Issue: Tasks 8.9.1 and 8.9.2: Configure IP address 10.1.X.0/24 in specific interface. Preserve the existing IP addressing.

Solution:

Configure the appropriate 10.1.X.0/24 address on the specified interface as ip secondary addresses.

Implementation:

R2:

```
interface FastEthernet0/0
description VLAN 10
ip address 10.10.1.2 255.255.255.0 secondary
ip address 172.16.25.2 255.255.255.0
```

R3:

```
interface FastEthernet0/0
description VLAN 20
ip address 10.10.2.3 255.255.255.0 secondary
ip address 172.16.38.3 255.255.255.0
```

Issue: Configure 10.10.11.0/24 between routers R2 and R3. Use RIP version 2 within VPN. Do not use generic encapsulation, use IP into IP encapsulation.

Solution:

If you are going to configure RIP version 2 between routers R2 and R3, they must share a common subnet. As the diagram reflects, these two routers do not share a common subnet. Therefore, you must create a common subnet over a tunnel. This is a tunneling task without ever explicitly mentioning the word “tunnel”. Task 8.9.4.instructs you to not use “generic” encapsulation, use IP into IP encapsulation. The most common tunnel encapsulation used is GRE. An alternative to GRE encapsulation is IPIP

Implementation:

Configure the tunnel interfaces on R2 and R3:

R2:

```
interface Tunnel0
description R2-R3 VPN
ip address 10.10.11.2 255.255.255.0
tunnel source 172.16.102.1
tunnel destination 172.16.103.1
tunnel mode ipip
```

R3:

```
interface Tunnel0
description R2-R3 VPN
ip address 10.10.11.3 255.255.255.0
tunnel source 172.16.103.1
tunnel destination 172.16.102.1
tunnel mode ipip
```

Verification:

To verify that tunnel is up, issue the **show interface Tunnel0** command:

```
R2#sh inte tun0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
Description: R2-R3 VPN
R2#
```

Ping across the tunnel using **ping 10.10.11.3** command on R2:

```
R2#ping 10.10.11.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.11.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
R2#
```



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”.

8.9 Router Maintenance



HIDDEN ISSUES TO SPOT WITH ROUTER MAINTENANCE

Issue: Configure specific IOS line parameters to the above listed settings on the Catalyst switches.

Solution:

In order to have a Cisco router or IOS enabled switch to supply the “enable” prompt at any time, enter the following command under the line configuration mode that reflects how the user will access the device (access via console, aux or vty): “privilege level 15”. When a line configuration mode is set with “privilege level 15”, whoever accesses that IOS device via that method – console, aux, vty – will be placed in “enable” mode immediately. To eliminate the prompting of a username or password when accessing an IOS enable device, enter the following command under the line configuration mode that reflects how the user will access the device (access via console, aux or vty): “no login”. To eliminate the termination of a session after the expiration of a specified amount of time, enter the following command under the line

configuration mode that reflects how the user will access the device (access via console, aux or vty):
“exec-timeout 0 0”.

Implementation:

Configure the line vty 5 using the mentioned parameters:

```
line vty 5
  exec-timeout 0 0
  privilege level 15
  no login
```

Issue: Wait only for the minimal time if the IP address entered with the TELNET command is mistyped.

Solution:

When you enter the wrong IP address with the TELNET command, it is very likely the IOS command prompt will hang for an extended amount of time. This is due to the default setting of an IOS service called “tcp syn wait-time”. This IOS parameter sets a time period to wait for a TCP session to get established. If a router initiates a TCP session, it sends out a TCP segment with the SYN bit set. The router will wait the “syn wait-time” period for a return TCP segment with SYN ACK flags set. If it does not receive a return TCP SYN ACK segment, the router initiated TCP session times out. The default setting for “syn wait-time” is 30 seconds. You can set it to as low as 5 seconds. To fulfill the requirements of this task, set “ip tcp syn wait-time 5” in global configuration mode.

Implementation:

Issue the **ip tcp synwait-time 5** command.



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”.

8.10 IP Features



HIDDEN ISSUES TO SPOT WITH IP FEATURES

Issue: *On router R3 configure solution recommended to improve TCP performance in network paths with large bandwidth, long-delay characteristics that are called Long Fat Networks (LFNs).*

Solution:

A larger window size is recommended to improve TCP performance in network paths with large bandwidth, long-delay characteristics that are called Long Fat Networks (LFNs). This TCP Window Scaling enhancement provides that support.

The Cisco IOS window scaling feature complies with RFC 1323, *TCP Extensions for High Performance*. The maximum window size has been increased to 1,073,741,823 bytes. The larger scalable window size will allow TCP to perform better over LFNs.

Implementation:

Use the following command in global configuration mode:

```
ip tcp window-size 65536
```

The number must be over 65535 to enable LFN scaling.



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".

8.11 IPv6 IGP



HIDDEN ISSUES TO SPOT WITH IPv6 IGP

Issue: *Configure OSPFv3 area 25 on the network between R2 and R5. Make it area 0. Configure OSPFv3 area 16 on the network between R1 and R6. Make it area 0.*

Solution:

These two separate OSPF domains are not connected to each other in term of OSPF by any means. Therefore, you can have area 0 in each one of them.

OSPF is configured on interface level (example from R1):


```
interface FastEthernet0/0.16
 encapsulation isl 16
 ip address 172.16.16.1 255.255.255.0
 no ip redirects
 no snmp trap link-status
 ipv6 address FEC0::16:1/125
 ipv6 ospf 16 area 0
```

Once you have configured OSPF in both areas, it is not possible to ping from one to another before you are done with IPv6 BGP section. BGP will connect two separate IPv6 networks and exchange prefixes. Mutual redistribution will be necessary between IGP and BGP in this case.

Verification:

Verify that OSPF adjacency is up:

```
R1#sh ipv6 ospf neighbors
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
172.16.106.1	1	FULL/DR	00:00:30	16	FastEthernet0/0.16

And that you are receiving prefixes from another router:

```
R1#sh ipv6 route ospf
```

```
IPv6 Routing Table - 10 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
O   FEC0::106:0/125 [110/2]
    via FE80::2B0:64FF:FEEF:9B80, FastEthernet0/0.16
O   FEC0::106:8/125 [110/2]
    via FE80::2B0:64FF:FEEF:9B80, FastEthernet0/0.16
O   FEC0::106:10/125 [110/2]
    via FE80::2B0:64FF:FEEF:9B80, FastEthernet0/0.16
O   FEC0::106:18/125 [110/2]
    via FE80::2B0:64FF:FEEF:9B80, FastEthernet0/0.16
```



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".

8.12 IPv6 BGP



HIDDEN ISSUES TO SPOT WITH IPv6 BGP

Issue: Configure IPv6 BGP AS 100 on R2. Configure IPv6 BGP AS 16 on R1. Connect BGP AS over link level address. Do not configure IPv6 on any adapter configured with IPv4 address for this interconnection. Do not use EBGP multihop. Solution must utilize Frame Relay physical layer only.

Solution:

BGP does not have different global configuration command for IPv6. It uses address-family subcommand to group IPv6 configuration together. To fulfill the second part of this task, you have to create IPv6inIP tunnel. At this point, it is clear which traffic is mentioned in "Frame Relay QoS" task. Do not use any global IPv6 addresses on the tunnel, link level addresses only will be enough. EBGP session will be established over these link level addresses.

R1:

```
interface Tunnell2
no ip address
ipv6 address FE80::124:1 link-local
tunnel source Serial0/0
tunnel destination 172.16.124.2
tunnel mode ipv6ip

router bgp 16
no synchronization
bgp log-neighbor-changes
neighbor FE80::124:2 remote-as 100
neighbor FE80::124:2 update-source Tunnell2
no auto-summary
!
address-family ipv6
neighbor FE80::124:2 activate
no synchronization
exit-address-family
```

R2:

```
interface Tunnell2
no ip address
ipv6 address FE80::124:2 link-local
tunnel source Serial0/0
tunnel destination 172.16.124.1
tunnel mode ipv6ip

router bgp 100
no synchronization
bgp cluster-id 25
bgp log-neighbor-changes
neighbor 172.16.25.7 remote-as 100
neighbor 172.16.25.7 update-source Loopback102
neighbor 172.16.25.7 route-reflector-client
```

```

neighbor 172.16.104.1 remote-as 100
neighbor 172.16.104.1 update-source Loopback102
neighbor 172.16.104.1 route-reflector-client
neighbor FE80::124:1 remote-as 16
neighbor FE80::124:1 update-source Tunnel12
no auto-summary
!
address-family ipv6
neighbor FE80::124:1 activate
no synchronization
exit-address-family

```

Verification:

Verify that BGP session is up:

```

R2#sh bgp ipv6 unicast summary
BGP router identifier 172.16.102.1, local AS number 100
BGP table version is 23, main routing table version 23
7 network entries using 1043 bytes of memory
7 path entries using 532 bytes of memory
6/3 BGP path/bestpath attribute entries using 744 bytes of memory
3 BGP rinfo entries using 72 bytes of memory
3 BGP AS-PATH entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2463 total bytes of memory
BGP activity 17/8 prefixes, 21/10 paths, scan interval 60 secs

Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
FE80::124:1   4   16   2992   2993    23    0    0 2d01h    1

```

Issue: Summarize both AS links to /96 prefix and advertise only this prefix to the other peer. Use BGP configuration to perform summarization.

Solution:

To do this, mutually redistribute BGP and OSPF on R1 and R2. This will provide IPv6 reachability and allow you to use aggregation in BGP.

Example on R2:

```

router bgp 100
no synchronization
bgp cluster-id 25
bgp log-neighbor-changes
neighbor 172.16.25.7 remote-as 100
neighbor 172.16.25.7 update-source Loopback102
neighbor 172.16.25.7 route-reflector-client
neighbor 172.16.104.1 remote-as 100
neighbor 172.16.104.1 update-source Loopback102
neighbor 172.16.104.1 route-reflector-client
neighbor FE80::124:1 remote-as 16
neighbor FE80::124:1 update-source Tunnel12

```

```

no auto-summary
!
address-family ipv6
neighbor FE80::124:1 activate
aggregate-address FEC1::/96 summary-only
redistribute ospf 25 metric 1 include-connected
no synchronization
exit-address-family
!
ip http server
no ip http secure-server
ip classless
!
!
!
ipv6 router ospf 25
log-adjacency-changes
redistribute bgp 100 metric 1 route-map IPV6-Ext-Only
!
!
!
ipv6 prefix-list IPV6-Local seq 5 permit FEC1::/96
route-map IPV6-Ext-Only deny 10
match ipv6 address prefix-list IPV6-Local
!
route-map IPV6-Ext-Only permit 20
  
```

You need to filter the feedback of FEC1::/96 as type-5 LSA into the same OSPF that originated more specific routes because it doesn't make any sense.

Verification:

Verify that you can ping from one OSPF network to another:

```

R5#ping fec0::106:11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::106:11, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/77/84 ms
R5#
  
```

Verify that you have only /96 aggregate for another network:

```

R5#sh ipv6 route
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
L   FE80::/10 [0/0]
   via ::, Null0
OE2  FEC0::/96 [110/1]
   via FE80::204:9AFF:FEE4:13C1, FastEthernet0/0
C   FEC1::25:0/125 [0/0]
  
```

```

    via ::, FastEthernet0/0
L   FEC1::25:5/128 [0/0]
    via ::, FastEthernet0/0
C   FEC1::105:0/125 [0/0]
    via ::, Loopback1051
L   FEC1::105:1/128 [0/0]
    via ::, Loopback1051
C   FEC1::105:8/125 [0/0]
    via ::, Loopback1059
L   FEC1::105:9/128 [0/0]
    via ::, Loopback1059
C   FEC1::105:10/125 [0/0]
    via ::, Loopback10511
L   FEC1::105:11/128 [0/0]
    via ::, Loopback10511
C   FEC1::105:18/125 [0/0]
    via ::, Loopback10519
L   FEC1::105:19/128 [0/0]
    via ::, Loopback10519
L   FF00::/8 [0/0]
    via ::, Null10
R5#

```



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”.

8.13 QOS



HIDDEN ISSUES TO SPOT WITH QOS

Issue: Provide a QOS solution to VPN users allowing traffic to be shaped between R2 and R3 to 30 Kbps.

Solution:

Apply generic traffic shaping to the VPN tunnel interfaces configured on routers R2 and R3. The following interface configuration command will fulfill the requirements of this task: traffic-shape rate 30000.

Implementation:

Issue the **traffic-shape rate 30000** command under tunnel interface. The rest of the parameters will follow up automatically after you press Enter.

Verification:

Issue the **show traffic-shape tunnel0** command:

R3#sh traffic-shape tun0

Interface	Tu0	Access List	Target Rate	Byte Limit	Sustain bits/int	Excess bits/int	Interval (ms)	Increment (bytes)	Adapt Active
-			30000	1980	7920	7920	264	990	-

R3#



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".

8.14 IOS Specialties



HIDDEN ISSUES TO SPOT WITH IOS SPECIALTIES

Issue: Configure router R1 to block login for 120 seconds after 5 failed attempts to login within 120 seconds. Report each successful and unsuccessful login to log. Allow telnet connection from Lo106 of R6 to pass at all times.

Solution:

This 12.4 feature is called "Cisco Login Enhancements". To configure it create a user, instruct logon facility what to do in case you have received a certain number of bad logon attempts and which source address is exempt from this rule:

```
login block-for 120 attempts 5 within 120
login quiet-mode access-class 10
login on-failure log
login on-success log
```

NB! Do not forget to setup vty to logon locally. If you have password on vty line, login enhancements will not be engaged.

Verification:

Attempt 5 unsuccessful logins from R6 to R1 within 2 minutes. On 6th attempt R1 will not accept login at all. However, if you specify /source-interface Loopback106 – it will let it through. You can also type "show login" on R1 at any time to check the current status:

```
R1#sh login
A default login delay of 1 seconds is applied.
Quiet-Mode access list 10 is applied.
All successful login is logged.
All failed login is logged.

Router enabled to watch for login Attacks.
If more than 5 login failures occur in 120 seconds or less,
```

```
logins will be disabled for 120 seconds.  
  
Router presently in Normal-Mode.  
Current Watch Window  
  Time remaining: 75 seconds.  
  Login failures for current window: 0.  
Total login failures: 5.
```

Read more about this feature on:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_4/qt_login.htm#wp1027265



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”.

8.15 Catalyst specialties



HIDDEN ISSUES TO SPOT WITH CATALYST SPECIALTIES

Issue: The workstations connected to ports Fa0/10 and Fa0/11 on CAT2 can communicate with the rest of the network but they cannot exchange traffic between each other. You cannot use filtering access-lists.

Solution:

In order to fulfill this requirement configure the following interface configuration command on ports Fa0/10 and Fa0/11 on CAT2: “switchport protected”. When you configure two or more ports with the “switchport protected” command, these ports will not exchange any unicast, broadcast or multicast traffic.

Implementation:

```
interface FastEthernet0/10  
  description Protected Workstations  
  switchport access vlan 30  
  switchport mode access  
  switchport protected  
  
interface FastEthernet0/11  
  description Protected Workstations  
  switchport access vlan 30  
  switchport mode access  
  switchport protected
```



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”.

8.16 Address Administration



HIDDEN ISSUES TO SPOT WITH ADDRESS ADMINISTRATION

Issue: Allow TFTP, DNS, BOOTP and NTP broadcast to be propagated on the links between R3, R4, CAT1 and CAT2. The solution should protect against broadcast storms.

Solution:

Emphasis must be placed on the words "Allow ... broadcasts to be propagated". Therefore, we cannot convert the broadcast traffic into unicast traffic with an IP helper command. We must make sure that it remains broadcast traffic when the traffic is forwarded along the specified path. One clue to solving this problem is: All of the traffic specified to be broadcasted is UDP traffic. The IOS possesses a technique for forwarding udp broadcasts. It involves the following global configuration command: "ip forward-protocol". It requires that you configure a bridge-group command on all participating interfaces and that you assure that spanning-tree bpdus are forwarded along the specified path. When you configure the "ip forward-protocol" command with the bridge-group command, you do not need to enable crb or irb. This configuration will not bridge all IP traffic. It will bridge only the udp broadcast traffic specified with the "ip forward-protocol" command.

Implementation:

Issue the following commands. You may notice that all or most of them are forwarded by default.

```
ip forward-protocol udp tftp
ip forward-protocol udp domain
ip forward-protocol udp bootpc
ip forward-protocol udp bootps
ip forward-protocol udp ntp
```

Issue: The solution should protect against broadcast storms.

Solution:

Make sure you enter the following command to protect against broadcast storms: "**ip forward-protocol spanning-tree**"

Implementation:

Create bridge 1 and use spanning-tree protocol vlan-bridge by issuing the **bridge 1 protocol vlan-bridge** command.

Attach bridge 1 to participating interfaces by issuing the **bridge-group 1** command under the interface.

Issue the **ip forward-protocol spanning-tree** command.

Verification:

Make sure that the spanning tree keeps the loop-free topology:

```
R3#show spanning-tree 1 | inc Port
Port 3 (FastEthernet0/0) of Bridge group 1 is blocking
Port path cost 19, Port priority 128, Port Identifier 128.3.
Port 11 (Virtual-Template1) of Bridge group 1 is down
Port path cost 19, Port priority 128, Port Identifier 128.11.
Port 12 (Virtual-Access1) of Bridge group 1 is forwarding
Port path cost 19, Port priority 128, Port Identifier 128.12.
```



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”.

8.17 Multicast



HIDDEN ISSUES TO SPOT WITH MULTICAST

Issue: Use Auto-RP discovery protocol.

Solution:

If you use the auto-RP discovery protocol, you must configure all interfaces participating in IP multicasting to “pim sparse-dense mode”. Auto-RP uses PIM Dense Mode to propagate Rendezvous Point related traffic to participating multicast routers.

Implementation:

Configure multicast routing on the router itself by issuing the **ip multicast-routing** command.

Configure PIM sparse-dense mode on all interfaces that participate in multicast routing by issuing the **ip pim sparse-dense-mode**.

Issue: Configure R3 to be both the Rendezvous Point and the Mapping Agent.

Solution:

To make R3 the Rendezvous Point enter the following global configuration command: “ip pim send-rp-announce loopback 0 scope 255 group-list 1” The interface that you select must have PIM configured on it and it must be reachable by all multicast routers. The scope sets the TTL for all packets generated by the Rendezvous Point. The group-list is tied to an access-list that limits the number of multicast group you want this router to be the candidate RP for. If you do not specify this, the router will be the candidate RP for all multicast groups. Not only must you configure a single candidate RP, you must also configure a

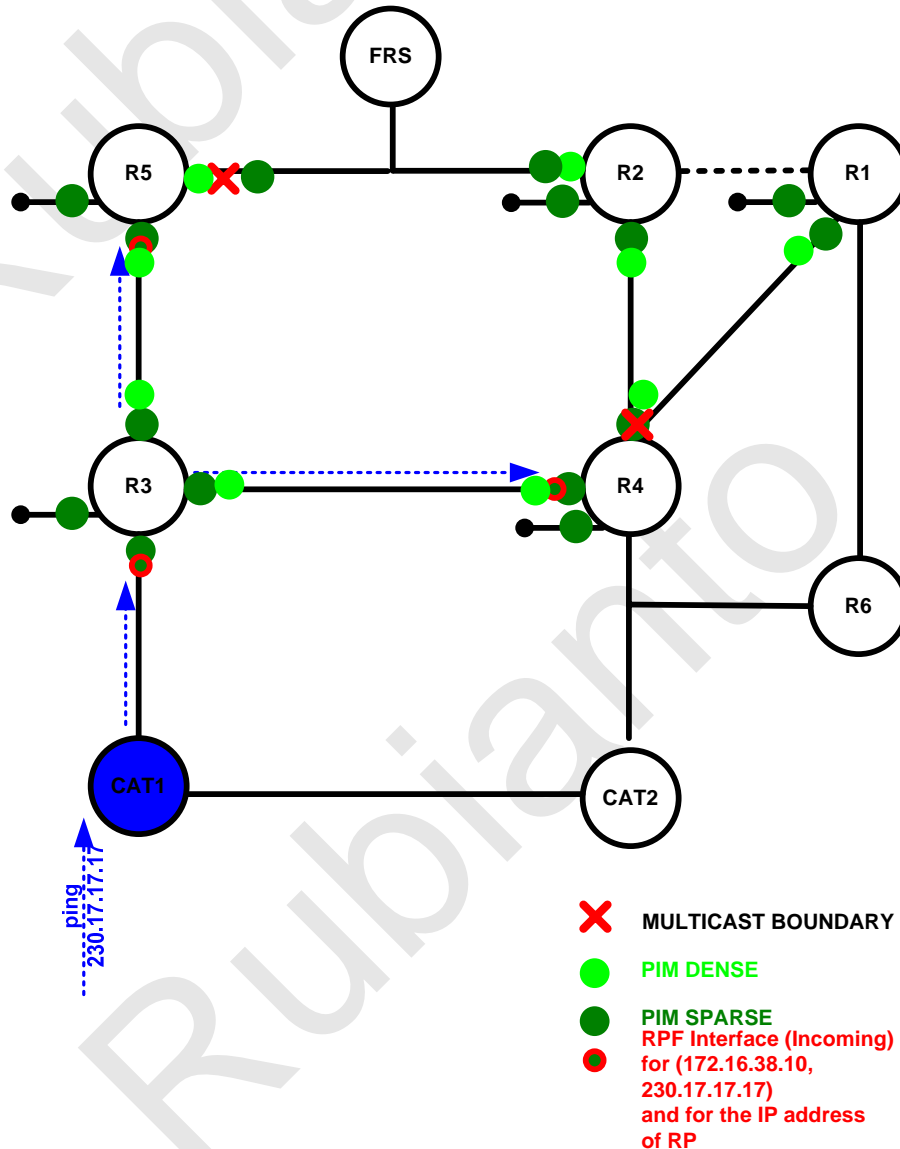
Mapping Agent. Task 8.17.3 instructs you to make R3 the Mapping Agent as well. This is performed with the following global configuration command: "ip pim send-rp-discovery loopback 0 scope 255". The interface that you select must have PIM configured on it and it must be reachable by all multicast routers. The scope sets the TTL for all packets generated by the Mapping Agent.

Implementation:

On router R3 configure MA and RP by issuing the following commands:

```

ip pim send-rp-announce Loopback0 scope 255
ip pim send-rp-discovery Loopback0 scope 255
  
```



Verification:

Verify that RP is propagated to all multicast routers by issuing the **show ip pim rp mapping** command:

```
R2#sh ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.103.1 (?), v2v1
    Info source: 172.16.34.3 (?), elected via Auto-RP
    Uptime: 3d22h, expires: 00:02:49
R2#
```

Issue: Join one of the loopback interfaces of the routers R3, R4, R5, R1 and R2 to group 230.17.17.17.

Solution:

Configure the command “ip igmp join-group 230.17.17.17: on a selected loopback interface on each of the listed routers. The restrictions of Multicast task lead to configure a multicast boundaries on R5 and R4 to not to allow multicast traffic destined to group 230.17.17.17 to pass thru them. This is often done to reuse the same multicast groups in different parts of the network for different purposes.

Implementation:

Issue the **ip igmp join-group 230.17.17.17** command and also issue the **ip pim sparse-dense-mode** under these loopbacks, otherwise they will not respond to ping.

Configure multicast boundary on R4:

R4:

```
interface Serial0/0
 ip address 172.16.124.4 255.255.255.0
 ip pim nbma-mode
 ip pim sparse-dense-mode
 ip multicast boundary Multicast-filter

ip access-list standard Multicast-filter
 deny 230.17.17.17
 permit 224.0.0.0 15.255.255.255
```

R5 may not have multicast enabled on its FastEthernet, therefore there is no need to configure multicast boundary on R5.



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”.