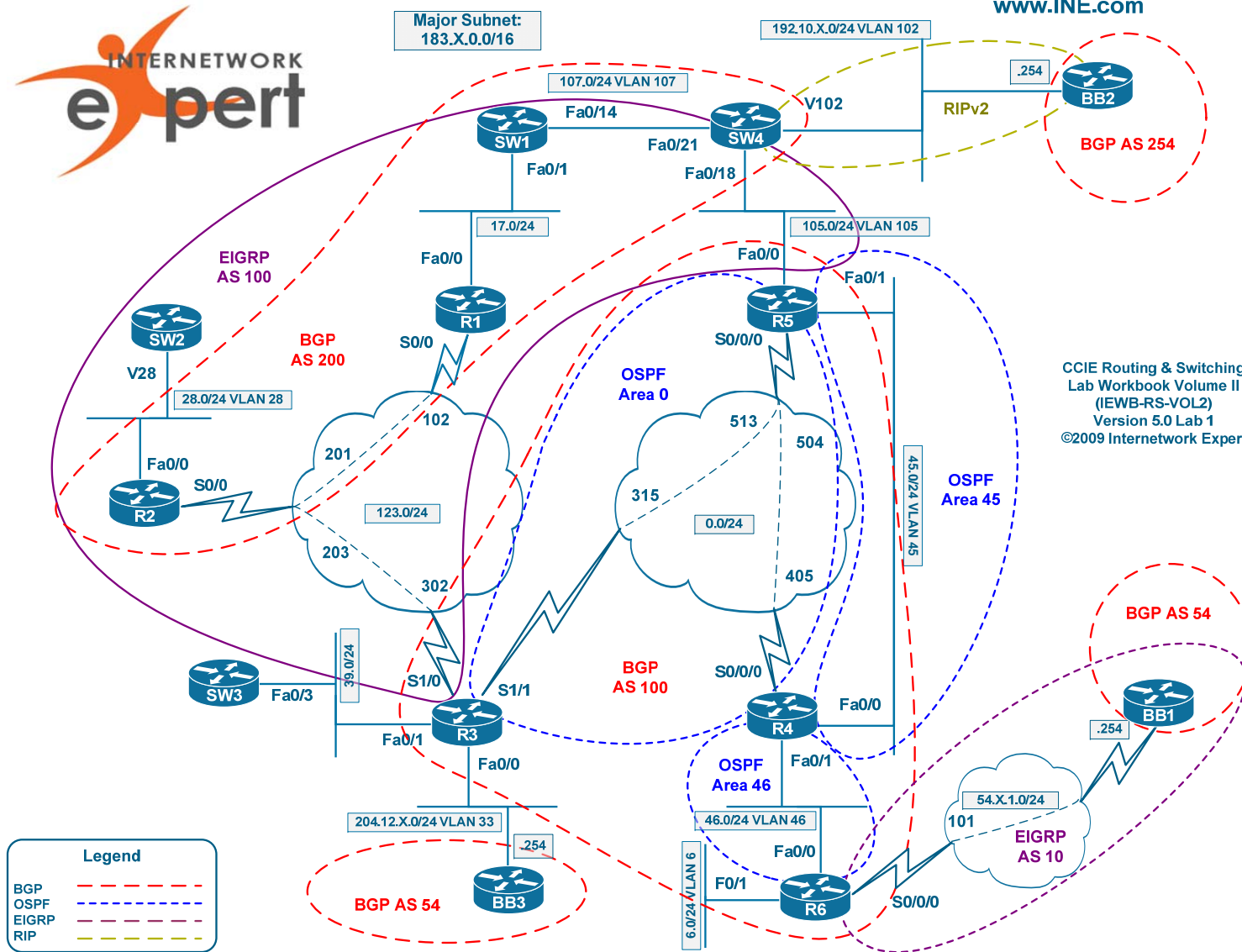




www.INE.com



IEWB-RS Volume 2 Lab 1

Difficulty Rating (10 highest): 6

Lab Overview:

The following scenario is a practice lab exam designed to test your skills at configuring Cisco networking devices. Specifically, this scenario is designed to assist you in your preparation for Cisco Systems' CCIE Routing & Switching Lab exam. However, remember that in addition to being designed as a simulation of the actual CCIE lab exam, this practice lab should be used as a learning tool. Instead of rushing through the lab in order to complete all the configuration steps, take the time to research the networking technology in question and gain a deeper understanding of the principles behind its operation.

Lab Instructions:

Prior to starting, ensure that the initial configuration scripts for this lab have been applied. For a current copy of these scripts, see the Internetwork Expert members' site at <http://members.INE.com>. If you have any questions related to the scenario solutions, visit our CCIE support forum at <http://IEOC.com>.

Refer to the attached diagrams for interface and protocol assignments. Any reference to X in an IP address refers to your rack number, while any reference to Y in an IP address refers to your router number.

Upon completion, all devices should have full IP reachability to all networks in the routing domain, including any networks generated by the backbone routers unless explicitly specified.

Lab Do's and Don'ts:

- Do not change or add any IP addresses from the initial configuration unless otherwise specified or required for troubleshooting
- If additional IP addresses are needed but not specifically permitted by the task use IP unnumbered
- Do not change any interface encapsulations unless otherwise specified
- Do not change the console, AUX, and VTY passwords or access methods unless otherwise specified
- Do not use any static routes, default routes, default networks, or policy routing unless otherwise specified
- Save your configurations often

Grading:

This practice lab consists of various sections totaling 79 points. A score of 64 points is required to pass the exam. A section must work 100% with the requirements given in order to be awarded the points for that section. No partial credit is awarded. If a section has multiple possible solutions, choose the solution that best meets the requirements.

Point Values:

The point values for each section are as follows:

Section	Point Value
Layer 2 Technologies	4
IPv4	12
IPv6	7
MPLS VPN	6
Multicast	7
Security	10
Network Services	21
QoS	12

GOOD LUCK!

1. Layer 2 Technologies

Some Layer 2 settings have been pre-configured for you; VLANs have been created and assigned to the switchports according to the diagram supplied with the task.

1.1 Layer 2 Features

- Ensure SW2 is the primary STP root bridge for VLAN105
- Configure the network in such a way to ensure that VLAN 102's traffic never traverses SW3.
- Additionally ensure that no other VLAN traffic follows the path that VLAN 102 does through the switched network.
- Ports Fa0/7 on SW1 and Fa0/7 on SW2 connect to your corporate conference room. Recently your network administrator has been getting complaints that when users plug their laptops into the conference room it either takes a very long time to get an IP address from the DHCP server, or the DHCP request times out. After further investigation, you have discovered that spanning-tree convergence time is to blame.
- In order to resolve this configure SW1 and SW2 so that users in VLAN 28 do not have to wait for spanning-tree's forwarding delay when they connect to the network.
- Ensure that any ports in VLAN 28 will be shut down if a device running spanning-tree protocol is detected.
- Additionally, the network administrator has requested the ports SW1 Fa0/7 and SW2 Fa0/7 should not be able to communicate directly with each other within VLAN 28.
- These ports should still be allowed to communicate with R2's F0/0 interface but not SW2's V28 interface.
- You are allowed to additionally create and use VLAN 281 for this purpose.
- Do not use a VLAN ACL to accomplish this.

4 Points

2. IPv4

Some IGP protocol settings and IP addressing have been preconfigured for you. Notice that there might be some issues deliberately introduced into the initial configurations. Use the diagram as you reference to fix those.

2.1 OSPF

- Ensure that R5 is always elected the Designated Router for the segment between R3, R4 and R5.
- Ensure that host devices running OSPF on the segment between R4 and R5 cannot intercept the OSPF communication between R4 and R5.
- Advertise VLAN 6 into OSPF on R6; do not use the `network` or `ip ospf` statements to accomplish this.
- Configure the network so that traffic is only sent over this Ethernet segment if the Frame Relay circuit between R4 and R5 is down.
- Do not use the `backup interface` command to accomplish this.
- To minimize downtime in the event of a failure configure the network so that R4 can detect a loss of the Frame Relay circuit to R5 within 1 second

4 Points

2.2 IGP Features

- Advertise VLAN 33 and R3's interface Fa0/1 into the EIGRP domain.
- These prefixes should appear as follows throughout the EIGRP domain:

```
D EX    204.12.X.0 [170/...  
D EX    183.X.39.0 [170/...
```

- In order to ensure that all routes learned over the Frame Relay cloud via EIGRP are legitimate configure R6 to use the most secure authentication for any neighbor relationships formed on this interface.
- Use key number 1 with a password of CISCO for this authentication.
- In order to protect against false route injection from RIP as well, configure SW4 to use the strongest authentication on any RIP updates received on this Ethernet segment using key 1 and the password CISCO.
- Redistribute between RIP and EIGRP on SW4.
- Redistribute between OSPF and EIGRP on R3, R5, and R6.
- R5 should still be able to reach this prefix if the Frame Relay circuit between R2 and R3 is down.

3 Points**2.3 BGP Bestpath Selection**

- For the purposes of load-sharing and redundancy, AS 100 has multiple connections to AS 54. In order to maximize throughput your corporate policy dictates that all traffic destined for prefixes originated in AS 54 should traverse the Frame Relay link between R6 and BB1.
- In the case that the Frame Relay link between R6 and BB1 goes down AS 100 should still have reachability to AS 54 via the Ethernet segment between R3 and BB3.
- Do not modify weight to accomplish this.
- Configure a new Loopback interface on R1 with the IP address 150.X.11.1/24 and advertise it into BGP.
- Configure AS 200 so that all traffic from AS 100 destined to this prefix traverses the Ethernet segment between SW4 and R5.
- In the case that the link between SW4 and R5 is down traffic destined for the 150.X.11.0/24 prefix should transit the Frame Relay link between R2 and R3.
- Do not use AS-Path prepending to accomplish this.

4 Points**3. IPv6****3.1 IPv6 Addressing**

- The network administrator has requested that VLAN 46 and VLAN 105 be configured to support a test deployment of IPv6.
 - Address R4's interface attached to VLAN 46 with the IPv6 network 2001:CC1E:X:404::/64.
 - Address R5's interface attached to VLAN 105 with the IPv6 network 2001:CC1E:X:505::/64.
- The host addresses on these interfaces should be derived from the interface's MAC address.
- In order to connect these two isolated networks you have decided to tunnel IPv6 over your existing IPv4 infrastructure, however you want to ensure that this connection can survive a failure of the Frame Relay circuit between R4 and R5.

- To accomplish this configure a tunnel between R4 and R5 using their Loopback0 interfaces as the source.
- The tunnel should use the addresses 2001:CC1E:X:4545::Y/64.
- This tunnel should use a mode that specifies IPv6 as the passenger protocol and IPv4 as the encapsulation and transport protocol.
- Enable EIGRPv6 on VLAN 46, VLAN 105 and the tunnel interfaces.
- Use 45 as the AS# for the processes on both R4 and R5.

4 Points

3.2 IPv6 Multicast Basics

- Configure R6 to join the multicast group FF06::6 on its connection to R4.
- Configure R4 to accept MLD messages only for the group range FF06::/16.
- Use R5 as the PIM RP and ensure multicast packets from R5 can reach R6.
- Do not configure the RP address statically and do not configure the same router as a BSR and RP.

3 Points

4. MPLS VPN

4.1 LDP

- Configure MPLS label redistribution between R4, R5 and R6 using the industry-standard protocol.
- Make sure the LDP labels are redistributed even if the primary Frame-Relay link between R4 and R5 fails.
- Configure so that the labels are only generated for the respective router's Loopback0 interfaces.

3 Points

4.2 VPN

- Using the RD 100:5 and 100:6 configure two VRFs named VPN_A and VPN_B in R5 and R6 respectively.
- Use the same route-target values to tag the respective routes.

- Create two new Loopback interfaces in R5 and R6 with the IP addresses 172.16.5.5/24 and 192.168.6.6/24 and assign them to VRFs VPN_A and VPN_B respectively.
- Configure R5 and R6 to provide reachability between the two subnets.

3 Points

5. IP Multicast

5.1 RP Assignment

- Discover the active multicast topology using the respective show commands.
- Configure R3 to announce its most reliable interface as the RP for all multicast groups using Auto-RP protocol.
- R2 should be responsible for group to RP mappings.

2 Points

5.2 Multicast Testing

- There is a Windows® Media Server located on VLAN 28 that is streaming a video feed into your network, however your administrators have been getting complaints from users on VLAN 105 that they are unable to receive this feed.
- In order to help track down the source of this problem configure R5's Ethernet interface attached to VLAN 105 to join the multicast group 226.26.26.26.
- Ensure that R5 responds to ICMP echo-requests sourced from VLAN 28 which are sent to 226.26.26.26.
- You are allowed to use one static multicast route to accomplish this.

3 Points

5.3 Multicast Filtering

- Development engineers are testing a new multicast application located on VLAN 28 prior to its deployment in your network. This application is generating random multicast streams destined for addresses in the administratively scoped multicast range.

- In order to prevent this test traffic from being unnecessarily forwarded throughout the network configure R3 so that hosts in VLAN 33 are not allowed to join any groups in this range.

2 Points

6. Security

6.1 Denial of Service Tracking

- Your network administrators have been getting complaints from users that the web server with the IP address 183.X.28.100 is inaccessible. After further investigation you have determined that this server is undergoing a TCP SYN attack.
- In order to assist in tracking down the source of this attack configure R3 and SW4 to generate a log message when HTTP SYN packets are received on VLANs 33 or 102 respectively that are destined for 183.X.28.100.
- These log messages should include the MAC address of the device which forwarded the packet onto the segment.

3 Points

6.2 Spoof Prevention

- After reviewing your log files you have determined that the DoS attack on your web server came from hosts with spoofed source addresses.
- To help prevent this type of attack in the future configure your network so that traffic will not be accepted from BB1, BB2, or BB3 if it sourced from your address space 183.X.0.0/16.

2 Points

6.3 Information Leaking

- Your security manager is concerned with potential network reconnaissance attacks originating from behind BB2.
- In order to minimize information exposure, configure SW4 not to notify hosts behind BB2 of any networks that it does not know about.
- Additionally, SW4 should not disclose its network mask to any host on the VLAN 102 segment.

2 Points

6.4 Control Plane Protection

- Configure R4 to drop and transit IP packets with the TTL lower than 3.
- Log the drop events to the system console and the router's memory buffer.

3 Points

7. Network Services

7.1 RMON

- In order to help detect possible flood attacks in the future configure R2 to generate an SNMP trap when the interface input unicast packets (ifEntry.11.1) value rises more than 15000 per minute, and when the value falls back below 5000 per minute.
- The sampling interval should be every sixty seconds.
- When the 15000 threshold is breached an event should be generated that reads "Above 15000 for ifInUcastPkts".
- When the value falls back to 5000 an event should be generated that reads "Below 5000 for ifInUcastPkts".
- The server to send these SNMP traps to is 183.X.17.100.
- This server will be expecting the community string to be IETRAP.

3 Points

7.2 NTP

- After implementing syslog logging your NOC engineers have noticed inconsistent timestamps on your device logs. In order to resolve this problem you have decided to maintain consistent time by implementing Network Time Protocol.
 - Configure R3 and R6 to get network time from BB3 and BB1 respectively.
 - Configure R1, R2, and SW1 to get network time from R3.
 - Configure R4, R5, and SW4 to get network time from R6.

- R3 should fail over and get network time from R6 in the event that BB3 becomes unavailable.
- R6 should fail over and get network time from R3 in the event that BB1 becomes unavailable.

3 Points

7.3 NTP Authentication

- In order to ensure that your internal time servers are not being spoofed, configure R3 and R6 to be authenticated using the MD5 password CISCO.

2 Points

7.4 Traffic Accounting

- Your design team would like to implement a new QoS policy using IP precedence on the Frame Relay circuit between R2 and R3. However, prior to implementing this new policy they need to know if packets transiting this link already have an IP precedence value set.
- To accomplish this configure R2 and R3 to collect usage statistics on packets with an IP precedence value and store them locally.
- R2 and R3 should store up to 50000 of these entries in their memory.

3 Points

7.5 Gateway Redundancy

- Your administrators are concerned about default gateway redundancy for the hosts located on VLAN 105. In order to allow them to survive a network failure you have assigned the virtual IP address 183.X.105.254 as the default gateway for these hosts.
- As long as R5's Frame Relay connection is up it should respond to ARP requests sent to this IP address.
- In the event that R5's Frame Relay connection goes down hosts should use SW4 as their default gateway.
- Do not use VRRP or GLBP to accomplish this.
- Configure your network to reflect this policy.

3 Points

7.6 Network Address Translation

- Your operations team does not want BB3 and its customers to have specific reachability information about your network. Instead, BB3 should only have reachability to your hosts if a connection is initiated from inside your network.
- Configure R3 to reflect this policy.
- Ensure that all devices in the 183.X.0.0/16 network can successfully ping BB3.

3 Points

7.7 Embedded Event Management

- Configure R6 to generate a syslog message once its Frame-Relay output utilization exceeds 80% of the total interface bandwidth.
- The sylog message should contain the interface name and the current interface load.
- Ensure your configuration responds to the transmit load changes as fast as possible with Cisco IOS routers.
- Do not use the RMON feature to accomplish this.

4 Points

8. QoS

8.1 Frame Relay Traffic Shaping

- You have been noticing drops on R5's connection to the Frame Relay cloud. After further investigation, you have discovered that R5 has been overwhelming R3 and R4's connections to the Frame Relay cloud. Configure Frame Relay Traffic Shaping on R5 in order to resolve this issue.
- R5's connection to the Frame Relay cloud supports a transmission rate of 1536Kbps.
- R5 should send at an average rate of 128Kbps on DLCI 513 to R3.
- R5 should send at an average rate of 512Kbps on DLCI 504 to R4.
- In the case that the Frame Relay cloud notifies R5 of congestion it should reduce its sending rate to no lower than 96Kbps for the DLCI to R3 and 384Kbps for the DLCI to R4.

- In the case that R5 has accumulated credit it should be allowed to burst up to the maximum transmission rate supported on the circuit to R4.
- Bursting on the circuit to R3 should not be allowed.
- Assume an interval (Tc) of 50ms.

3 Points

8.2 Rate Limiting

- One of your NOC engineers has noticed suspiciously high utilization on the Ethernet segment of R1. After further investigation you have found that a large number of ICMP packets have been traversing this link.
- In order to alleviate congestion configure R1 so that it does not send more than 128Kbps of ICMP traffic out this interface.
- Allow for a burst of 1/4th of this rate.

3 Points

8.3 CBWFQ

- Your company plans to reduce expenses by sending PSTN calls to the remote office connected to R5 across the WAN. Currently the WAN link is used primarily for data transfers and remote desktop application.
- Configure R5 to allocate 64Kbps of PVC bandwidth to VoIP bearer traffic, which is marked as DSCP EF.
- At the same time, guarantee 30% of remaining bandwidth to Citrix application traffic.
- Set the queue depth for the Citrix traffic class to 16 packets.
- All other remaining traffic should receive flow-based fair scheduling.

3 Points

8.4 Catalyst QoS

- R6 should see OSPF packets sent from R4 marked with IP Precedence value of 3.
- Limit the aggregate traffic rate for packets sent to R6 to 4 Mbps but ensure the exceeding packets are queued.
- At the same time, traffic flows generated by HTTP server responses should be limited to 1Mbps.
- Use DSCP value of CS2 to mark the HTTP packets for this task.
- Tune the queue depth for HTTP packets to 64 packets.
- Do not configure any of the routers to accomplish this.

3 Points

IEWB-RS Volume 2 Lab 1 Solutions

Task 1.1

SW2:

```
spanning-tree vlan 105 root primary
```

SW1 and SW2:

```
interface FastEthernet0/7
 spanning-tree portfast
 spanning-tree bpduguard enable
```

Note

Spanning Tree manipulation by means of VLAN filtering implements traffic engineering.

SW1:

```
interface FastEthernet0/21
 switchport trunk allowed vlan 102
```

 **Quick Note**
Only VLAN 102 is allowed.

SW2:

```
interface range Fa0/16 - 18
 switchport trunk allowed vlan 1-101,103-4094
```

SW3:

```
interface range Fa0/16 - 20
 switchport trunk allowed vlan 1-101,103-4094
```

 **Quick Note**
The **switchport trunk allowed vlan except 102** command will produce the same output in the switch's configuration.

SW4:

```
interface FastEthernet0/15
 switchport trunk allowed vlan 102
!
interface range Fa0/19 - 20
 switchport trunk allowed vlan 1-101,103-4094
```

Note

Private VLAN configuration part – ports 0/7 and 0/8 are segregated using a single secondary isolate VLAN.

SW1:

```
vlan 281
 private-vlan isolated
!
vlan 28
 name VLAN_28
 private-vlan primary
 private-vlan association 281
```

```
!  
interface FastEthernet0/7  
  switchport private-vlan host-association 28 281  
  switchport mode private-vlan host  
  
SW2:  
vlan 281  
  private-vlan isolated  
!  
vlan 28  
  name VLAN_28  
  private-vlan primary  
  private-vlan association 281  
!  
!  
interface FastEthernet0/2  
  switchport private-vlan mapping 28 281  
  switchport mode private-vlan promiscuous  
!  
interface FastEthernet0/7  
  switchport private-vlan host-association 28 281  
  switchport mode private-vlan host
```

Task 1.1 Breakdown

By default all ports within a VLAN have layer 2 reachability between each other. Private VLANs allow for the separation of a single VLAN into multiple segments or sub-broadcast domains by restricting layer 2 communications within the VLAN. A common implementation for Private VLANs would be to restrict communication between web servers within a VLAN but allow access to a DNS server and their default gateway. Although this configuration could be accomplished using protected ports, protected ports only restrict traffic within a single switch. Private VLANs allow for this configuration to span across multiple switches.

Private VLANs require that the switches to be in VTP transparent mode. There are three types of VLANs that make up a private VLAN. The first one is called the primary VLAN. The other two, community and isolated, are referred to as secondary VLANs. Ports that are assigned to an isolated VLAN can not communicate with other ports at layer 2, with the exception of ports in the primary VLAN. Ports assigned within a community can communicate with other ports assigned within the same community, along with ports assigned to the primary VLAN. This means that layer 2 communication is not permitted between two isolated ports, an isolated port and a port within a community, or between two ports within different communities. Also note that these restrictions exclude trunk ports.

There are three types of ports for Private VLANs. The first one is called a promiscuous port. A promiscuous port can communicate via layer 2 to all other promiscuous ports, isolated ports, and community ports. Promiscuous ports are

assigned to the primary VLAN. The second port type is called an isolated port. Isolated ports can only communicate via layer 2 to promiscuous ports. The last type is called a community port. A community port can talk to other ports that are within the same community and ports that are promiscuous ports.

 **Note**

Private VLAN Guidelines:

- Private VLANs must be configured in the global configuration; the VLAN database mode configuration is not supported for Private VLANs.
- Private VLAN information is not propagated via VTP.
- Isolated and community VLANs do not run their own instance of spanning tree; if fine-tuning of spanning tree is needed the configuration should be applied to the primary VLAN.
- Although Private VLANs restrict layer 2 communication devices may still be able to communicate if their traffic is routed through a layer 3 device.

Task 1.1 Verification

 **Note**

Verify Private VLANs configuration:

```
Rack1SW1#show interfaces fa0/7 switchport | include private|28|281
Administrative Mode: private-vlan host
Administrative private-vlan host-association: 28 (VLAN_28) 281
(VLAN0281)
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
```

```
Rack1SW2#show interfaces fa0/2 switchport | include private|28|281
Administrative Mode: private-vlan promiscuous
Operational Mode: private-vlan promiscuous
Administrative private-vlan host-association: none
```

```
Administrative private-vlan mapping: 28 (VLAN_28) 281 (VLAN0281)
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan:
  28 (VLAN_28) 281 (VLAN0281)
```

```
Rack1SW2#show interfaces fa0/7 switchport | include private|28|281
Administrative Mode: private-vlan host
Administrative private-vlan host-association: 28 (VLAN_28) 281
(VLAN0281)
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
```

For testing purposes we will temporarily change R6's Fa0/0 IP address and VLAN to facilitate the test.

```
Rack1SW2#show running-config interface fa0/6
Building configuration...
```

```
Current configuration : 117 bytes
!
interface FastEthernet0/6
  switchport private-vlan host-association 28 281
  switchport mode private-vlan host
end
```

```
Rack1R6#show running-config interface Fa0/0
Building configuration...
```

```
Current configuration : 98 bytes
!
interface FastEthernet0/0
  ip address 183.1.28.6 255.255.255.0
end
```

```
Rack1R6#ping 183.1.28.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
Rack1R6#ping 183.1.28.8
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 183.1.28.8, timeout is 2 seconds:
```

```
.....  
Success rate is 0 percent (0/5)
```

Rack1SW2#ping 183.1.28.2

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 183.1.28.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Rack1SW2#ping 183.1.28.6

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 183.1.28.6, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

Task 2.1

R3:

```
interface Serial1/1  
 ip ospf priority 0  
!  
interface Serial1/1  
 ip ospf dead-interval minimal hello-multiplier 3
```

R4:

```
interface Serial0/0/0  
 ip ospf priority 0  
!  
interface FastEthernet0/0  
 ip ospf network non-broadcast  
!  
router ospf 1  
 neighbor 183.1.45.5
```

R5:

```
interface FastEthernet0/1  
 ip ospf network non-broadcast  
!  
router ospf 1  
 neighbor 183.1.45.4
```

R6:

```
router ospf 1  
 redistribute connected route-map CONNECTED->OSPF subnets  
!  
 ip prefix-list VLAN_6 permit 183.1.6.0/24  
!  
 route-map CONNECTED->OSPF permit 10  
 match ip address prefix-list VLAN_6
```

R4:

```
interface FastEthernet0/0  
 ip ospf cost 10000  
!
```

```

router ospf 1
  area 45 virtual-link 150.1.5.5
!
interface Serial0/0/0
  ip ospf dead-interval minimal hello-multiplier 3

```

R5:

```

interface FastEthernet0/1
  ip ospf cost 10000
!
router ospf 1
  area 45 virtual-link 150.1.4.4
!
interface Serial0/0/0
  ip ospf dead-interval minimal hello-multiplier 3

```

Task 2.1 Verification

Verify the OSPF configuration:

Rack1R5#show ip ospf interface

```

Serial0/0/0 is up, line protocol is up
  Internet Address 183.1.0.5/24, Area 0
  Process ID 1, Router ID 150.1.5.5, Network Type BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 150.1.5.5, Interface address 183.1.0.5
  No backup designated router on this network
<output omitted>
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 150.1.3.3
    Adjacent with neighbor 150.1.4.4

```

```
<output omitted>
```

```

Loopback0 is up, line protocol is up
  Internet Address 150.1.5.5/24, Area 0
  Process ID 1, Router ID 150.1.5.5, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

```

Rack1R5#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
150.1.3.3	0	FULL/DROTHER	00:00:37	183.1.0.3	Serial0/0/0
150.1.4.4	0	FULL/DROTHER	00:00:38	183.1.0.4	Serial0/0/0

Rack1R4#show ip ospf interface loopback 0

```

Loopback0 is up, line protocol is up
  Internet Address 150.1.4.4/24, Area 0
  Process ID 1, Router ID 150.1.4.4, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

```

Rack1R3#show ip ospf interface loopback 0

```

Loopback0 is up, line protocol is up
  Internet Address 150.1.3.3/24, Area 0
  Process ID 1, Router ID 150.1.3.3, Network Type LOOPBACK, Cost: 1

```

Loopback interface is treated as a stub Host

Rack1R5#show ip route ospf

```
150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.4.4/32 [110/65] via 183.1.0.4, 00:09:06, Serial0/0/0
O       150.1.3.3/32 [110/65] via 183.1.0.3, 00:09:06, Serial0/0/0
```

Rack1R4#show ip route ospf

```
150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.5.5/32 [110/65] via 183.1.0.5, 00:09:40, Serial0/0/0
O       150.1.3.3/32 [110/65] via 183.1.0.3, 00:09:40, Serial0/0/0
```

Verify the OSPF network types on the segment between R4 and R5

Rack1R4#show ip ospf interface FastEthernet 0/0

```
FastEthernet0/0 is up, line protocol is up
  Internet Address 183.1.45.4/24, Area 45
  Process ID 1, Router ID 150.1.4.4, Network Type NON_BROADCAST, Cost: 10
  <output omitted>
```

Rack1R5#sh ip ospf interface FastEthernet 0/1

```
FastEthernet0/1 is up, line protocol is up
  Internet Address 183.1.45.5/24, Area 45
  Process ID 1, Router ID 150.1.5.5, Network Type NON_BROADCAST, Cost: 10
  <output omitted>
```

Rack1R5#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
150.1.3.3	0	FULL/DROTHER	00:00:37	183.1.0.3	Serial0/0
150.1.4.4	0	FULL/DROTHER	00:00:34	183.1.0.4	Serial0/0
150.1.4.4	1	FULL/BDR	00:01:59	183.1.45.4	FastEthernet0/1

Check that VLAN6 prefix is being listed as external:

Rack1R4#show ip route ospf

```
183.1.0.0/24 is subnetted, 4 subnets
O E2    183.1.6.0 [110/20] via 183.1.46.6, 00:00:10, FastEthernet0/1
<output omitted>
```

Verify the OSPF virtual link:

Rack1R4#show ip ospf virtual-links

```
Virtual Link OSPF_VL0 to router 150.1.5.5 is up
<output omitted>
  Transit area 45, via interface FastEthernet0/0, Cost of using 10000
<output omitted>
```

Check the OSPF routes:

```
Rack1R4#show ip route ospf
<output omitted>
O      150.1.5.5/32 [110/65] via 183.1.0.5, 00:00:21, Serial0/0/0
O      150.1.3.3/32 [110/65] via 183.1.0.3, 00:00:21, Serial0/0/0
```

Verify backup configuration:

```
Rack1R4(config)#interface serial 0/0/0
Rack1R4(config-if)#shutdown
Rack1R4(config-if)#do show ip route ospf
<output omitted>
O      183.1.0.0 [110/10064] via 183.1.45.5, 00:00:23, FastEthernet0/0
<output omitted>
O      150.1.5.5/32 [110/10001] via 183.1.45.5, 00:00:23,
FastEthernet0/0
O      150.1.3.3/32 [110/10065] via 183.1.45.5, 00:00:23,
FastEthernet0/0
Rack1R4(config-if)#no shutdown
```

Verify the OSPF timers:

```
Rack1R5#show ip ospf interface S0/0 | include Timer
Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5
```

```
Rack1R4#show ip ospf interface S0/0 | include Timer
Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5
```

```
Rack1R3#show ip ospf interface S1/1 | include Timer
Timer intervals configured, Hello 333 msec, Dead 1, Wait 1,
Retransmit 5
```

Task 2.2

R3:

```
router eigrp 100
 redistribute connected metric 10000 100 255 1 1500 route-map
CONNECTED->EIGRP
!
route-map CONNECTED->EIGRP permit 10
 match interface FastEthernet0/0 FastEthernet0/1
```

R6:

```
key chain EIGRP
 key 1
  key-string CISCO
!
interface Serial0/0/0
 ip authentication mode eigrp 10 md5
 ip authentication key-chain eigrp 10 EIGRP
```

Quick Note

Arbitrary metric value. Since the task did not specify a value to be used any value could have been used.

SW4:

```
key chain RIP
  key 1
    key-string CISCO
  !
interface Vlan102
  ip rip authentication mode md5
  ip rip authentication key-chain RIP
```

R3:

```
router eigrp 100
  redistribute ospf 1 metric 10000 100 255 1 1500
  !
router ospf 1
  redistribute eigrp 100 subnets
  !
route-map CONNECTED->EIGRP permit 20
  match interface Serial1/1 Loopback0
```

R5:

```
router eigrp 100
  redistribute ospf 1 metric 10000 100 255 1 1500
  !
router ospf 1
  redistribute eigrp 100 subnets
```

R6:

```
router eigrp 10
  redistribute ospf 1 metric 10000 100 255 1 1500
  !
router ospf 1
  redistribute eigrp 10 subnets
  !
route-map CONNECTED->OSPF permit 20
  match interface serial0/0/0
```

SW4:

```
router eigrp 100
  redistribute rip metric 10000 100 255 1 1500
  !
router rip
  redistribute eigrp 100 metric 1
```

Task 2.2 Breakdown

© Strategy Tip

Take route redistribution step-by-step and verify each step as you go. Example: Redistribute between EIGRP and RIP on SW4. Verify the redistribution by having SW2 ping BB2. If redistribution isn't working properly SW2 would not be able to ping BB2.

The above redistribution presents three problems based on the current configuration. One of these problems is located on R6, and involves the redistribution of EIGRP into OSPF. In a previous OSPF section on R6 VLAN 6 was advertised into OSPF through redistribution. When this redistribution was configured a route-map was used to limit redistribution to only the VLAN 6 interface. However when EIGRP is then redistributed into OSPF on R6, connected interfaces running EIGRP will not be redistributed into OSPF. This is due to the fact that the route-map *CONNECTED->OSPF* ends in an implicit deny. Therefore either the route-map could be removed from the configuration, or it could be modified to allow the connected Serial interface to be redistributed into OSPF. The same problem occurs on R3 when redistributing into EIGRP.

Since connected redistribution is already occurring with a route-map filter, the Serial1/1 Frame Relay link in the OSPF domain will not be redistributed into EIGRP. This is because the link is treated as a connected interface first before being treated as an OSPF interface. To solve this, like on R6, either the connected to EIGRP route-map could be removed on R3, or it could be modified to include the Serial1/1 link.

Task 2.2 Verification

Check that the networks appear as EIGRP external routes:

```
Rack1R1#show ip route eigrp | include D EX
D EX 204.12.1.0/24 [170/2707456] via 183.1.123.2, 00:00:51, Serial0/0/0
D EX 183.1.39.0 [170/2707456] via 183.1.123.2, 00:02:20, Serial0/0/0
```

Check that we have BB1 as EIGRP neighbor with authentication enabled:

```
Rack1R6#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H Address Interface Hold Uptime SRTT RTO Q Seq Type
0 54.1.1.254 Se0/0/0 13 00:01:38 70 420 0 91
```

See if we actually receive authenticated packets:

```
Rack1R6#debug eigrp packets hello
<output omitted>
EIGRP: received packet with MD5 authentication, key id = 1
EIGRP: Received HELLO on Serial0/0/0 nbr 54.1.1.254
AS 10, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
```

Check if we have RIP enabled and have the key-chain attached:

```
Rack1SW4#show ip protocols | begin rip
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 14 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: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
  Vlan102              2    2          RIP
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.10.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.10.1.254    120          00:00:03
  Distance: (default is 120)
```

Check that we are receiving routing information via RIP from BB2:

```
Rack1SW4#show ip route rip
R    222.22.2.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
R    220.20.3.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
R    205.90.31.0/24 [120/7] via 192.10.1.254, 00:00:22, Vlan102
```

Check that R5 sees 150.1.1.0/24 via OSPF:

```
Rack1R5#show ip route 150.1.1.1
Routing entry for 150.1.1.0/24
  Known via "ospf 1", distance 89, metric 20, type extern 2, forward
  metric 64
  Redistributing via eigrp 100
  Advertised by eigrp 100 metric 10000 100 255 1 1500
  Last update from 183.1.0.3 on Serial0/0/0, 00:02:24 ago
  Routing Descriptor Blocks:
  * 183.1.0.3, from 150.1.3.3, 00:02:24 ago, via Serial0/0
    Route metric is 20, traffic share count is 1
```

```
Rack1R5#traceroute 150.1.1.1
```

```
Type escape sequence to abort.
Tracing the route to 150.1.1.1
```

```
1 183.1.0.3 32 msec 28 msec 32 msec
2 183.1.123.2 56 msec 56 msec 88 msec
3 183.1.123.1 32 msec * 32 msec
```

Verify full connectivity with the following TCL script:

```
tclsh
proc ping-internal {} {
  foreach i {
    150.1.1.1
    150.1.2.2
    150.1.3.3
    150.1.4.4
    150.1.5.5
    150.1.6.6
    150.1.7.7
    150.1.8.8
    150.1.10.10
    183.1.0.3
    183.1.0.4
    183.1.0.5
    183.1.123.1
    183.1.123.2
    183.1.123.3
    183.1.17.1
    183.1.17.7
    183.1.28.2
    183.1.28.8
    183.1.45.4
    183.1.45.5
    183.1.46.4
    183.1.46.6
    183.1.105.5
    183.1.105.10
    183.1.6.6
    183.1.107.7
    183.1.107.10
    192.10.1.10
    204.12.1.3
    54.1.1.6
  } { puts [ exec "ping $i" ] }
}
```

© Strategy Tip

By using procedures within TCL it allows you to re-run your ping test without having to paste the foreach loop back into the router. The procedure can be called at any time by just typing the procedure's name on the command line.

Use the following script, to check backbone IGP connectivity:

```
proc ping-external {} {  
    foreach i {  
        200.0.0.1  
        200.0.1.1  
        200.0.2.1  
        200.0.3.1  
        222.22.2.1  
        220.20.3.1  
        205.90.31.1  
    } { puts [ exec "ping $i" ] }  
}
```

Rack1R1#**tclsh**

```
Rack1R1(tcl)#proc ping-internal {} {  
+> foreach i {  
+> 150.1.1.1  
+> 150.1.2.2  
+> 150.1.3.3  
+> 150.1.4.4  
+> 150.1.5.5  
+> 150.1.6.6  
+> 150.1.7.7  
+> 150.1.8.8  
+> 150.1.10.10  
+> 183.1.0.3  
+> 183.1.0.4  
+> 183.1.0.5  
+> 183.1.123.1  
+> 183.1.123.2  
+> 183.1.123.3  
+> 183.1.17.1  
+> 183.1.17.7  
+> 183.1.28.2  
+> 183.1.28.8  
+> 183.1.39.3  
+> 183.1.39.9  
+> 183.1.45.4  
+> 183.1.45.5  
+> 183.1.46.4  
+> 183.1.46.6  
+> 183.1.105.5  
+> 183.1.105.10  
+> 183.1.6.6  
+> 183.1.107.7  
+> 183.1.107.10  
+> 192.10.1.10  
+> 204.12.1.3  
+> 54.1.1.6  
+> } { puts [ exec "ping $i" ] }  
+>}
```

Rack1R1(tcl)#**ping-internal**

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

<output omitted>

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 54.1.1.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 140/143/145
ms
```

```
Rack1R1(tcl)#ping-external
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/205/208
ms
```

<output omitted>

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 205.90.31.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/205/212
ms
```

```
Rack1R1(tcl)#tclquit
Rack1R1#
```

Pitfall

Remember to exit the TCL shell using the `tclquit` command when finished with the reachability verification. If the TCL shell is enabled commands that overlap between TCL and the IOS will be interpreted by TCL and not the IOS. An example is the `set` command used in a route-map. Both TCL and the IOS use the `set` command. If you try to use the `set` command in a route-map when the TCL shell is still enabled the TCL shell will display an error message:

```
Rack1R1(tcl)#conf t
Rack1R1(config)#route-map TEST
Rack1R1(config-route-map)#set ip next-hop 1.1.1.1
wrong # args: should be "set varName ?newValue?"
Rack1R1(config-route-map)#do tclquit
Rack1R1(config-route-map)#set ip next-hop 1.1.1.1
Rack1R1(config-route-map)#
```

 **Note**

Older Catalyst IOS versions do not support TCL scripting. A smartport macro can be used in place of the TCL shell for ping tests on the switches as follows.

Rack1SW3(config)#macro name PINGS

Enter macro commands one per line. End with the character '@'.

do ping 150.1.1.1

do ping 150.1.2.2

<output omitted>

@

Rack1SW3(config)#macro global apply PINGS

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 150.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 112/113/116 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 150.1.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 150.1.3.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

<output omitted>

Task 2.3

R6:

```
ip as-path access-list 1 permit _54$
```

```
!
```

```
route-map LOCAL_PREFERENCE permit 10
```

```
  match as-path 1
```

```
  set local-preference 200
```

```
!
```

```
route-map LOCAL_PREFERENCE permit 1000
```

```
!
```

```
router bgp 100
```

```
  neighbor 54.1.1.254 route-map LOCAL_PREFERENCE in
```

R1:

```
interface Loopback1
```

```
  ip address 150.1.11.1 255.255.255.0
```

```
!
```

```
router bgp 200
```

```
  network 150.1.11.0 mask 255.255.255.0
```

R2:

```
ip prefix-list R1_BGP_LOOPBACK seq 5 permit 150.1.11.0/24
```

```
!
```

```
route-map MED permit 10
```

```

match ip address prefix-list R1_BGP_LOOPBACK
set metric 200
!
route-map MED permit 1000
!
router bgp 200
neighbor 183.1.123.3 route-map MED out

```

R5:

```

router bgp 100
neighbor 183.1.0.3 next-hop-self

```

SW4:

```

ip prefix-list R1_BGP_LOOPBACK seq 5 permit 150.1.11.0/24
!
route-map MED permit 10
match ip address prefix-list R1_BGP_LOOPBACK
set metric 100
!
route-map MED permit 1000
!
router bgp 200
neighbor 183.1.105.5 route-map MED out

```

Task 2.3 Verification

Verify that local preference is correctly set to 200 for routes originating from AS 54:

```
Rack1R6#show ip bgp regexp _54$
```

```
<output omitted>
```

```

*> 28.119.16.0/24    54.1.1.254          200      0 54 i
*> 28.119.17.0/24    54.1.1.254          200      0 54 i
*> 114.0.0.0         54.1.1.254          0        200     0 54 i
*> 115.0.0.0         54.1.1.254          0        200     0 54 i
*> 116.0.0.0         54.1.1.254          0        200     0 54 i
*> 117.0.0.0         54.1.1.254          0        200     0 54 i
*> 118.0.0.0         54.1.1.254          0        200     0 54 i
*> 119.0.0.0         54.1.1.254          0        200     0 54 i

```

And that the other AS paths have a local preference of 100:

```
Rack1R6#show ip bgp regexp _254$
```

```
<output omitted>
```

```

*>i205.90.31.0      183.1.105.10        0        100     0 200 254 ?
*>i220.20.3.0       183.1.105.10        0        100     0 200 254 ?
*>i222.22.2.0       183.1.105.10        0        100     0 200 254 ?

```

Confirm that R3 has two paths to 150.1.11.0/24:

```
Rack1R3#show ip bgp 150.1.11.0
```

```
BGP routing table entry for 150.1.11.0/24, version 44
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
    1
  200
    183.1.123.1 from 183.1.123.2 (150.1.2.2)
      Origin IGP, metric 200, localpref 100, valid, external
  200
    183.1.0.5 from 183.1.0.5 (150.1.5.5)
      Origin IGP, metric 100, localpref 100, valid, internal, best
```

```
Rack1R5#show ip bgp 150.1.11.0
```

```
BGP routing table entry for 150.1.11.0/24, version 38
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
    2          3
  200
    183.1.105.10 from 183.1.105.10 (150.1.10.10)
      Origin IGP, metric 100, localpref 100, valid, external, best
```

Verify that backup works:

```
Rack1R5#conf t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Rack1R5(config)#interface FastEthernet 0/0
```

```
Rack1R5(config-if)#shut
```

```
Rack1R5#show ip bgp 150.1.11.0
```

```
BGP routing table entry for 150.1.11.0/24, version 29
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
    3
  200, (Received from a RR-client)
    183.1.123.1 (metric 20) from 183.1.0.3 (150.1.3.3)
      Origin IGP, metric 200, localpref 100, valid, internal, best
```

```
Rack1R5#traceroute 150.1.11.1
```

```
Type escape sequence to abort.
```

```
Tracing the route to 150.1.11.1
```

```
 1 183.1.0.3 28 msec 32 msec 32 msec
 2 183.1.123.2 44 msec 48 msec 44 msec
 3 183.1.123.1 52 msec * 48 msec
```

Task 3.1

```
R4:
```

```
ipv6 unicast-routing
```

```
!
```

```
interface FastEthernet0/1
```

```
ipv6 address 2001:CC1E:1:404::/64 eui-64
```

R5:

```
ipv6 unicast-routing
!
interface FastEthernet0/0
  ipv6 address 2001:CC1E:1:505::/64 eui-64
```

R4:

```
interface Tunnel45
  ipv6 address 2001:CC1E:1:4545::4/64
  tunnel source 150.1.4.4
  tunnel destination 150.1.5.5
  tunnel mode ipv6ip
```

R5:

```
interface Tunnel45
  ipv6 address 2001:CC1E:1:4545::5/64
  tunnel source 150.1.5.5
  tunnel destination 150.1.4.4
  tunnel mode ipv6ip
```

R4:

```
ipv6 router eigrp 45
  no shutdown
!
interface Tunnel45
  ipv6 eigrp 45
!
interface FastEthernet0/1
  ipv6 eigrp 45
```

R5:

```
ipv6 router eigrp 45
  no shutdown
!
interface Tunnel45
  ipv6 eigrp 45
!
interface FastEthernet0/0
  ipv6 eigrp 45
```

Task 3.1 Verification

Verify IPv6 addressing:

```
Rack1R5#show ipv6 interface brief
FastEthernet0/0          [up/up]
  FE80::207:EBFF:FEDE:5621
  2001:CC1E:1:505:207:EBFF:FEDE:5621
```

```
Rack1R4#show ipv6 interface brief
FastEthernet0/1          [up/up]
  FE80::230:94FF:FE7E:E582
  2001:CC1E:1:404:250:80FF:FE04:8E01
```

Verify the tunnel:

```
Rack1R5#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
<output omitted>
  Tunnel source 150.1.5.5, destination 150.1.4.4
  Tunnel protocol/transport IPv6/IP
```

```
Rack1R5#ping 2001:CC1E:1:4545::4
```

```
Sending 5, 100-byte ICMP Echos to 2001:CC1E:1:4545::4, timeout is 2
seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/71/76 ms
```

Task 3.1 Verification

Verify IPv6 addressing:

```
Rack1R5#show ipv6 interface brief fastEthernet 0/0
FastEthernet0/0 [up/up]
FE80::C004:CFF:FE7A:0
2001:CC1E:1:505:C004:CFF:FE7A:0
```

```
Rack1R4#show ipv6 interface brief fastEthernet 0/1
FastEthernet0/1 [up/up]
FE80::C003:CFF:FE7A:1
2001:CC1E:1:404:C003:CFF:FE7A:1
```

Verify the tunnel:

```
Rack1R5#show interfaces tunnel 45
Tunnel45 is up, line protocol is up
<output omitted>
  Tunnel source 150.1.5.5, destination 150.1.4.4
  Tunnel protocol/transport IPv6/IP
```

```
Rack1R5#ping 2001:CC1E:1:404:C003:CFF:FE7A:1
```

```
Sending 5, 100-byte ICMP Echos to 2001:CC1E:1:404:C003:CFF:FE7A:1,
timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/71/76 ms
```

Check to see that EIGRPv6 is configured correctly:

```
Rack1R5#show ipv6 eigrp neighbors
IPv6-EIGRP neighbors for process 45
```

```
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num
0 Link-local address: Tu45 11 00:02:55 122 5000 0 3
FE80::9601:404
```

Rack1R4#show ipv6 eigrp neighbors

```
IPv6-EIGRP neighbors for process 45
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num
0 Link-local address: Tu45 12 00:05:48 45 5000 0 4
FE80::9601:505
```

Reachability Verification

Rack1R4#show ipv6 route eigrp

```
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route, M - MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D - EIGRP, EX - EIGRP external
D 2001:CC1E:1:505::/64 [90/297246976]
via FE80::9601:505, Tunnel45
```

Rack1R5#show ipv6 route eigrp

```
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route, M - MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D - EIGRP, EX - EIGRP external
D 2001:CC1E:1:404::/64 [90/297246976]
via FE80::9601:404, Tunnel45
```

Task 3.2

R5:

```
!
! The below is the IPv6 EUI-64 address of R5's interface
! It will be different depending on the interface MAC address
!
ipv6 multicast-routing
!
ipv6 pim bsr candidate rp 2001:CC1E:1:505:21B:D4FF:FEFE:6298
```

R4:

```
!
! The below is the IPv6 EUI-64 address of R4's interface
! It will be different depending on the interface MAC address
!
ipv6 multicast-routing
```

```

ipv6 pim bsr candidate bsr 2001:CC1E:1:404:21B:D4FF:FE47:37B1 priority
100
!
ipv6 access-list MLD_FILTER
 permit ipv6 any host FF06::6
!
interface FastEthernet 0/0
 ipv6 mld access-group MLD_FILTER

```

R6:

```

interface FastEthernet 0/0
 ipv6 enable
 ipv6 mld join-group ff06::6

```

Task 3.2 Verification

Check the RP and mroute:

Rack1R4#show ipv6 pim bsr rp-cache

```
PIMv2 BSR C-RP Cache
```

```
BSR Candidate RP Cache
```

```

Group(s) FF00::/8, RP count 1
  RP 2001:CC1E:1:505:21B:D4FF:FEFE:6298 SM
    Priority 100, Holdtime 150
    Uptime: 00:12:31, expires: 00:02:01

```

Rack1R4#show ipv6 mroute

```
Multicast Routing Table
```

```

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host
Report,

```

```

       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT

```

```
Timers: Uptime/Expires
```

```
Interface state: Interface, State
```

```
(*, FF06::6), 00:44:57/never, RP 2001:CC1E:1:505:21B:D4FF:FEFE:6298,
flags: SCJ
```

```
Incoming interface: Tunnel45
```

```
RPF nbr: FE80::9601:505
```

```
Immediate Outgoing interface list:
```

```
FastEthernet0/1, Forward, 00:44:57/never
```

Simulate multicast traffic and make sure it's being received:

Rack1R6#debug ipv6 icmp

```
ICMP Packet debugging is on
```

Rack1R5#ping ff06::6 repeat 100

```
Output Interface: Tunnel45
```

```
Type escape sequence to abort.
```

Sending 100, 100-byte ICMP Echos to FF06::6, timeout is 2 seconds:
Packet sent with a source address of 2001:CC1E:1:4545::5

Rack1R6#

ICMPv6: Sent N-Solicit, Src=FE80::21B:D4FF:FE70:3D8, Dst=FF02::1:FF00:5

ICMPv6: Received echo request, Src=2001:CC1E:1:4545::5, Dst=FF06::6

ICMPv6: Sent echo reply, Src=FE80::21B:D4FF:FE70:3D8,

Dst=2001:CC1E:1:4545::5

ICMPv6: Sent N-Solicit, Src=FE80::21B:D4FF:FE70:3D8, Dst=FF02::1:FF00:5

Rack1R6#

ICMPv6: Received echo request, Src=2001:CC1E:1:4545::5, Dst=FF06::6

ICMPv6: Sent echo reply, Src=FE80::21B:D4FF:FE70:3D8,

Dst=2001:CC1E:1:4545::5

ICMPv6: Sent N-Solicit, Src=FE80::21B:D4FF:FE70:3D8, Dst=FF02::1:FF00:5

Task 4.1

R4:

```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1

interface FastEthernet 0/0
 mpls ip
!
interface FastEthernet 0/1
 mpls ip
!
interface Serial 0/0/0
 mpls ip
```

R5:

```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1

interface FastEthernet 0/0
 mpls ip
!
interface Serial 0/0/0
 mpls ip
```

R6:

```
mpls ip
access-list 1 permit 150.1.0.0 0.0.255.255
!
no mpls ldp advertise-labels
mpls ldp advertise-labels for 1
!
interface FastEthernet 0/0
 mpls ipp
```

Task 4.1 Verification

Check MPLS LDP neighbors on R4, since it peers with both R5 and R6:

Rack1R4#show mpls ldp neighbor

```
Peer LDP Ident: 150.1.6.6:0; Local LDP Ident 150.1.4.4:0
  TCP connection: 150.1.6.6.52236 - 150.1.4.4.646
  State: Oper; Msgs sent/rcvd: 124/127; Downstream
  Up time: 01:34:08
  LDP discovery sources:
    FastEthernet0/1, Src IP addr: 183.1.46.6
  Addresses bound to peer LDP Ident:
    183.1.46.6      183.1.6.6      54.1.1.6      150.1.6.6
Peer LDP Ident: 150.1.5.5:0; Local LDP Ident 150.1.4.4:0
  TCP connection: 150.1.5.5.48660 - 150.1.4.4.646
  State: Oper; Msgs sent/rcvd: 136/157; Downstream
  Up time: 01:34:06
  LDP discovery sources:
    Serial0/0/0, Src IP addr: 183.1.0.5
  Addresses bound to peer LDP Ident:
    183.1.105.5    183.1.45.5     183.1.0.5     150.1.5.5
```

Now check that labels were only generated for Loopback0 interfaces. Notice that all prefixes with except to the 150.1.0.0/24 range don't have labels assigned.

Rack1R4#show mpls forwarding-table

Local Hop	Outgoing	Prefix	Bytes	Label	Outgoing	Next
16	No Label	150.1.3.3/32	0		Se0/0/0	
17	Pop Label	150.1.5.5/32	11949		Se0/0/0	
18	Pop Label	150.1.6.6/32	13789		Fa0/1	

Rack1R5#show mpls forwarding-table

Local Hop	Outgoing	Prefix	Bytes	Label	Outgoing	Next
16	No Label	150.1.1.0/24	0		Fa0/0	
17	No Label	150.1.2.0/24	0		Fa0/0	
18	No Label	150.1.3.3/32	0		Se0/0/0	
19	Pop Label	150.1.4.4/32	0		Se0/0/0	
20	18	150.1.6.6/32	0		Se0/0/0	
21	No Label	150.1.7.0/24	0		Fa0/0	

```

22      No Label      150.1.8.0/24      0      Fa0/0
183.1.105.10
23      No Label      150.1.10.0/24     0      Fa0/0
183.1.105.10
24      No Label      183.1.17.0/24     0      Fa0/0
183.1.105.10
25      No Label      183.1.28.0/24     0      Fa0/0
183.1.105.10
26      No Label      183.1.46.0/24     0      Se0/0/0
183.1.0.4
27      No Label      183.1.107.0/24    0      Fa0/0
183.1.105.10
28      No Label      183.1.123.0/24    0      Fa0/0
183.1.105.10

```

Rack1R6#show mpls forwarding-table

Local Hop	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next
16	16	150.1.3.3/32	0		Fa0/0	
17	Pop Label	150.1.4.4/32	0		Fa0/0	
18	17	150.1.5.5/32	0		Fa0/0	
19	No Label	183.1.0.0/24	0		Fa0/0	
20	No Label	183.1.45.0/24	0		Fa0/0	

Task 4.2

R5:

```

ip vrf VPN_A
  rd 100:5
  route-target export 100:5
  route-target import 100:6
!
interface Loopback 1
  ip vrf forwarding VPN_A
  ip address 172.16.5.5 255.255.255.0
!
router bgp 100
  address-family vpnv4
  neighbor 150.1.6.6 activate
  neighbor 150.1.6.6 send-community extended
  address-family ipv4 unicast vrf VPN_A
  redistribute connected

```

R6:

```

ip vrf VPN_B
  rd 100:6
  route-target export 100:6
  route-target import 100:5
!
interface Loopback 1

```

```

ip vrf forwarding VPN_B
ip address 192.168.6.6 255.255.255.0
!
router bgp 100
 address-family vpnv4
  neighbor 150.1.5.5 activate
  neighbor 150.1.5.5 send-community extended
  address-family ipv4 unicast vrf VPN_B
  redistribute connected

```

Task 4.2 Verification

Check BGP peering session:

```

Rack1R6#show bgp vpnv4 unicast all summary
BGP router identifier 150.1.6.6, local AS number 100
BGP table version is 5, main routing table version 5
3 network entries using 468 bytes of memory
3 path entries using 204 bytes of memory
6/2 BGP path/bestpath attribute entries using 1008 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP extended community entries using 48 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
Bitfield cache entries: current 1 (at peak 1) using 32 bytes of memory
BGP using 1784 total bytes of memory
BGP activity 14/0 prefixes, 20/6 paths, scan interval 15 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
Up/Down	State/PfxRcd						
150.1.5.5	4	100	153	149	5	0	0
00:07:25	1						

Check that prefixes have been exchanged over BGP:

```

Rack1R6#show bgp vpnv4 unicast all
BGP table version is 5, local router ID is 150.1.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 100:5
*>i172.16.5.0/24    150.1.5.5          0      100      0 ?
Route Distinguisher: 100:6 (default for vrf VPN_B)
*>i172.16.5.0/24    150.1.5.5          0      100      0 ?
*> 192.168.6.0      0.0.0.0            0              32768 ?

```

```

Rack1R5#show bgp vpnv4 unicast all summary
BGP router identifier 150.1.5.5, local AS number 100
BGP table version is 5, main routing table version 5
3 network entries using 468 bytes of memory

```

```

3 path entries using 204 bytes of memory
8/2 BGP path/bestpath attribute entries using 1344 bytes of memory
3 BGP AS-PATH entries using 72 bytes of memory
2 BGP extended community entries using 48 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
Bitfield cache entries: current 3 (at peak 3) using 96 bytes of memory
BGP using 2232 total bytes of memory
BGP activity 24/0 prefixes, 24/0 paths, scan interval 15 secs

```

```

Neighbor          V          AS MsgRcvd MsgSent   TblVer  InQ  OutQ
Up/Down  State/PfxRcd
150.1.6.6      4          100     150     155         5    0    0
00:08:47      1

```

```
Rack1R5#show bgp vpnv4 unicast all
```

```

BGP table version is 5, local router ID is 150.1.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

```

```

r RIB-failure, S Stale

```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```

Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 100:5 (default for vrf VPN_A)
*> 172.16.5.0/24  0.0.0.0           0         32768 ?
*>i192.168.6.0    150.1.6.6         0        100     0 ?
Route Distinguisher: 100:6
*>i192.168.6.0    150.1.6.6         0        100     0 ?

```

Check the label stacks for VPN prefixes in R5 and R6:

```

Rack1R6#show ip cef vrf VPN_B 172.16.5.5
172.16.5.0/24
  nexthop 183.1.46.4 FastEthernet0/0 label 17 29

```

```

Rack1R5#show ip cef vrf VPN_A 192.168.6.6
192.168.6.0/24
  nexthop 183.1.0.4 Serial0/0/0 label 18 21

```

Do a ping and a traceroute to VPN prefixes:

```
Rack1R5#ping vrf VPN_A 192.168.6.6 source loopback 1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.6.6, timeout is 2 seconds:
Packet sent with a source address of 172.16.5.5
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/64 ms
```

```
Rack1R5#traceroute vrf VPN_A 192.168.6.6 source loopback 1
```

```
Type escape sequence to abort.
```

```
Tracing the route to 192.168.6.6
```

```
1 183.1.0.4 [MPLS: Labels 18/21 Exp 0] 64 msec 60 msec 60 msec
2 192.168.6.6 32 msec * 28 msec
```

Task 5.1

R2:

```
interface Loopback0
 ip pim sparse-mode
!
ip pim send-rp-discovery Loopback0 scope 16
```

R3:

```
interface Loopback0
 ip pim sparse-mode
!
ip pim send-rp-announce Loopback0 scope 16
```

Task 5.1 Verification

Verify that RP mapping information has been disseminated to routers:

Rack1R2#show ip pim rp mapping

```
PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)
```

```
Group(s) 224.0.0.0/4
 RP 150.1.3.3 (?), v2v1
   Info source: 150.1.3.3 (?), elected via Auto-RP
   Uptime: 00:03:26, expires: 00:02:31
```

Rack1R3#show ip pim rp mapping

```
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
```

```
Group(s) 224.0.0.0/4
 RP 150.1.3.3 (?), v2v1
   Info source: 150.1.2.2 (?), elected via Auto-RP
   Uptime: 00:04:03, expires: 00:02:53
```

Rack1R5#show ip pim rp mapping

```
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
 RP 150.1.3.3 (?), v2v1
   Info source: 150.1.2.2 (?), elected via Auto-RP
   Uptime: 00:04:32, expires: 00:02:26
```

Task 5.2

R5:

```
interface FastEthernet0/0
 ip igmp join-group 226.26.26.26
!
```

```
ip mroute 0.0.0.0 0.0.0.0 183.1.0.3
```

Task 5.2 Verification

Before the static mroute is configured on R5:

```
Rack1R2#ping
Protocol [ip]:
Target IP address: 226.26.26.26
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: Serial0/0
Time to live [255]:
Source address: 183.1.28.2
...
Rack1R5# debug ip mpacket
IP(0): s=183.1.28.2 (Serial0/0/0) d=226.26.26.26 id=165, ttl=254,
prot=1, len=104(100), not RPF interface
IP(0): s=183.1.28.2 (Serial0/0/0) d=226.26.26.26 id=166, ttl=254,
prot=1, len=104(100), not RPF interface

Rack1R5#sh ip mroute
<output omitted>

(183.1.28.2, 226.26.26.26), 00:00:15/00:02:44, flags: L
  Incoming interface: FastEthernet0/0, RPF nbr 183.1.105.10
  Outgoing interface list:
    Serial0/0/0, Forward/Sparse-Dense, 00:00:16/00:00:00
```

After the static mroute is configured:

```
Rack1R2#ping
Protocol [ip]:
Target IP address: 226.26.26.26
Repeat count [1]: 100
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Interface [All]: Serial0/0
Time to live [255]:
Source address: 183.1.28.2

Reply to request 0 from 183.1.0.5, 64 ms
Reply to request 0 from 183.1.0.5, 192 ms
Reply to request 1 from 183.1.0.5, 60 ms
Reply to request 1 from 183.1.0.5, 188 ms

Rack1R5#sh ip mroute
<output omitted>

(183.1.28.2, 226.26.26.26), 00:00:15/00:02:59, flags: LJT
```

```
Incoming interface: Serial0/0/0, RPF nbr 183.1.0.3, Mroute
Outgoing interface list:
  FastEthernet0/0, Forward/Sparse-Dense, 00:00:16/00:02:54
```

Task 5.3

```
R3:
access-list 1 deny 239.0.0.0 0.255.255.255
access-list 1 permit any
!
interface FastEthernet0/0
 ip igmp access-group 1
```

Task 5.3 Verification

```
Rack1R3#show ip igmp interface FastEthernet 0/0 | include access
Inbound IGMP access group is 1
```

```
Rack1R3#show ip access-lists 1
Standard IP access list 1
 10 deny 239.0.0.0, wildcard bits 0.255.255.255
 20 permit any (1 match)
```

Task 6.1

```
R3:
ip access-list extended SYN_ATTACK
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
!
interface FastEthernet0/0
 ip access-group SYN_ATTACK in
```

```
SW4:
ip access-list extended SYN_ATTACK
 permit tcp any host 183.1.28.100 eq www syn log-input
 permit ip any any
!
interface Vlan102
 ip access-group SYN_ATTACK in
```

Task 6.1 Verification

Generate TCP SYN packets from BB2 and watch the ACL log hits on SW2:

```
BB2>telnet 183.1.28.100 80
Trying 183.1.28.100, 80 ...
```

```
Rack1SW2#show logging
```

```
<output omitted>
%SEC-6-IPACCESSLOGP: list SYN_ATTACK permitted tcp 192.10.1.254(18518) (Vlan102
0010.7b3a.14cc) -> 183.1.28.100(80), 1 packet
```

Task 6.2

R3:

```
ip access-list extended SYN_ATTACK
deny ip 183.1.0.0 0.0.255.255 any
permit tcp any host 183.1.28.100 eq www syn log-input
permit ip any any
```

SW4:

```
ip access-list extended SYN_ATTACK
deny ip 183.1.0.0 0.0.255.255 any
permit tcp any host 183.1.28.100 eq www syn log-input
permit ip any any
```

R6:

```
ip access-list extended SYN_ATTACK
deny ip 183.1.0.0 0.0.255.255 any
permit ip any any
!
interface Serial0/0/0
ip access-group SYN_ATTACK in
```

Task 6.2 Verification

```
Rack1R3#sh ip access-lists | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
 10 deny ip 183.1.0.0 0.0.255.255 any
 20 permit tcp any host 183.1.28.100 eq www syn log-input
 30 permit ip any any (3 matches)
```

```
Rack1R6#sh ip access-lists | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
 10 deny ip 183.1.0.0 0.0.255.255 any
 20 permit ip any any (20 matches)
```

```
Rack1SW2#sh ip access-lists | beg SYN_ATTACK
Extended IP access list SYN_ATTACK
 10 deny ip 183.1.0.0 0.0.255.255 any
 20 permit tcp any host 183.1.28.100 eq www syn log-input
 30 permit ip any any (19 matches)
```

Task 6.3

SW4:

```
interface Vlan102
no ip unreachable
no ip mask-reply
```

Task 6.3 Verification

```
Rack1SW4#show ip interface vlan 102
Vlan102 is up, line protocol is up
<snip>
  ICMP unreachable are never sent
  ICMP mask replies are never sent
<snip>
```

Task 6.4

R4:

```
ip access-list extended TTL
  deny ip any any ttl lt 3 log
  permit ip any any
!
interface FastEthernet0/0
  ip access-group TTL out
!
interface FastEthernet0/1
  ip access-group TTL out
!
interface Serial0/0/0
  ip access-group TTL out

logging on
logging console informational
logging buffered informational
no logging monitor
```

Task 6.4 Verification

Send out traceroute packets with different TTL value. The first one using TTL up from 4:

```
Rack1R6#traceroute 150.1.1.1 ttl 4 10
```

Type escape sequence to abort.
Tracing the route to 150.1.1.1

```
 4 183.1.107.7 28 msec 32 msec 32 msec
 5 183.1.17.1 28 msec * 28 msec
```

The second one use TTL starting up from one

```
Rack1R6#traceroute 150.1.1.1 ttl 1 10
```

Type escape sequence to abort.
Tracing the route to 150.1.1.1

```
 1 183.1.46.4 0 msec 4 msec 0 msec
 2 183.1.46.4 !A * !A
```

Rack1R4#show logging

```
Syslog logging: enabled (0 messages dropped, 3 messages rate-limited,
0 flushes, 0 overruns, xml disabled, filtering
disabled)
```

```
<snip>
```

```
Log Buffer (4096 bytes):
```

```
%SEC-6-IPACCESSLOGP: list TTL denied udp 183.1.46.6(0) -> 150.1.5.5(0),
1 packet
```

```
%SEC-6-IPACCESSLOGP: list TTL denied udp 183.1.46.6(0) -> 150.1.1.1(0),  
1 packe
```

Task 7.1

R2:

```
rmon alarm 1 ifEntry.11.1 60 delta rising-threshold 15000 1 falling-  
threshold 5000 2  
rmon event 1 trap IETRAP description "Above 15000 for ifInUcastPkts"  
rmon event 2 trap IETRAP description "Below 5000 for ifInUcastPkts"  
snmp-server host 183.17.1.100 IETRAP
```

Task 7.1 Verification

Verify RMON configuration:

Rack1R2#show rmon alarms

```
Alarm 1 is active, owned by config  
Monitors ifEntry.11.1 every 60 second(s)  
Taking delta samples, last value was 0  
Rising threshold is 15000, assigned to event 1  
Falling threshold is 5000, assigned to event 2  
On startup enable rising or falling alarm
```

Rack1R2#show rmon events

```
Event 1 is active, owned by config  
Description is Above 15000 for ifInUcastPkts  
Event firing causes trap to community IETRAP,  
last event fired at 0y0w0d,00:00:00,  
Current uptime 0y0w0d,06:11:00  
Event 2 is active, owned by config  
Description is Below 5000 for ifInUcastPkts  
Event firing causes trap to community IETRAP,  
last event fired at 0y0w0d,00:00:00,  
Current uptime 0y0w0d,06:11:00
```

Task 7.2

R3:

```
ntp server 204.12.1.254  
ntp peer 150.1.6.6
```

R6:

```
ntp server 54.1.1.254  
ntp server 150.1.3.3
```

R1, R2, and SW1:

```
ntp server 150.1.3.3
```

R4, R5, and SW4:

```
ntp server 150.1.6.6
```

Task 7.2 Verification

Verify NTP status and associations:

Rack1R3#**show ntp status**

Clock is synchronized, stratum 5, reference is 204.12.1.254
<output omitted>

Rack1R3#**show ntp associations**

address	ref clock	st	when	poll	reach	delay	offset	disp
+~150.1.6.6	54.1.1.254	5	61	64	6	92.7	50583.	15875.
*~204.12.1.254	127.127.7.1	4	35	64	377	7.5	-1.70	0.7

* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured

Rack1R3#**show ntp associations detail**

150.1.6.6 configured, selected, sane, valid, stratum 5
ref ID 54.1.1.254, time AF67AB02.8F6D2C86 (06:19:46.560 UTC Sat Apr 3 1993)
our mode active, peer mode passive, our poll intvl 64, peer poll intvl 64
<output omitted>

204.12.1.254 configured, our_master, sane, valid, stratum 4
ref ID 127.127.7.1, time AF67AAB6.27A770F0 (06:18:30.154 UTC Sat Apr 3 1993)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
<output omitted>

Rack1SW1#**show ntp status**

Clock is synchronized, stratum 6, reference is 150.1.3.3
<output omitted>

Rack1SW1#**show ntp associations**

address	ref clock	st	when	poll	reach	delay	offset	disp
*~150.1.3.3	204.12.1.254	5	50	64	340	38.1	0.75	16000.

* master (syncd), # master (unsyncd), + selected, - candidate, ~ configured

Task 7.3

R3:

ntp authentication-key 1 md5 CISCO

R6:

ntp authentication-key 1 md5 CISCO

R1, R2, and SW1:

ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 150.1.3.3 key 1

R4, R5, and SW4:

```
ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 150.1.6.6 key 1
```

Task 7.3 Verification**Rack1R1#show ntp associations detail**

```
150.1.3.3 configured, authenticated, our_master, sane, valid, stratum 6
ref ID 204.12.1.254, time CCEC61CE.6070F38F (04:06:38.376 UTC Fri Dec 12 2008)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 47.26 msec, root disp 11.40, reach 377, sync dist 74.097
delay 70.27 msec, offset 0.8069 msec, dispersion 3.94
precision 2**18, version 3
org time CCEC6203.7CB0702A (04:07:31.487 UTC Fri Dec 12 2008)
rcv time CCEC6203.8729CA DF (04:07:31.527 UTC Fri Dec 12 2008)
xmt time CCEC6203.715D99BE (04:07:31.442 UTC Fri Dec 12 2008)
filtdelay =    84.85    84.67    84.37    84.37    70.27    69.08    69.27    69.96
filtoffset =     1.52     0.88    -0.17    -0.67     0.81     0.86     0.21     0.04
filtererror =     0.02     0.99     1.97     2.62     3.60     4.58     5.55     5.57
```

Rack1R4#show ntp associations detail

```
150.1.6.6 configured, authenticated, our_master, sane, valid, stratum 5
ref ID 54.1.1.254, time CCEC6217.A1919786 (04:07:51.631 UTC Fri Dec 12 2008)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 29.75 msec, root disp 2.81, reach 377, sync dist 19.302
delay 3.05 msec, offset -1.2642 msec, dispersion 0.09
precision 2**18, version 3
org time CCEC621B.BE0A1D73 (04:07:55.742 UTC Fri Dec 12 2008)
rcv time CCEC621B.BEC170E5 (04:07:55.745 UTC Fri Dec 12 2008)
xmt time CCEC621B.BDE7063D (04:07:55.741 UTC Fri Dec 12 2008)
filtdelay =     3.05     3.08     3.10     3.45     3.17     3.14     3.13     3.23
filtoffset =    -1.26    -1.26    -1.28    -1.03    -0.99    -0.75    -0.23    -0.19
filtererror =     0.02     0.99     1.97     2.61     3.59     4.56     5.54     5.55
```

Task 7.4**R2:**

```
interface Serial0/0
 ip accounting precedence input
 ip accounting precedence output
!
ip accounting-threshold 50000
```

R3:

```
interface Serial1/0
 ip accounting precedence input
 ip accounting precedence output
!
ip accounting-threshold 50000
```

Task 7.4 Verification

Verify precedence accounting:

```
Rack1R2#show interfaces serial 0/0 precedence
```

```
Serial0/0
  Input
    Precedence 6: 114 packets, 8737 bytes
  Output
    Precedence 0: 1 packets, 114 bytes
    Precedence 6: 119 packets, 8051 bytes
```

Rack1R3#show interfaces serial 1/0 prec

```
Serial1/0
  Input
    Precedence 6: 35 packets, 2706 bytes
  Output
    Precedence 0: 1 packets, 114 bytes
    Precedence 6: 98 packets, 6966 bytes
```

Task 7.5

R5:

```
interface FastEthernet0/0
  standby 1 ip 183.1.105.254
  standby 1 preempt
  standby 1 track Serial0/0/0 100
```

SW4:

```
interface FastEthernet0/18
  standby 1 ip 183.1.105.254
  standby 1 priority 50
  standby 1 preempt
```

Task 7.5 Verification

Verify HSRP configuration:

Rack1R5#show standby

```
FastEthernet0/0 - Group 1
  State is Active
    2 state changes, last state change 00:01:16
  Virtual IP address is 183.1.105.254
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.896 secs
  Preemption enabled
  Active router is local
  Standby router is 183.1.105.10, priority 50 (expires in 7.892 sec)
  Priority 100 (default 100)
    Track interface Serial0/0/0 state Up decrement 100
```

Rack1R5(config)#interface Serial 0/0/0

Rack1R5(config-if)#shutdown

<output omitted>

```
%HSRP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Speak
```

Rack1R5(config-if)#do show standby

```
FastEthernet0/0 - Group 1
```

```

State is Standby
<output omitted>
Active router is 183.1.105.10, priority 50 (expires in 8.200 sec)
Standby router is local
Priority 0 (default 100)
  Track interface Serial0/0/0 state Down decrement 100
IP redundancy name is "hsrp-Fa0/0-1" (default)

```

Task 7.6

R3:

```

access-list 2 permit 183.1.0.0 0.0.255.255
!
ip nat inside source list 2 interface FastEthernet0/0 overload
!
interface FastEthernet0/0
  ip nat outside
!
interface Serial1/0
  ip nat inside
!
interface Serial1/1
  ip nat inside

```

Task 7.6 Verification

Verify the NAT translations:

Rack1R1#ping 204.12.1.254

Type escape sequence to abort.
 Sending 5, 100-byte ICMP Echos to 204.12.1.254, timeout is 2 seconds:
 !!!!!

Rack1R3#sh ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
icmp	204.12.1.3:3179	183.1.123.1:3179	204.12.1.254:3179	204.12.1.254:3179
icmp	204.12.1.3:3180	183.1.123.1:3180	204.12.1.254:3180	204.12.1.254:3180
icmp	204.12.1.3:3181	183.1.123.1:3181	204.12.1.254:3181	204.12.1.254:3181
icmp	204.12.1.3:3182	183.1.123.1:3182	204.12.1.254:3182	204.12.1.254:3182
icmp	204.12.1.3:3183	183.1.123.1:3183	204.12.1.254:3183	204.12.1.254:3183

Task 7.7

R6:

```

!
! Create an interface event detector based applet.
! notice that the command "event interface..." should be entered
! along on a single line - it defines a detector condition
!
event manager applet INTERFACE_MONITOR
  event interface name Serial 0/0/0 parameter txload entry-op gt entry-
va 204 entry-type value poll-interval 30
  action 1.0 syslog msg "R6's $_interface_name output rate:
$_interface_parameter = $_interface_value"

```

```
!
interface Serial 0/0/0
 load-interval 30
```

Task 7.7 Verification

Generate a flood of ICMP packets across the frame-relay interface using the command:

```
Rack1R6#ping 54.1.1.254 size 1000 repeat 1000000 timeout 0
```

And observe syslog messages. Notice that you may need to set R6's Frame-Relay interface bandwidth to a low value e.g. 64 using the command **bandwidth 64**

```
%HA_EM-6-LOG: INTERFACE_MONITOR: R6's Serial0/0/0 output rate: txload =
231
```

You may also validate the registered policies with EEM and the published events:

```
Rack1R6#show event manager policy registered
```

```
No.  Class      Type      Event Type      Trap  Time Registered
Name
1    applet      user      interface        Off   Sun Dec 6 20:02:35
2009  INTERFACE_MONITOR
    name {Serial0/0/0} parameter {txload} entry_op gt entry_val 204
    entry_type value exit_op lt poll_interval 30.000
    maxrun 20.000
    action 1.0 syslog msg "R6's $_interface_name output rate:
    $_interface_parameter = $_interface_value
```

```
Rack1R6#show event manager history events
```

```
No.  Job Id Proc Status  Time of Event      Event Type
Name
1    1      Actv success  Sun Dec 6 20:08:35 2009  interface
applet: INTERFACE_MONITOR
```

Task 8.1

R5:

```
map-class frame-relay DLCI_504
 frame-relay cir 512000
 frame-relay bc 25600
 frame-relay be 51200
 frame-relay mincir 384000
 frame-relay adaptive-shaping becn
!
map-class frame-relay DLCI_513
 frame-relay cir 128000
 frame-relay bc 6400
 frame-relay be 0
```

```

frame-relay mincir 96000
frame-relay adaptive-shaping becn
!
interface Serial0/0/0
frame-relay traffic-shaping
frame-relay interface-dlci 504
class DLCI_504
frame-relay interface-dlci 513
class DLCI_513

```

Task 8.1 Verification

Check the FRTS configuration:

Rack1R5#show traffic-shape

```

Interface   Se0/0/0
Access Target Byte Sustain Excess Interval Increment Adapt
VC List Rate Limit bits/int bits/int (ms) (bytes) Active
502          56000 875 7000 0 125 875 -
503          56000 875 7000 0 125 875 -
504          512000 9600 25600 51200 50 3200 BECN
513          128000 800 6400 0 50 800 BECN
501          56000 875 7000 0 125 875 -

```

Double-check for more detailed information:

Rack1R5#show frame-relay pvc 504

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

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

<output omitted>

Shaping adapts to BECN

pvc create time 05:50:23, last time pvc status changed 01:50:51

cir 512000 bc 25600 be 51200 byte limit 9600 interval 50

mincir 384000 byte increment 3200 Adaptive Shaping BECN

<output omitted>

Note Be is set to 0, to disable bursting:

Rack1R5#show frame-relay pvc 513

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

DLCI = 513, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

<output omitted>

Shaping adapts to BECN

```
pvc create time 05:50:56, last time pvc status changed 04:16:14
cir 128000    bc 6400    be 0    byte limit 800    interval 50
mincir 96000    byte increment 800    Adaptive Shaping BECN
<output omitted>
```

Task 8.2

R1:

```
ip cef
!
class-map match-all ICMP
  match protocol icmp
!
policy-map POLICE_ICMP
  class ICMP
    police cir 128000 bc 4000
!
interface FastEthernet0/0
  service-policy output POLICE_ICMP
```

Task 8.2 Verification

Check policing parameters:

```
Rack1R1#show policy-map interface fastEthernet 0/0
FastEthernet0/0
```

```
Service-policy output: POLICE_ICMP

Class-map: ICMP (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol icmp
police:
  cir 128000 bps, bc 4000 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    drop
  conformed 0 bps, exceed 0 bps
```

Task 8.3

R5:

```
ip cef
!
class-map match-all CITRIX
  match protocol citrix
!
class-map match-all VOICE
  match dscp ef
!
policy-map CBWFQ
  class VOICE
    priority 64
```

```

class CITRIX
  bandwidth remaining percent 30
  queue-limit 16
class class-default
  fair-queue
!
map-class frame-relay DLCI_504
  service-policy output CBWFQ

```

Task 8.3 Verification

Rack1R5#show frame-relay pvc 504

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

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

```

input pkts 6          output pkts 3          in bytes 204
out bytes 102         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 bcst pkts 3      out bcst bytes 102
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
Shaping adapts to BECN
pvc create time 01:01:30, last time pvc status changed 01:01:10
cir 512000   bc 25600   be 51200   byte limit 9600   interval 50
mincir 384000   byte increment 3200 Adaptive Shaping BECN
pkts 0         bytes 0         pkts delayed 0         bytes delayed 0
shaping inactive
traffic shaping drops 0
service policy CBWFQ
Serial0/0/0: DLCI 504 -

```

Service-policy output: CBWFQ

```

Class-map: VOICE (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: dscp ef (46)
Queueing
  Strict Priority
  Output Queue: Conversation 40
  Bandwidth 64 (kbps) Burst 1600 (Bytes)
  (pkts matched/bytes matched) 0/0
  (total drops/bytes drops) 0/0

Class-map: CITRIX (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol citrix
Queueing
  Output Queue: Conversation 41
  Bandwidth remaining 30 (%)Max Threshold 16 (packets)
  (pkts matched/bytes matched) 0/0
  (depth/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps

```

```
Match: any
Queueing
  Flow Based Fair Queueing
  Maximum Number of Hashed Queues 32
  (total queued/total drops/no-buffer drops) 0/0/0
Output queue size 0/max total 600/drops 0
```

Task 8.4

SW4:

```
mls qos
mls qos map dscp-mutation OSPF_CHANGE 48 to 24
!
interface FastEthernet 0/4
  mls qos dscp-mutation OSPF_CHANGE
  mls qos trust dscp
```

SW2:

```
mls qos
!
interface FastEthernet 0/13
  mls qos vlan-based
!
interface FastEthernet 0/15
  mls qos vlan-based
!
interface FastEthernet 0/16
  mls qos vlan-based
!
interface FastEthernet 0/17
  mls qos vlan-based
!
interface FastEthernet 0/18
  mls qos vlan-based

!
! Classify HTTP traffic
!
ip access-list extended HTTP
  permit tcp any eq 80 any
!
class-map HTTP
  match access-group name HTTP
!
! Mark HTTP traffic with DSCP 16
!
policy-map VLAN46_MARK
  class HTTP
    set dscp 16
  class class-default
    trust dscp
!
interface vlan 46
  service-policy input VLAN46_MARK

!
! Map HTTP traffic (DSCP 16=CS2) to queue 4
```

```

!
mls qos srr-queue output dscp-map queue 4 16

!
! Set interface speed limit to 4 Mbps (40%*10Mbps)
! Shape queue 4 to 1/10 of the interface speed
!
interface FastEthernet 0/6
  srr-queue bandwidth limit 40
  speed 10
  srr-queue bandwidth shape 0 0 0 10

```

Task 8.4 Verification

Configure R6 as follows to perform verification:

```

R6:
ip access-list extended HTTP
  permit tcp any eq www any
ip access-list extended OSPF
  permit ospf any any
!
class-map match-all HTTP_DSCP16
  match ip dscp cs2
  match access-group name HTTP
!
class-map match-all OSPF_DSCP24
  match ip dscp cs3
  match access-group name OSPF
!
!
policy-map METER
  class OSPF_DSCP24
  class HTTP_DSCP16
!
interface FastEthernet 0/0
  service-policy input METER
  load-interval 30

```

Now check the policy-map stats for OSPF packet matches:

```

Rack1R6#show policy-map interface fastEthernet 0/0
FastEthernet0/0

  Service-policy input: METER

    Class-map: OSPF_DSCP24 (match-all)
      18 packets, 1684 bytes
      5 minute offered rate 0 bps
      Match: ip dscp cs3 (24)
      Match: access-group name OSPF

    Class-map: HTTP_DSCP16 (match-all)

```

```
0 packets, 0 bytes
5 minute offered rate 0 bps
Match: ip dscp cs2 (16)
Match: access-group name HTTP

Class-map: class-default (match-any)
32 packets, 2699 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

Now configure R4 as a web-server and transfer the IOS image in R4 to R6:

R4:

```
ip http server
ip http path flash:
ip http authentication enable
```

```
Rack1R6#copy http://cisco:cisco@150.1.4.4/c1841-adventerprisek9-mz.124-24.T.bi$
```

```
Loading http://*****:*****@150.1.4.4/c1841-adventerprisek9-mz.124-24.T.bin !!!!!!!!
```

```
Rack1R6#sh policy-map interface fastEthernet 0/0
FastEthernet0/0
```

```
Service-policy input: METER
```

```
Class-map: OSPF_DSCP24 (match-all)
87 packets, 8162 bytes
30 second offered rate 0 bps
Match: ip dscp cs3 (24)
Match: access-group name OSPF
```

```
Class-map: HTTP_DSCP16 (match-all)
26989 packets, 15906896 bytes
30 second offered rate 979000 bps
Match: ip dscp cs2 (16)
Match: access-group name HTTP
```

```
Class-map: class-default (match-any)
99 packets, 6871 bytes
30 second offered rate 0 bps, drop rate 0 bps
Match: any
```

Notice the HTTP class matches and the traffic rate that is close to 1Mbps.