

1. Bridging and Switching

Task 1.1

SW1 and SW2:

```

define interface-range DOT-ONE-Q FastEthernet0/13 - 15
!
interface FastEthernet0/13
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface FastEthernet0/14
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface FastEthernet0/15
  switchport trunk encapsulation dot1q
  switchport mode trunk

```

Task 1.1 Breakdown

In order to ease in management of ports that have similar configuration, the switches support interface-range macros. These macros are user defined identifiers that represent a range of ports. To define an interface-range macro, use the global configuration command **define interface-range [name] [range]**. After the macro is defined, these ports can be referenced by issuing the **interface range macro [name]** command.

Task 1.1 Verification

Verify macro definition:

```

Rack1SW1#show running-config | include define
define interface-range DOT-ONE-Q FastEthernet0/13 - 15

```

Verify trunking configuration:

```

Rack1SW1#show interfaces trunk

```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	802.1q	trunking	1
Fa0/14	on	802.1q	trunking	1
Fa0/15	on	802.1q	trunking	1

```

Port          Vlans allowed on trunk

```

```

Fa0/13       1-4094

```

```

Fa0/14       1-4094

```

```

Fa0/15       1-4094

```

```

Port          Vlans allowed and active in management domain

```

```

Fa0/13       1,3-8,10,18,68,73,77,232

```

```

Fa0/14       1,3-8,10,18,68,73,77,232

```

```

Fa0/15      1,3-8,10,18,68,73,77,232

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/13      1,3-8,10,18,68,73,77,232
Fa0/14      none
Fa0/15      none
    
```

Rack1SW2#**show interfaces trunk**

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	802.1q	trunking	1
Fa0/14	on	802.1q	trunking	1
Fa0/15	on	802.1q	trunking	1

```

Port        Vlans allowed on trunk
Fa0/13      1-4094
Fa0/14      1-4094
Fa0/15      1-4094
    
```

```

Port        Vlans allowed and active in management domain
Fa0/13      1,3-8,10,18,68,73,77,232
Fa0/14      1,3-8,10,18,68,73,77,232
Fa0/15      1,3-8,10,18,68,73,77,232
    
```

```

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/13      1,3-8,10,18,68,73,77,232
Fa0/14      1,3-8,10,18,68,73,77,232
Fa0/15      1,3-8,10,18,68,73,77,232
    
```

Task 1.2

SW2 and SW3:

```

interface Port-channel13
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface range FastEthernet0/16 - 18
  switchport trunk encapsulation dot1q
  switchport mode trunk
  channel-group 13 mode on
    
```

Quick Note
Any etherchannel group number and mode would be acceptable

Task 1.2 Verification

Rack1SW3#**show etherchannel summary | begin Group**

```

Group Port-channel Protocol Ports
-----+-----+-----+-----+-----+-----
13    Po13(SU)          -          Fa0/16(P) Fa0/17(P) Fa0/18(P)
    
```

Rack1SW3#**show interfaces trunk**

Port	Mode	Encapsulation	Status	Native vlan
Po13	on	802.1q	trunking	1

```

Port        Vlans allowed on trunk
Po13        1-4094
    
```

```
Port          Vlans allowed and active in management domain
Po13          1
```

```
Port          Vlans in spanning tree forwarding state and not pruned
Po13          1
```

Task 1.3

SW3:

```
vlan dot1q tag native
!
interface FastEthernet0/19
  switchport trunk encapsulation dot1q
!
interface FastEthernet0/20
  switchport trunk encapsulation dot1q
```

SW4:

```
vlan dot1q tag native
!
interface FastEthernet0/19
  switchport trunk encapsulation dot1q
  switchport mode dynamic desirable
!
interface FastEthernet0/20
  switchport trunk encapsulation dot1q
  switchport mode dynamic desirable
```

Task 1.3 Verification

```
Rack1SW3#show interfaces trunk | exclude Po13
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/19	auto	802.1q	trunking	1
Fa0/20	auto	802.1q	trunking	1

```
Port          Vlans allowed on trunk
Fa0/19        1-4094
Fa0/20        1-4094
```

```
Port          Vlans allowed and active in management domain
Fa0/19        1
Fa0/20        1
```

```
Port          Vlans in spanning tree forwarding state and not pruned
Fa0/19        1
Fa0/20        none
```

```
Rack1SW4#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/19	desirable	802.1q	trunking	1
Fa0/20	desirable	802.1q	trunking	1

```
Port Vlan allowed on trunk
```

Fa0/19	1-4094
Fa0/20	1-4094

```
Port Vlan allowed and active in management domain
```

Fa0/19	1
Fa0/20	1

```
Port Vlan in spanning tree forwarding state and not pruned
```

Fa0/19	1
Fa0/20	1

Task 1.4

SW1:

```
vtp mode transparent
!
vlan 3,7,18,73,77
!
interface FastEthernet0/1
  switchport access vlan 18
!
interface FastEthernet0/3
  switchport access vlan 3
!
interface FastEthernet0/16
  switchport access vlan 3
```

SW2:

```
vtp mode transparent
!
vlan 4,6,7,8,18,68,73,232
!
interface FastEthernet0/2
  switchport access vlan 232
!
interface FastEthernet0/4
  switchport access vlan 4
!
interface FastEthernet0/6
  switchport access vlan 6
!
interface FastEthernet0/24
  switchport access vlan 232
```

SW3:

```
vtp mode transparent
!
vlan 5,7,68,73,232
!
interface FastEthernet0/3
  switchport access vlan 232
!
interface FastEthernet0/5
  switchport access vlan 5
!
interface FastEthernet0/21
  switchport access vlan 7
!
interface FastEthernet0/24
  switchport access vlan 73
```

SW4:

```
vtp mode transparent
!
vlan 68
!
interface FastEthernet0/6
  switchport access vlan 68
```

Task 1.1 Breakdown

In order to determine what VLANs are necessary on the various devices, start by drawing out the trunk ports that have been configured. We have trunks between SW1 and SW2, SW2 and SW3, and SW3 and SW4, so logically we have a linear switch topology SW1 – SW2 – SW3 – SW4. For some of the VLANs, there is only a single switch that includes ports for that VLAN, so only that switch needs to know about it. These include the following:

SW1 – VLAN 77

SW2 – VLAN 6, VLAN 4

SW3 – VLAN 5

VLAN 18 connects R1 (connected to SW1) to SW2, so needs to be created on SW1 and SW2.

VLAN 73 connects SW1 to BB3 (connected to SW3), so needs to be created on SW1, SW2, and SW3.

VLAN 68 connects SW2 to R6 (connected to SW4), so needs to be created on SW2, SW3, and SW4.

VLAN 232 connects R2 (connected to SW2) to R3 (connected to SW4) and BB2 (connected to SW2), so needs to be created on SW2 and SW3.

VLAN 7 has a connection to SW4 from SW1. Looking at the configuration on SW4, you can see that SW4 is acting as a layer 3 port, so the layer 2 connectivity

for VLAN 7 only needs to go as far as the port where SW4 is connected. SW4's interface Fa0/21 is connected to SW3, so VLAN 7 needs to be configured on SW1, SW2, and SW3, but not SW4.

VLAN 3 has a connection to SW1 from SW3, but the connection on SW3 is configured as a layer 3 port. The port used on SW3 is Fa0/13, which is connected to SW1. Therefore, VLAN 3 only needs to be configured on SW1. If SW3 was configured with the interface as an SVI instead, then layer 2 connectivity for the VLAN would need to be from SW1 to SW3, including SW2.

Task 1.4 Verification

```
Rack1R1#ping 148.1.18.8
```

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

```
Sending 5, 100-byte ICMP Echos to 148.1.18.8, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

```
Rack1R2#ping 192.10.1.3
```

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

```
Sending 5, 100-byte ICMP Echos to 192.10.1.3, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
Rack1R2#ping 192.10.1.254
```

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

```
Sending 5, 100-byte ICMP Echos to 192.10.1.254, timeout is 2 seconds:
```

```
.!!!!
```

```
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/5/8 ms
```

```
Rack1R3#ping 192.10.1.254
```

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

```
Sending 5, 100-byte ICMP Echos to 192.10.1.254, timeout is 2 seconds:
```

```
.!!!!
```

```
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/5/8 ms
```

Rack1R3#ping 148.1.3.9

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.3.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Rack1R5#ping 148.1.57.7

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.57.7, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Rack1R6#ping 148.1.68.8

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.68.8, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Rack1SW1#ping 148.1.7.10

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.7.10, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Rack1SW1#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:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/5/9 ms

© Strategy Tip

Perform a basic connectivity test between the directly connected FastEthernet interfaces before moving forward. This should be one of the goals that you plan to complete within the early stages of the lab.

Task 1.5

SW2:

```
interface FastEthernet0/24
 spanning-tree guard root
```

Task 1.5 Breakdown

Spanning-tree root guard is typically used when a provider is leasing an FastEthernet line out to a customer. In the case that a switch in the customer's network is elected root, all traffic from the provider and its other customers must follow sub-optimal forwarding. Root guard can be used to prevent this case by disabling the port connected to the customer if a superior BPDU is received. The term *superior* BPDU implies that the cost to the root out that port is better than the current root port. To enable root guard, use the interface level command **spanning-tree guard root**.



Further Reading

[Spanning-Tree Protocol Root Guard Enhancement](#)

Task 1.5 Verification

Rack1SW2#**show spanning-tree interface fa0/24 detail**

```
Port 26 (FastEthernet0/24) of VLAN0232 is forwarding
  Port path cost 100, Port priority 128, Port Identifier 128.26.
  Designated root has priority 33000, address 0015.63c8.8800
  Designated bridge has priority 33000, address 0016.9d31.8380
  Designated port id is 128.26, designated path cost 9
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is shared by default
  Root guard is enabled on the port
  BPDU: sent 2346, received 0
```

Task 1.6

SW2:

```
spanning-tree vlan 68 root primary diameter 3 hello-time 1
```


Task 1.6 Verification

```
Rack1SW2#show span vlan 68
```

```
VLAN0068
```

```
Spanning tree enabled protocol ieee
Root ID      Priority      24644
             Address      001b.8f0c.2a00
             This bridge is the root
Hello Time   1 sec      Max Age    7 sec      Forward Delay  5 sec
```

```
<output omitted>
```

Task 1.7

SW1:

```
system mtu 1504
```

```
interface range FastEthernet0/17, Fa0/20
switchport access vlan 100
switchport mode dot1q-tunnel
l2protocol-tunnel cdp
no cdp enable
spanning-tree bpdufilter enable
!
```

```
interface range FastEthernet0/18, Fa0/21
switchport access vlan 101
switchport mode dot1q-tunnel
l2protocol-tunnel cdp
no cdp enable
spanning-tree bpdufilter enable
```

SW3:

```
interface Port-channel1
no switchport
ip address 148.1.1.9 255.255.255.0
!
```

```
interface FastEthernet0/14
no switchport
no ip address
channel-group 1 mode on
!
```

```
interface FastEthernet0/15
no switchport
no ip address
channel-group 1 mode on
```

SW4:

```
interface Port-channel1
no switchport
ip address 148.1.1.10 255.255.255.0
!
```

```
interface FastEthernet0/14
no switchport
no ip address
```

```

channel-group 1 mode on
!
interface FastEthernet0/15
no switchport
no ip address
channel-group 1 mode on

```



Further Reading

[Configuring Layer 2 Tunneling for EtherChannels](#)

Task 1.7 Verification

```
Rack1SW3#show etherchannel summary | begin Group
```

Group	Port-channel	Protocol	Ports		
1	Po1(RU)	-	Fa0/14(P)	Fa0/15(P)	
13	Po13(SU)	-	Fa0/16(P)	Fa0/17(P)	Fa0/18(P)

```
Rack1SW3#ping 148.1.1.10
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.1.10, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Task 1.7 Breakdown

Use a separate VLAN for each of the two links. The VLANs only need to be present on SW1, since SW1 has the connections to both SW3 and SW4.

Task 1.8

R1:

```

interface Serial0/0
ip address 148.1.0.1 255.255.255.0
frame-relay map ip 148.1.0.2 102 broadcast
frame-relay map ip 148.1.0.3 102
frame-relay map ip 148.1.0.4 104 broadcast
no frame-relay inverse-arp

```

R2:

```

interface Serial0/0
ip address 148.1.0.2 255.255.255.0
frame-relay map ip 148.1.0.1 201 broadcast
frame-relay map ip 148.1.0.3 203 broadcast
frame-relay map ip 148.1.0.4 201
no frame-relay inverse-arp

```

R3:

```

interface Serial1/0.302 point-to-point
ip address 148.1.0.3 255.255.255.0

```

```
frame-relay interface-dlci 302
```

R4:

```
interface Serial0/0.401 point-to-point
 ip address 148.1.0.4 255.255.255.0
 frame-relay interface-dlci 401
```

Task 1.8 Verification**Rack1R1#show frame-relay map**

```
Serial0/0 (up): ip 148.1.0.2 dlci 102(0x66,0x1860), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 148.1.0.3 dlci 102(0x66,0x1860), static,
                CISCO, status defined, active
Serial0/0 (up): ip 148.1.0.4 dlci 104(0x68,0x1880), static,
                broadcast,
                CISCO, status defined, active
```

Rack1R2#show frame-relay map

```
Serial0/0 (up): ip 148.1.0.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 148.1.0.3 dlci 203(0xCB,0x30B0), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 148.1.0.4 dlci 201(0xC9,0x3090), static,
                CISCO, status defined, active
```

Rack1R3#show frame-relay map

```
Serial1/0.302 (up): point-to-point dlci, dlci 302(0x12E,0x48E0),
broadcast
                status defined, active
```

Rack1R4#show frame-relay map

```
Serial0/0.401 (up): point-to-point dlci, dlci 401(0x191,0x6410),
broadcast
                status defined, active
```

Rack1R1#ping 148.1.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.0.2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

Rack1R1#ping 148.1.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.0.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms

Rack1R1#ping 148.1.0.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.0.4, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

Task 1.9

R3:

```
interface Serial1/1
 ip address 148.1.35.3 255.255.255.0
 no frame-relay inverse-arp ip 311
 no frame-relay inverse-arp ip 312
 no frame-relay inverse-arp ip 314
```

R5:

```
interface Serial0/0
 ip address 148.1.35.5 255.255.255.0
 no frame-relay inverse-arp ip 501
 no frame-relay inverse-arp ip 502
 no frame-relay inverse-arp ip 503
 no frame-relay inverse-arp ip 504
```

Task 1.9 Verification

Rack1R3#**show frame-relay map**

```
Serial1/0.302 (up): point-to-point dlci, dlci 302(0x12E,0x48E0),
broadcast
```

```
    status defined, active
```

```
Serial1/1 (up): ip 148.1.35.5 dlci 315(0x13B,0x4CB0), dynamic,
    broadcast,, status defined, active
```

Rack1R5#**show frame-relay map**

```
Serial0/0 (up): ip 148.1.35.3 dlci 513(0x201,0x8010), dynamic,
    broadcast,, status defined, active
```

Rack1R5#**ping 148.1.35.3**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 148.1.35.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Task 1.10

R4:

```
interface Serial0/0.401 point-to-point
 backup delay 0 300
 backup interface Serial0/1
```

R5:

```
interface Serial0/1
 clockrate 64000
```

Task 1.10 Verification

Verify the backup configuration:

```
Rack1R4#show backup
Primary Interface    Secondary Interface    Status
-----
Serial0/0.401       Serial0/1              normal operation
```

Test the backup configuration:

```
Rack1R4(config)#interface s0/0.401
Rack1R4(config-subif)#do debug backup
Backup events debugging is on
Rack1R4(config-subif)#no frame-relay interface-dlci 401
BACKUP(Serial0/0.401): event = primary interface went down
BACKUP(Serial0/0.401): changed state to "waiting to backup"
BACKUP(Serial0/0.401): event = timer expired on primary
BACKUP(Serial0/0.401): secondary interface (Serial0/1) made active
BACKUP(Serial0/0.401): changed state to "backup mode"
%LINK-3-UPDOWN: Interface Serial0/1, changed state to up
BACKUP(Serial0/1): event = secondary interface came up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, changed
state to up
BACKUP(Serial0/1): event = secondary interface came up
```

```
Rack1R4(config-subif)#do show backup
Primary Interface    Secondary Interface    Status
-----
Serial0/0.401       Serial0/1              backup mode
```

```
Rack1R4(config-subif)# frame-relay interface-dlci 401
BACKUP(Serial0/0.401): event = primary interface came up
BACKUP(Serial0/0.401): changed state to "waiting to revert"
Rack1R4(config-fr-dlci)#exit
```

```
Rack1R4(config-subif)#do show backup
Primary Interface    Secondary Interface    Status
-----
Serial0/0.401       Serial0/1              waiting to revert (290 more
seconds)
```

2. IP IGP Routing

Task 2.1

R1:

```
router ospf 1
 router-id 150.1.1.1
 network 148.1.18.1 0.0.0.0 area 168
 network 150.1.1.1 0.0.0.0 area 168
!
interface FastEthernet0/0
 ip ospf priority 0
```

R6:

```
router ospf 1
 router-id 150.1.6.6
 network 148.1.6.6 0.0.0.0 area 168
```

```

network 148.1.68.6 0.0.0.0 area 168
network 150.1.6.6 0.0.0.0 area 168
!
interface FastEthernet0/1
 ip ospf priority 0

```

SW2:

```

ip routing
!
router ospf 1
 router-id 150.1.8.8
 network 148.1.8.8 0.0.0.0 area 168
 network 148.1.18.8 0.0.0.0 area 168
 network 148.1.68.8 0.0.0.0 area 168
 network 150.1.8.8 0.0.0.0 area 168

```

 **Note**

Single area OSPF implementations do not require area 0.

Task 2.1 Verification

Verify the OSPF neighbors:

Rack1SW2#**show ip ospf neighbor**

Neighbor ID	Pri	State	Dead Time	Address
Interface				
150.1.6.6	0	FULL/DROTHER	00:00:39	148.1.68.6 Vlan68
150.1.1.1	0	FULL/DROTHER	00:00:34	148.1.18.1 Vlan18

Verify the loopback network advertisement:

Rack1R1#**show ip route ospf | include 150**

```

150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.8.8/32 [110/2] via 148.1.18.8, 00:02:46, FastEthernet0/0
O       150.1.6.6/32 [110/3] via 148.1.18.8, 00:02:46, FastEthernet0/0

```

Rack1R6#**show ip route ospf | include 150**

```

150.1.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       150.1.8.8/32 [110/2] via 148.1.68.8, 00:03:11, FastEthernet0/1
O       150.1.1.1/32 [110/3] via 148.1.68.8, 00:03:11, FastEthernet0/1

```

Task 2.2

R1 and SW2:

```
router ospf 1
  ignore lsa mospf
```

Task 2.2 Breakdown

Cisco's OSPF implementation does not support type 6 LSA (multicast OSPF). By default, every time one of these LSAs is received, a syslog message is generated. To disable this behavior, issue the OSPF routing process subcommand **ignore lsa mospf**.



Standard

[RFC 1584: Multicast Extensions to OSPF](#)

Task 2.3

R1:

```
service password-encryption
!
interface FastEthernet0/0
  ip ospf message-digest-key 7 md5 CISCO
!
router ospf 1
  area 168 authentication message-digest
```

SW2:

```
service password-encryption
!
interface Vlan18
  ip ospf message-digest-key 7 md5 CISCO
!
interface Vlan68
  ip ospf authentication null
!
router ospf 1
  area 168 authentication message-digest
```

Task 2.3 Breakdown

As previously covered, there are two ways to enable OSPF authentication, on a per area basis and on a per interface basis. As the above task states, that the **ip ospf authentication message-digest** command cannot be used, area authentication must be used. However, this task also states that the adjacency between R6 and SW2 must not be authenticated. Since R1, R6, and SW2 are all in the same area, this presents a problem. This task illustrates that there are actually three types of OSPF authentication, MD5, clear text, and NULL. By setting the OSPF authentication type to NULL on VLAN 68, SW2 has effectively disabled OSPF authentication on that interface.

Next, this task states that R1 and SW2 should use a pre-encrypted key with the number 7. This task is designed to illustrate the difference between key number and encryption type. The key number of an MD5 key is used as a seed or salt value in the MD5 hash algorithm. This seed is a number used to randomize the output of the hash algorithm, and decrease the effectiveness of a brute force attack on the MD5 algorithm. Key numbers must match on all devices authenticating on the segment.

The encryption type determines whether or not the password is stored in a clear-text or encrypted form in the router's configuration file. By issuing the **service password-encryption** global configuration command, all clear text passwords in the routers configuration are encrypted with type 7 encryption. Type 7 encryption uses a Cisco proprietary insecure reversible encryption algorithm, based on a Vigenere cipher. This encryption is simply used to shield a password from an over the shoulder user seeing the password in show commands or backups or configuration files.



Further Reading

[Passwords and Privileges Commands](#)

[Cisco IOS Password Encryption Facts](#)

Task 2.3 Verification

```
Rack1SW2#show ip ospf interface vl18 | begin Message
Message digest authentication enabled
Youngest key id is 7
```

```
Rack1SW2#show running-config interface vl18 | begin Message
Rack1SW2#
```

Verify password encryption:

```
Rack1R1#show running-config interface fa0/0

interface FastEthernet0/0
 ip address 148.1.18.1 255.255.255.0
 ip ospf message-digest-key 7 md5 7 106D202A2638
 ip ospf priority 0
 duplex auto
 speed auto
```

```
Rack1SW2#show running-config interface vl18

interface Vlan18
 ip address 148.1.18.8 255.255.255.0
 ip ospf message-digest-key 7 md5 7 00273A352774
```

Task 2.4

R3:

```
router eigrp 100
 eigrp router-id 150.1.3.3
 network 148.1.3.3 0.0.0.0
 network 148.1.35.3 0.0.0.0
 no auto-summary
```

R4:

```
router eigrp 100
 eigrp router-id 150.1.4.4
 network 148.1.45.4 0.0.0.0
 no auto-summary
```

R5:

```
router eigrp 100
 eigrp router-id 150.1.5.5
 network 150.1.5.5 0.0.0.0
 network 148.1.5.5 0.0.0.0
 network 148.1.35.5 0.0.0.0
 network 148.1.45.5 0.0.0.0
 network 148.1.57.5 0.0.0.0
 no auto-summary
```

SW1:

```

ip routing
!
router eigrp 100
  eigrp router-id 150.1.7.7
  network 150.1.7.7 0.0.0.0
  network 148.1.7.7 0.0.0.0
  network 148.1.57.7 0.0.0.0
  network 148.1.77.7 0.0.0.0
  no auto-summary

```

Task 2.4 Verification

Verify the EIGRP neighbors (note that R4 will not appear until the backup link is active):

Rack1R5#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
1	148.1.57.7	Fa0/0	14	00:02:39	1	200	0	2
0	148.1.35.3	Se0/0	122	00:02:46	39	234	0	8

Verify the EIGRP routes:

Rack1R3#show ip route eigrp

```

148.1.0.0/24 is subnetted, 7 subnets
D    148.1.5.0 [90/337920] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    148.1.7.0 [90/286720] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    148.1.1.0 [90/284160] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    148.1.57.0 [90/312320] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    148.1.77.0 [90/286976] via 148.1.3.9, 00:00:22, FastEthernet0/0
150.1.0.0/24 is subnetted, 5 subnets
D    150.1.7.0 [90/414720] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    150.1.5.0 [90/440320] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    150.1.10.0 [90/412160] via 148.1.3.9, 00:00:22, FastEthernet0/0
D    150.1.9.0 [90/409600] via 148.1.3.9, 00:00:22, FastEthernet0/0

```

Task 2.5

R3:

```
interface Serial1/1
 ip hello-interval eigrp 100 4
 ip hold-time eigrp 100 12
```

R5:

```
interface Serial0/0
 ip hello-interval eigrp 100 4
 ip hold-time eigrp 100 12
```

Task 2.5 Breakdown

To adjust neighbor hello and dead intervals in EIGRP, use the interface level commands `ip hello-interval eigrp [AS] [hello_interval]` and `ip hold-time eigrp [AS] [hold_time]`. By default, the EIGRP hello interval is 60 seconds for low speed NBMA interfaces and 5 seconds for all other media. The hold-time defaults to three times these values.

Task 2.5 Verification

Verify the EIGRP interface characteristics:

```
Rack1R5#show ip eigrp interfaces detail s0/0
IP-EIGRP interfaces for process 100
```

Interface	Peers	Xmit Queue Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Se0/0	1	0/0	39	0/15	159	0

```

Hello interval is 4 sec
Next xmit serial <none>
Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/7
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 1
Retransmissions sent: 1  Out-of-sequence rcvd: 0
Authentication mode is not set
```

```
Rack1R3#show ip eigrp interfaces detail s1/1
IP-EIGRP interfaces for process 100
```

Interface	Peers	Xmit Queue Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Se1/1	1	0/0	663	5/190	3454	0

```

Hello interval is 4 sec
Next xmit serial <none>
Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/18
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 1
Retransmissions sent: 11  Out-of-sequence rcvd: 0
Authentication mode is not set
```

Task 2.6

R5 and SW1:

```
key chain EIGRP
key 1
key-string CISCO2009
accept-lifetime 00:00:00 Jan 1 1993 00:15:00 Jan 1 2010
send-lifetime 00:00:00 Jan 1 1993 23:45:00 Dec 31 2009
key 2
key-string CISCO2010
accept-lifetime 23:15:00 Dec 31 2009 infinite
send-lifetime 23:45:00 Dec 31 2009 infinite
```

R5:

```
interface FastEthernet0/0
 ip authentication mode eigrp 100 md5
 ip authentication key-chain eigrp 100 EIGRP
```

SW1:

```
interface FastEthernet0/5
 ip authentication mode eigrp 100 md5
 ip authentication key-chain eigrp 100 EIGRP
```

Task 2.6 Breakdown

Key chain authentication allows for key lifetime and rotation based on time. This option allows for smooth transition between authentication keys throughout the entire network at the same time. The two options that dictate a key's timing are the **accept-lifetime** and the **send-lifetime**. As their names imply, the accept lifetime is the time period for which the specified key will be accepted from a neighbor as valid for authentication. The send-lifetime specifies during which time interval the key will be valid for transmission to a neighbor. The **infinite** option dictates that the specified key is valid from the start time on.

Note

To ensure smooth key transition in a real network, NTP should be used in any practical time based key chain authentication implementations.

Further Reading

[IP Routing Protocol Independent Commands: accept-lifetime and send-lifetime](#)

Task 2.6 Verification

Verify EIGRP authentication:

```
Rack1SW1#show ip eigrp interfaces detail fa0/5
```

```
IP-EIGRP interfaces for process 100
```

```

          Xmit Queue   Mean   Pacing Time   Multicast   Pending
Interface Peers Un/Reliable SRTT   Un/Reliable Flow Timer Routes
Fa0/5      1      0/0         4      0/10        50         0
  Next xmit serial <none>
  Un/reliable mcasts: 0/2  Un/reliable ucasts: 5/5
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 3  Out-of-sequence rcvd: 0
  Authentication mode is md5, key-chain is "EIGRP"

```

```
Rack1SW1#show key chain EIGRP
```

```
Key-chain EIGRP:
```

```

  key 1 -- text "CISCO2005"
    accept lifetime (00:00:00 UTC Jan 1 1993) - (00:15:00 UTC Jan 1
2006) [valid now]
    send lifetime (00:00:00 UTC Jan 1 1993) - (23:45:00 UTC Dec 31
2005) [valid now]
  key 2 -- text "CISCO2006"
    accept lifetime (23:15:00 UTC Dec 31 2005) - (infinite)
    send lifetime (23:45:00 UTC Dec 31 2005) - (infinite)

```

```
Rack1SW1#show ip eigrp neighbors
```

```
IP-EIGRP neighbors for process 100
```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num	Type
0	148.1.57.5	Fa0/5	13 00:01:38	4	200	0	14	

Task 2.7

SW1:

```

interface Vlan73
 ip rip send version 1 2
!
router rip
 version 2
 network 204.12.1.0
 no auto-summary

```

Task 2.7 Breakdown

When the RIP routing process is enabled, all interfaces are configured for RIP send RIP version 1 updates, and listed for version 2 updates. This can be modified by the routing process subcommand **rip version [1 | 2]** or the interface level commands **ip rip send version** and **ip rip receive version** commands. The interface level commands always override the process level version command.

☑ RIP Version Verification

```

Router#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 23 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 1, receive any version
    Interface          Send Recv Triggered RIP Key-chain
  FastEthernet0/0      1      1 2
  Serial0/0            1      1 2
  Loopback0            1      1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway           Distance      Last Update
  Distance: (default is 120)

```

Task 2.7 Verification

```

Rack1SW1#show ip route rip
  31.0.0.0/16 is subnetted, 4 subnets
R       31.3.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       31.2.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       31.1.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       31.0.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
  30.0.0.0/16 is subnetted, 4 subnets
R       30.2.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       30.3.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       30.0.0.0 [120/1] via 204.12.1.254, 00:00:07, Vlan73
R       30.1.0.0 [120/1] via 204.12.1.254, 00:00:08, Vlan73

```

Task 2.8**R1:**

```
router rip
  version 2
  passive-interface default
  no passive-interface Serial0/0
  network 148.1.0.0
  no auto-summary
```

R2:

```
router rip
  version 2
  passive-interface default
  no passive-interface FastEthernet0/0
  no passive-interface Serial0/0
  network 150.1.0.0
  network 148.1.0.0
  network 192.10.1.0
  no auto-summary
```

R3:

```
router rip
  version 2
  passive-interface default
  no passive-interface FastEthernet0/1
  no passive-interface Serial1/0.302
  network 150.1.0.0
  network 148.1.0.0
  network 192.10.1.0
  no auto-summary
```

R4:

```
router rip
  version 2
  passive-interface default
  no passive-interface Serial 0/0.401
  network 150.1.0.0
  network 148.1.0.0
  no auto-summary
```

Task 2.8 Verification

Verify the RIP routes:

```
Rack1R3#show ip route rip
  148.1.0.0/24 is subnetted, 9 subnets
R       148.1.18.0 [120/2] via 192.10.1.2, 00:00:17, FastEthernet0/1
        [120/2] via 148.1.0.1, 00:00:17, Serial1/0.302
R       148.1.4.0 [120/3] via 192.10.1.2, 00:00:17, FastEthernet0/1
        [120/3] via 148.1.0.4, 00:00:17, Serial1/0.302
  150.1.0.0/24 is subnetted, 5 subnets
R       150.1.4.0 [120/3] via 192.10.1.2, 00:00:17, FastEthernet0/1
        [120/3] via 148.1.0.4, 00:00:17, Serial1/0.302
R       150.1.2.0 [120/1] via 192.10.1.2, 00:00:17, FastEthernet0/1
        [120/1] via 148.1.0.2, 00:00:17, Serial1/0.302
```

```
Rack1R2#show ip route rip
  148.1.0.0/24 is subnetted, 5 subnets
R       148.1.18.0 [120/1] via 148.1.0.1, 00:00:11, Serial0/0
R       148.1.4.0 [120/2] via 148.1.0.4, 00:00:11, Serial0/0
R       148.1.3.0 [120/1] via 192.10.1.3, 00:00:06, FastEthernet0/0
        [120/1] via 148.1.0.3, 00:00:15, Serial0/0
R       148.1.35.0 [120/1] via 148.1.0.3, 00:00:15, Serial0/0
        [120/1] via 192.10.1.3, 00:00:06, FastEthernet0/0
  150.1.0.0/24 is subnetted, 3 subnets
R       150.1.4.0 [120/2] via 148.1.0.4, 00:00:11, Serial0/0
R       150.1.3.0 [120/1] via 148.1.0.3, 00:00:15, Serial0/0
        [120/1] via 192.10.1.3, 00:00:06, FastEthernet0/0
```

```
Rack1R4#show ip route rip
R       192.10.1.0/24 [120/2] via 148.1.0.2, 00:00:24, Serial0/0.401
  148.1.0.0/24 is subnetted, 4 subnets
R       148.1.18.0 [120/1] via 148.1.0.1, 00:00:24, Serial0/0.401
R       148.1.35.0 [120/3] via 148.1.0.3, 00:00:24, Serial0/0.401
  150.1.0.0/24 is subnetted, 3 subnets
R       150.1.3.0 [120/3] via 148.1.0.3, 00:00:24, Serial0/0.401
R       150.1.2.0 [120/2] via 148.1.0.2, 00:00:24, Serial0/0.401
```

Verify connectivity:

```
Rack1R4#ping 150.1.3.3
```

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 = 64/65/68 ms

```
Rack1R4#traceroute 150.1.3.3
```

Type escape sequence to abort.

Tracing the route to 150.1.3.3

```
 1 148.1.0.1 16 msec 16 msec 20 msec
 2 148.1.0.2 16 msec 16 msec 20 msec
 3 148.1.0.3 32 msec * 32 msec
```


Task 2.9

SW2:

```
vlan access-map RIP_FILTER 10
  action drop
  match ip address 100
vlan access-map RIP_FILTER 20
  action forward
vlan filter RIP_FILTER vlan-list 232
!
access-list 100 permit udp host 192.10.1.254 eq rip any eq rip
```

Task 2.9 Breakdown

Although SW2 is only switching at layer two for the above network segment, it can still filter traffic above layer 3 by using VLAN Access-Lists (VACLs). The above VACL matches all RIP traffic coming from BB2 and discards it. Make sure to add the explicit permit sequence so that all other traffic is forwarded unmodified. Alternatively, an access list could also be applied inbound on the port connecting to BB2.



Further Reading

[Configuring Network Security with ACLs](#)

Task 2.9 Verification

Before filter has been applied:

```
Rack1R2(config)#access-list 100 permit udp any any eq 520
Rack1R2#debug interface fastEthernet 0/0
Condition 1 set
Rack1R2#debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100
IP: s=192.10.1.254 (FastEthernet0/0), d=224.0.0.9, len 132, rcvd 2
    UDP src=520, dst=520
IP: s=192.10.1.2 (local), d=224.0.0.9 (FastEthernet0/0), len 172,
sending broad/multicast
    UDP src=520, dst=520
IP: s=192.10.1.254 (FastEthernet0/0), d=224.0.0.9, len 132, rcvd 2
```

After filter has been applied:

```
Rack1R2#debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100

IP: s=192.10.1.2 (local), d=224.0.0.9 (FastEthernet0/0), len 132,
sending broad/multicast
    UDP src=520, dst=520
IP: s=192.10.1.3 (FastEthernet0/0), d=224.0.0.9, len 112, rcvd 2
```

```

UDP src=520, dst=520
IP: s=192.10.1.2 (local), d=224.0.0.9 (FastEthernet0/0), len 132,
sending broad/multicast
UDP src=520, dst=520

```

Task 2.10

R3:

```

router rip
no passive-interface FastEthernet0/0

```

SW1:

```

router rip
passive-interface default
no passive-interface Vlan7
no passive-interface Vlan73
network 148.1.0.0

```

SW3 and SW4:

```

router rip
version 2
network 148.1.0.0
network 150.1.0.0
distance 80
distance 200 0.0.0.0 255.255.255.255 1
no auto-summary
!
access-list 1 permit 150.1.0.0 0.0.255.255

```

Quick Note

Any value lower than EIGRP's administrative distance could be used here

Quick Note

Ensure EIGRP is used for any of the 150.1.0.0/16 subnets. Altering EIGRP administrative distance for these routes would also be a valid solution.

Task 2.10 Verification

Note: Make sure to verify this section after completing redistribution. The networks for the loopbacks of R2, R3 and R4, for example, are only known via RIP initially, since those devices are not running EIGRP.

```
Rack1SW3#show ip route rip
```

```

R    204.12.1.0/24 [80/2] via 148.1.1.10, 00:00:22, Port-channel1
R    192.10.1.0/24 [80/1] via 148.1.3.3, 00:00:10, FastEthernet0/13
    148.1.0.0/24 is subnetted, 13 subnets
R      148.1.18.0 [80/3] via 148.1.3.3, 00:00:10, FastEthernet0/13
R      148.1.5.0 [80/1] via 148.1.3.3, 00:00:10, FastEthernet0/13
R      148.1.4.0 [80/4] via 148.1.3.3, 00:00:10, FastEthernet0/13
R      148.1.7.0 [80/1] via 148.1.1.10, 00:00:22, Port-channel1
R      148.1.6.0 [80/3] via 148.1.3.3, 00:00:10, FastEthernet0/13
R      148.1.0.0 [80/1] via 148.1.3.3, 00:00:11, FastEthernet0/13
R      148.1.8.0 [80/3] via 148.1.3.3, 00:00:11, FastEthernet0/13
R      148.1.57.0 [80/1] via 148.1.3.3, 00:00:11, FastEthernet0/13
R      148.1.35.0 [80/1] via 148.1.3.3, 00:00:11, FastEthernet0/13
R      148.1.68.0 [80/3] via 148.1.3.3, 00:00:11, FastEthernet0/13
R      148.1.77.0 [80/2] via 148.1.1.10, 00:00:22, Port-channel1
    31.0.0.0/16 is subnetted, 4 subnets
R      31.3.0.0 [80/3] via 148.1.1.10, 00:00:22, Port-channel1
R      31.2.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channel1
R      31.1.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channel1
R      31.0.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channel1
    30.0.0.0/16 is subnetted, 4 subnets

```

```
R      30.2.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channell
R      30.3.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channell
R      30.0.0.0 [80/3] via 148.1.1.10, 00:00:23, Port-channell
R      30.1.0.0 [80/3] via 148.1.1.10, 00:00:24, Port-channell
```

Rack1SW3#**show ip route eigrp**

```
150.1.0.0/16 is variably subnetted, 10 subnets, 2 masks
D      150.1.7.0/24 [90/158720] via 148.1.1.10, 00:00:14, Port-
channell
D      150.1.5.0/24 [90/20642560] via 148.1.3.3, 00:23:20,
FastEthernet0/13
D EX   150.1.4.0/24 [170/514560] via 148.1.3.3, 00:23:20,
FastEthernet0/13
D EX   150.1.3.0/24 [170/514560] via 148.1.3.3, 00:23:20,
FastEthernet0/13
D EX   150.1.2.0/24 [170/514560] via 148.1.3.3, 00:23:20,
FastEthernet0/13
D EX   150.1.1.0/24 [170/514560] via 148.1.3.3, 00:23:20,
FastEthernet0/13
D EX   150.1.8.8/32 [170/514560] via 148.1.3.3, 00:23:21,
FastEthernet0/13
D EX   150.1.6.6/32 [170/514560] via 148.1.3.3, 00:16:54,
FastEthernet0/13
D      150.1.10.0/24 [90/143360] via 148.1.1.10, 00:23:21, Port-
channell
Rack1SW3#
```

Rack1SW4#**show ip route rip**

```
R      204.12.1.0/24 [80/1] via 148.1.7.7, 00:00:24, FastEthernet0/21
R      192.10.1.0/24 [80/2] via 148.1.1.9, 00:00:03, Port-channell
148.1.0.0/24 is subnetted, 13 subnets
R      148.1.18.0 [80/4] via 148.1.1.9, 00:00:03, Port-channell
R      148.1.5.0 [80/2] via 148.1.1.9, 00:00:03, Port-channell
R      148.1.4.0 [80/5] via 148.1.1.9, 00:00:03, Port-channell
R      148.1.6.0 [80/4] via 148.1.1.9, 00:00:03, Port-channell
R      148.1.0.0 [80/2] via 148.1.1.9, 00:00:03, Port-channell
R      148.1.3.0 [80/1] via 148.1.1.9, 00:00:04, Port-channell
R      148.1.8.0 [80/4] via 148.1.1.9, 00:00:04, Port-channell
R      148.1.57.0 [80/1] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      148.1.35.0 [80/2] via 148.1.1.9, 00:00:04, Port-channell
R      148.1.68.0 [80/4] via 148.1.1.9, 00:00:04, Port-channell
R      148.1.77.0 [80/1] via 148.1.7.7, 00:00:25, FastEthernet0/21
31.0.0.0/16 is subnetted, 4 subnets
R      31.3.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      31.2.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      31.1.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      31.0.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
30.0.0.0/16 is subnetted, 4 subnets
R      30.2.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      30.3.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      30.0.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
R      30.1.0.0 [80/2] via 148.1.7.7, 00:00:25, FastEthernet0/21
```

Rack1SW4#**show ip route eigrp**

```
150.1.0.0/16 is variably subnetted, 10 subnets, 2 masks
D      150.1.7.0/24 [90/156160] via 148.1.7.7, 00:01:10,
FastEthernet0/21
```

```

D      150.1.5.0/24 [90/20645120] via 148.1.1.9, 00:24:16, Port-
channell
D EX   150.1.4.0/24 [170/517120] via 148.1.1.9, 00:24:09, Port-
channell
D EX   150.1.3.0/24 [170/517120] via 148.1.1.9, 00:24:09, Port-
channell
D EX   150.1.2.0/24 [170/517120] via 148.1.1.9, 00:24:09, Port-
channell
D EX   150.1.1.0/24 [170/517120] via 148.1.1.9, 00:24:09, Port-
channell
D EX   150.1.8.8/32 [170/517120] via 148.1.1.9, 00:24:09, Port-
channell
D EX   150.1.6.6/32 [170/517120] via 148.1.1.9, 00:17:50, Port-
channell
D      150.1.9.0/24 [90/143360] via 148.1.1.9, 00:27:07, Port-channell

```

Rack1R3#**show ip route | include 150.1.9|150.1.10|148.1.1.0**

```

D      148.1.1.0 [90/284160] via 148.1.3.9, 01:28:45, FastEthernet0/0
D      150.1.10.0/24 [90/412160] via 148.1.3.9, 00:30:07,
FastEthernet0/0
D      150.1.9.0/24 [90/409600] via 148.1.3.9, 01:28:45,
FastEthernet0/0

```

Rack1SW1#**show ip route | include 150.1.9|150.1.10|148.1.1.0**

```

D      148.1.1.0 [90/15616] via 148.1.7.10, 01:34:27, Vlan7
D      150.1.10.0/24 [90/130816] via 148.1.7.10, 01:34:27, Vlan7
D      150.1.9.0/24 [90/143616] via 148.1.7.10, 00:29:10, Vlan7

```

Task 2.11

R1:

```

router ospf 1
 redistribute rip subnets
!
router rip
 redistribute ospf 1 metric 1

```

R3:

```

router eigrp 100
 redistribute rip metric 10000 1000 255 1 1500
!
router rip
 redistribute eigrp 100 metric 1

```

R5:

```

interface Serial0/1
 ip summary-address eigrp 100 0.0.0.0 0.0.0.0
!
router eigrp 100
 redistribute static metric 64 10 255 1 1500
!
ip route 150.1.4.0 255.255.255.0 148.1.45.4 5

```

Task 2.11 Breakdown

The `default-information [in | out]` statement in EIGRP does not generate a default route advertisement like other IGPs. Instead, it simply allows a default route which already exists to be received or propagated. EIGRP, like IGRP, uses the `ip default-network` statement to propagate default information. A default network must be a classful network dynamically learned that is not directly connected. However, a default-network propagates in all directions. In the above scenario, it is specified that R5 should generate a default route specifically to R4. Since a default-network advertisement cannot be filtered out without filtering the actual network, `ip default-network` cannot be used in this case.

A default route is the most generic IPv4 summary address there is, having a subnet mask of zero. Therefore, a default route can be generated by using an interface summary-address of 0.0.0.0. The above configuration example dictates so.



Note

Although nothing has been official announced by Cisco we should see a `default-information originate` command added to the IOS in future versions.

We are told that we need to make sure that R4's HDLC and loopback interfaces still have reachability when R4's frame connection is down, and that we can add a static route. R5 is already advertising the HDLC link into EIGRP when that interface is up, so the network to be concerned with is R4's loopback, which is advertised into RIP when the Frame connection is up. R5 will need to have connectivity for this network. Here, that is achieved by adding a static route to the loopback, with R4's serial address as the next hop. The static is then redistributed into EIGRP. Alternatively, we could have R4 add the loopback network to EIGRP, since the adjacency between R4 and R5 is only up when the serial link between those two is not in the backup state.

Task 2.12

SW1:

```
router eigrp 100
 redistribute rip metric 10000 1000 255 1 1500
!
router rip
 redistribute eigrp 100 metric 15
```

Task 2.12 Breakdown

RIP defines an infinite (unreachable) metric as 16. Metric is incremented as a route advertisement exits an interface. The above task states that the devices on VLAN 73 should not be able to pass on RIP updates learned from SW1. By redistributing prefixes from the EIGRP domain into the RIP domain with a metric of 15, their metric will be infinite when BB3 or any other device tries to pass them on.

Task 2.11 – 2.12 Verification

First make sure that SW1 advertises prefixes with a metric of 15:

```
Rack1SW1#debug ip rip
```

```
RIP protocol debugging is on
```

```
RIP: sending v1 update to 255.255.255.255 via Vlan73 (204.12.1.7)
RIP: build update entries
    network 148.1.0.0 metric 15
    network 150.1.0.0 metric 15
    network 192.10.1.0 metric 15
RIP: sending v2 update to 224.0.0.9 via Vlan73 (204.12.1.7)
RIP: build update entries
    148.1.0.0/24 via 0.0.0.0, metric 15, tag 0
    148.1.3.0/24 via 0.0.0.0, metric 15, tag 0
    148.1.4.0/24 via 0.0.0.0, metric 15, tag 0
<output omitted>
```

*Next, confirm full connectivity between internal routers. There are two cases here: first when R4 primary link is up and second when backup link is active. For the second case to settle, you need to wait for old RIP routes to expire on R3. You may want to speed up convergence and to do so just do a "clear ip route *" on every RIP speaking router.*

Use the following TCL script to test connectivity:

```
foreach i {
148.1.18.1
148.1.0.1
150.1.1.1
148.1.0.2
150.1.2.2
192.10.1.2
148.1.3.3
148.1.0.3
150.1.3.3
148.1.35.3
192.10.1.3
148.1.0.4
150.1.4.4
148.1.5.5
150.1.5.5
148.1.57.5
148.1.35.5
148.1.6.6
150.1.6.6
```

```
148.1.68.6
148.1.7.7
150.1.7.7
148.1.57.7
204.12.1.7
148.1.77.7
148.1.18.8
148.1.8.8
150.1.8.8
148.1.68.8
148.1.1.9
148.1.1.10
150.1.9.9
150.1.10.10
} { ping $i }
```

Note that the Frame Relay link on R6 as well as backup link between R4 & R5, and VLAN4 are excluded from this connectivity test.

Task 2.13

R3:

```
ipv6 unicast-routing
!
interface Loopback0
  ipv6 address 2002:9601:303::3/64
!
interface FastEthernet0/0
  ipv6 address 2002:9601:303:1::3/64
```

R4:

```
ipv6 unicast-routing
!
interface Loopback0
  ipv6 address 2002:9601:404::4/64
!
interface FastEthernet0/0
  ipv6 address 2002:9601:404:1::4/64
```

R5:

```
ipv6 unicast-routing
!
interface Loopback0
  ipv6 address 2002:9601:505::5/64
!
interface FastEthernet0/1
  ipv6 address 2002:9601:505:1::5/64
```

R6:

```
ipv6 unicast-routing
!
interface Loopback0
  ipv6 address 2002:9601:606::6/64
!
interface FastEthernet0/0
  ipv6 address 2002:9601:606:1::6/64
```

Task 2.14**R3:**

```
interface Tunnel3456
  ipv6 address 2002:9601:303:3456::3/64
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel3456
```

R4:

```
interface Tunnel3456
  ipv6 address 2002:9601:404:3456::4/64
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel3456
```

R5:

```
interface Tunnel3456
  ipv6 address 2002:9601:505:3456::5/64
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel3456
```

R6:

```
interface Tunnel3456
  ipv6 address 2002:9601:606:3456::6/64
  tunnel source Loopback0
  tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel3456
```

Task 2.13 – 2.14 Verification

Verify the 6to4 tunneling by looking at the routing table.

```
Rack1R3#show ipv6 route static
```

```
IPv6 Routing Table - 9 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
```

```
S 2002::/16 [1/0]
```

```
via ::, Tunnel3456
```

Test connectivity:

```
Rack1R3#ping 2002:9601:404:3456::4
```

Type escape sequence to abort.


```
Sending 5, 100-byte ICMP Echos to 2002:9601:404:3456::4, timeout is 2
seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/76/80 ms
```

```
Rack1R3#ping 2002:9601:505:3456::5
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:9601:505:3456::5, timeout is 2
seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
```

```
Rack1R3#ping 2002:9601:606:3456::6
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:9601:606:3456::6, timeout is 2
seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/23/28 ms
```

```
Rack1R3#traceroute 2002:9601:606:3456::6
```

```
Type escape sequence to abort.
Tracing the route to 2002:9601:606:3456::6

  1 2002:9601:606:3456::6 20 msec 20 msec 24 msec
```

Task 2.15

```
R6:
ipv6 access-list PC_IN_VLAN6
permit ipv6 host 2002:9601:606:1:0209:6BFF:FE06:47EF any
!
line vty 0 4
  ipv6 access-class PC_IN_VLAN6 in
```

Task 2.15 Verification

Telnet to R6 from any other IPv6-enabled host:

```
Rack1R3#telnet 2002:9601:606::6
Trying 2002:9601:606::6 ...
% Connection refused by remote host
```

3. BGP

Task 3.1

R2:

```
router bgp 200
  neighbor 192.10.1.254 password CISCO
  neighbor 150.1.3.3 password CISCO
```

R3:

```
router bgp 200
  neighbor 192.10.1.254 password CISCO
  neighbor 150.1.2.2 password CISCO
```

SW2:

```
router bgp 100
  neighbor 150.1.1.1 route-reflector-client
  neighbor 150.1.6.6 route-reflector-client
```

R6:

```
router bgp 100
  neighbor 150.1.8.8 next-hop-self
```

Task 3.1 Breakdown

For BGP AS 100, there is not a full mesh configured. By default, routes learned from an iBGP neighbor will not be passed on to another iBGP peer. In order to adjust the behavior, we could configure a confederation or a route reflector. A route reflector will not require as much configuration, and we just need to configure SW2 to reflect the routes. Additionally, the network connecting R6 to BB1 is not advertised into the routing protocol. Since the next hop value is not changed when passing to an iBGP peer, SW2 will have difficulty with these updates, since SW2 does not have a route to the next hop, and will not install the routes in the routing table or pass them on to R1. The network connecting to BB1 will be added in section 3.4, so you can remove it when that section has been configured.

Task 3.1 Verification

Verify the BGP neighbors:

```
Rack1R6#show ip bgp summary | begin Neighbor
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
54.1.3.254	4	54	10	7	14	0	0	00:02:05	10
150.1.8.8	4	100	6	9	14	0	0	00:01:29	3

```
Rack1SW2#show ip bgp summary | begin Neighbor
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
150.1.1.1	4	100	5	5	4	0	0	00:00:20	3

```
150.1.6.6 4 100 8 5 4 0 0 00:00:04 10
```

```
Rack1R1#show ip bgp summary | begin Neighbor
```

```
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
148.1.0.2 4 200 7 6 4 0 0 00:02:40 3
148.1.0.4 4 400 5 6 4 0 0 00:01:44 0
150.1.8.8 4 100 5 5 4 0 0 00:00:58 10
```

```
Rack1R2#show ip bgp summary | begin Neighbor
```

```
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
148.1.0.1 4 100 7 8 4 0 0 00:03:05 10
150.1.3.3 4 200 6 7 4 0 0 00:02:47 3
192.10.1.254 4 254 7 6 4 0 0 00:02:30 3
```

```
Rack1R3#show ip bgp summary | begin Neighbor
```

```
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
148.1.35.5 4 65057 5 6 4 0 0 00:02:58 0
150.1.2.2 4 200 8 7 4 0 0 00:03:28 13
192.10.1.254 4 254 8 8 4 0 0 00:03:12 3
```

```
Rack1R5#show ip bgp summary | begin Neighbor
```

```
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
148.1.35.3 4 200 7 6 4 0 0 00:03:51 13
148.1.57.7 4 65057 6 7 4 0 0 00:03:28 0
```

Task 3.2

R6:

```
router bgp 100
 neighbor 54.1.3.254 filter-list 25 in
 !
 ip as-path access-list 25 permit ^54(_[0-9]+)?$
```

Task 3.2 Breakdown

A common view to take of the BGP table is the directly connected AS's customer routes. Taking this type of view conserves memory since the router does not have to store the entire BGP table.

Recall that the characters used in regular expressions:

Character	Meaning
^	Start of string
\$	End of string
[]	Range of characters
-	Used to specify range (i.e. [0-9])
()	Logical grouping
.	Any single character
*	Zero or more instances

+	One or more instance
?	Zero or one instance
_ (underscore)	Comma, open or close brace, open or close parentheses, start or end of string, or space

The goal of the regular expression used in the above task is to match routes originated in AS 54, or routes originated in AS 54's customer's networks. Therefore, the possible AS paths to match are either "54" or "54 X", where "X" is any single AS. First, let us match just routes originated in AS 54:

```
ip as-path access-list 25 permit ^54$
```

This means:

Character(s)	Meaning
^	Start of line
54	Exactly AS 54
\$	End of line

Now, we need to check for the case "54 X" where X is any single AS:

```
ip as-path access-list 25 permit ^54_[0-9]+$
```

This means:

Character(s)	Meaning
^	Start of line
54	Exactly AS 54
	Space
[0-9]	Any number 0 through 9
+	One or more instance of 0 to 9
\$	End of line

Comparing the above two expressions, it is evident that the difference between them is the sequence "[0-9]+"

Therefore, these expressions can be combined by checking for zero or one instance (true or false) of this sequence. Zero or one instances is the character ? Remember that to match the question mark in the line the escape sequence CTRL-V must be issued first.

☑ Verification

```
[root@CoachZ /]#telnet route-server.net
##### route-server.xx.net #####
##### xx Route Monitor #####
```

This router maintains peerings with customer-facing routers throughout the xx Backbone:

<output deleted>

This router has the global routing table view from each of the above routers, providing a glimpse to the Internet routing table from the xx network's perspective.

```
Route-server>show ip bgp regexp ^701(_[0-9]+)?$
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i	208.172.146.30		100	0	701 10912 i
* i24.206.128.0/20	208.172.146.30		100	0	701 21615 i
*>i	208.172.146.29		100	0	701 21615 i
* i24.206.128.0/18	208.172.146.30		100	0	701 21615 i
*>i	208.172.146.29		100	0	701 21615 i
* i24.206.144.0/20	208.172.146.30		100	0	701 21615 i
*>i	208.172.146.29		100	0	701 21615 i
*>i24.204.42.0/24	208.172.146.29		100	0	701 i
* i	208.172.146.30		100	0	701 i
*>i24.204.43.0/24	208.172.146.29		100	0	701 i
* i	208.172.146.30		100	0	701 i
* i24.204.80.0/22	208.172.146.30		100	0	701 i
*>i	208.172.146.29		100	0	701 i
* i24.204.84.0/22	208.172.146.30		100	0	701 i
*>i	208.172.146.29		100	0	701 i
* i24.204.88.0/22	208.172.146.30		100	0	701 i
*>i	208.172.146.29		100	0	701 i
*>i24.204.136.0/22	208.172.146.29		100	0	701 10912 i
* i	208.172.146.30		100	0	701 10912 i
*>i24.204.139.0/24	208.172.146.29		100	0	701 10912 i
* i24.206.160.0/19	208.172.146.30		100	0	701 21615 i
*>i	208.172.146.29		100	0	701 21615 i
*>i24.214.128.0/18	208.172.146.29		100	0	701 12083 ?
* i	208.172.146.30		100	0	701 12083 ?
*>i24.214.192.0/18	208.172.146.29		100	0	701 12083 ?
* i	208.172.146.30		100	0	701 12083 ?
* i24.215.64.0/18	208.172.146.30		100	0	701 11260 i
*>i	208.172.146.29		100	0	701 11260 i
*>i24.216.0.0/16	208.172.146.29		100	0	701 i

Task 3.2 Verification

Before applying the solutions try the regexp:

"Clean" BGP table:

```
Rack1R6#show ip bgp | begin Network
  Network          Next Hop          Metric LocPrf Weight Path
*> 28.119.16.0/24  54.1.3.254              0 54 i
*> 28.119.17.0/24  54.1.3.254              0 54 i
*> 112.0.0.0       54.1.3.254              0 54 50 60 i
*> 113.0.0.0       54.1.3.254              0 54 50 60 i
*> 114.0.0.0       54.1.3.254              0 54 i
*> 115.0.0.0       54.1.3.254              0 54 i
*> 116.0.0.0       54.1.3.254              0 54 i
*> 117.0.0.0       54.1.3.254              0 54 i
*> 118.0.0.0       54.1.3.254              0 54 i
*> 119.0.0.0       54.1.3.254              0 54 i
*>i205.90.31.0    148.1.0.2              0 100 0 200 254 ?
*>i220.20.3.0     148.1.0.2              0 100 0 200 254 ?
*>i222.22.2.0     148.1.0.2              0 100 0 200 254 ?
```

Apply the regexp:

```
Rack1R6#show ip bgp quote-regexp ^54(_[0-9]+)?$ | begin Network
  Network          Next Hop          Metric LocPrf Weight Path
*> 28.119.16.0/24  54.1.3.254              0 54 i
*> 28.119.17.0/24  54.1.3.254              0 54 i
*> 114.0.0.0       54.1.3.254              0 54 i
*> 115.0.0.0       54.1.3.254              0 54 i
*> 116.0.0.0       54.1.3.254              0 54 i
*> 117.0.0.0       54.1.3.254              0 54 i
*> 118.0.0.0       54.1.3.254              0 54 i
*> 119.0.0.0       54.1.3.254              0 54 i
```

Modify regexp slightly for verification:

```
Rack1R6#show ip bgp quote-regexp ^200(_[0-9]+)?$ | begin Network
  Network          Next Hop          Metric LocPrf Weight Path
*>i205.90.31.0    148.1.0.2              0 100 0 200 254 ?
*>i220.20.3.0     148.1.0.2              0 100 0 200 254 ?
*>i222.22.2.0     148.1.0.2              0 100 0 200 254 ?
```

Task 3.3

R4:

```
router bgp 400
 redistribute connected route-map CONNECTED->BGP
 auto-summary
!
interface Fa0/0
 ip address 10.1.4.4 255.255.255.0
!
route-map CONNECTED->BGP permit 10
 match interface Fa0/0
```

Task 3.3 Breakdown

The **auto-summary** keyword in BGP only affects prefixes that were redistributed into BGP. When it is on, routes redistributed into the BGP domain are automatically summarized to the classful boundary. When auto-summary is disabled, subnets will retain their subnet mask information.

Task 3.3 Verification

Verify the prefix origination:

```
Rack1R4#show ip bgp
```

```
BGP table version is 9, local router ID is 150.1.4.4
```

```
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
*> 10.0.0.0	0.0.0.0	0		32768	?
*> 205.90.31.0	148.1.0.2			0 100 200 254	?
*> 220.20.3.0	148.1.0.2			0 100 200 254	?
*> 222.22.2.0	148.1.0.2			0 100 200 254	?

Task 3.4

R6:

```
router bgp 100
 network 54.1.3.0 mask 255.255.255.0
 network 148.1.6.0 mask 255.255.255.0
 no neighbor 150.1.8.8 next-hop-self
 aggregate-address 148.1.0.0 255.255.128.0 suppress-map SUPPRESS_MAP
!
route-map SUPPRESS_MAP permit 10
```

Task 3.4 Breakdown

The **summary-only** keyword suppresses all subnets of an aggregate in order to prevent them from being advertised. The **suppress-map** is a route-map that can selectively suppress one or more subnets of the aggregate.

Typically, this configuration is used when the amount of prefixes that should be suppressed is outweighed by the amount that should not be suppressed. Unlike an **unsuppress-map**, the **suppress-map** is configured on the aggregate itself, and affects the local BGP table. The **unsuppress-map** is applied on a per neighbor basis.

The above configuration uses an **unsuppress-map** that does not match anything. This is effectively an explicit permit statement for the route-map, and will suppress all subnets of the aggregate.

Another way to solve this task would have been to simply filter the subnets from being advertised to BB1 by using an access-list or prefix-list.

Task 3.4 Verification

Verify the summary prefix generation. Confirm that the specific prefix has been suppressed.

```
Rack1R6#show ip bgp | begin Network
  Network          Next Hop          Metric LocPrf Weight Path
*>i10.0.0.0        148.1.0.4         0      100      0 400 ?
*> 28.119.16.0/24  54.1.3.254        0              0 54 i
*> 28.119.17.0/24  54.1.3.254        0              0 54 i
*> 54.1.3.0/24     0.0.0.0           0              32768 i
*> 114.0.0.0       54.1.3.254        0              0 54 i
*> 115.0.0.0       54.1.3.254        0              0 54 i
*> 116.0.0.0       54.1.3.254        0              0 54 i
*> 117.0.0.0       54.1.3.254        0              0 54 i
*> 118.0.0.0       54.1.3.254        0              0 54 i
*> 119.0.0.0       54.1.3.254        0              0 54 i
*> 148.1.0.0/17    0.0.0.0           0              32768 i
s> 148.1.6.0/24    0.0.0.0           0              32768 i
*>i205.90.31.0    148.1.0.2         0      100      0 200 254 ?
*>i220.20.3.0     148.1.0.2         0      100      0 200 254 ?
```



```
*>i222.22.2.0      148.1.0.2          0    100    0 200 254 ?
```

Task 3.5

SW1:

```
interface Loopback1
 ip address 148.1.177.7 255.255.255.0
!
router bgp 65057
 neighbor 148.1.57.5 send-community
 network 148.1.177.0 mask 255.255.255.0 route-map SET_COMM
!
route-map SET_COMM permit 10
 set community no-export
```

Task 3.5 Breakdown

The above task illustrates the usage of the **route-map** keyword on the **network** statement. This route-map can be used to modify attributes of the prefix as it is originated into the BGP domain. These attributes may include weight, local-preference, MED, and community.

Previous Reference

BGP Communities: Lab 3

Task 3.5 Verification

Confirm that R5 does not advertise the prefix to any eBGP peer:

```
Rack1R5#show ip bgp 148.1.177.0
BGP routing table entry for 148.1.177.0/24, version 62
Paths: (1 available, best #1, table Default-IP-Routing-Table, not
advertised to EBGp peer)
Flag: 0x820
    Not advertised to any peer
    Local
      148.1.57.7 from 148.1.57.7 (150.1.7.7)
        Origin IGP, metric 0, localpref 100, valid, internal, best
        Community: no-export
```

4. IP and IOS Features

Task 4.1

R6:

```
ip ftp username R6CORE
ip ftp password CISCO
exception dump 148.1.3.100
exception protocol ftp
exception core-file R6DUMP.txt
```

Task 4.1

When a router crashes, it can be possible to create a core dump of what is currently in RAM. This output can be useful for TAC to troubleshoot a hardware or software problem. Core dumps can be created through TFTP, FTP, RCP, or direct to flash.



Further Reading

[Creating Core Dumps](#)



Note

Any time you need to create a core dump for a device that has over 32megs of RAM, it is recommended to use FTP over TFTP. Many TFTP applications have issues with file sizes larger than 32meg.

Task 4.2

R5:

```
ntp server 204.12.1.254
ntp peer 148.1.57.7
ntp master 5
```

SW1:

```
ntp server 204.12.1.254
```

Task 4.2 Verification

Verify NTP status and associations:

```
Rack1R5#show ntp associations
```

	address	ref clock	st	when	poll	reach	delay	offset	disp
+~	127.127.7.1	127.127.7.1	4	7	64	377	0.0	0.00	0.0
+~	148.1.57.7	204.12.1.254	5	39	64	377	2.9	-2.49	2.2
*~	204.12.1.254	127.127.7.1	4	62	64	37	7.5	0.70	876.3

* master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1R5#show ntp status

Clock is synchronized, stratum 5, reference is 204.12.1.254
 nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
 reference time is C83A20EA.5AFD0A73 (05:45:14.355 UTC Wed Jun 14 2006)
 clock offset is 0.0000 msec, root delay is 6.87 msec
 root dispersion is 378.75 msec, peer dispersion is 377.79 msec

Rack1SW1#show ntp associations

	address	ref clock	st	when	poll	reach	delay	offset	disp
+	148.1.57.5	204.12.1.254	5	4	64	7	0.8	3.52	7876.7
*~	204.12.1.254	127.127.7.1	4	3	64	377	7.4	6.35	5.3

* master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1SW1#show ntp status

Clock is synchronized, stratum 5, reference is 204.12.1.254
 nominal freq is 250.0000 Hz, actual freq is 249.9998 Hz, precision is 2**18
 reference time is C83A20F2.E8E44FFD (05:45:22.909 UTC Wed Jun 14 2006)
 clock offset is 6.3512 msec, root delay is 7.40 msec
 root dispersion is 13.78 msec, peer dispersion is 7.40 msec

Task 4.3

R5:

```
ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 204.12.1.254 key 1 prefer
```

SW1:

```
ntp authentication-key 1 md5 CISCO
ntp authenticate
ntp trusted-key 1
ntp server 204.12.1.254 key 1 prefer
```

Task 4.3 Verification

Verify NTP authentication:

Rack1SW1#show ntp associations detail

204.12.1.254 configured,authenticated,our_master,sane, valid, stratum 4
 ref ID 127.127.7.1, time C83A2190.82116150 (05:48:00.508 UTC Wed Jun 14 2006)
 our mode client,peer mode server, our poll intvl 64, peer poll intvl 64
 root delay 0.00 msec, root disp 0.03, reach 377, sync dist 4.959
 delay 5.57 msec, offset 9.4310 msec, dispersion 2.15
 precision 2**19, version 3
 org time C83A21B2.EB5345F8 (05:48:34.919 UTC Wed Jun 14 2006)
 rcv time C83A21B2.E99FB4B5 (05:48:34.912 UTC Wed Jun 14 2006)
 xmt time C83A21B2.E7F278EE (05:48:34.906 UTC Wed Jun 14 2006)

```

filtdelay = 5.57    5.98    6.99    7.40    7.87    6.04    5.81    5.89
filtoffset =9.43    8.71    7.74    6.35    3.80    0.24    0.09    0.09
filterror = 0.02    0.99    1.97    2.94    3.92    4.90    4.91    4.93

```

```

Rack1R5#show ntp associations detail | begin 204.12.1.254 config
204.12.1.254 configured,authenticated,our_master,sane, valid, stratum 4
ref ID 127.127.7.1, time C83A21D0.8268E8C0 (05:49:04.509 UTC Wed Jun 14
2006)
our mode client, peer mode server, our poll intvl 64,peer poll intvl 64
root delay 0.00 msec, root disp 0.03, reach 377, sync dist 5.737
delay 9.26 msec, offset 3.7436 msec, dispersion 0.99
precision 2**19, version 3
org time C83A21DB.5BB3F3DE (05:49:15.358 UTC Wed Jun 14 2006)
rcv time C83A21DB.5BEE48E6 (05:49:15.359 UTC Wed Jun 14 2006)
xmt time C83A21DB.59558521 (05:49:15.348 UTC Wed Jun 14 2006)
filtdelay = 9.26  8.65    7.48    8.24    6.87    7.48    6.96    6.88
filtoffset =3.74  3.49    2.56    2.26    0.94    0.70    -0.65    -0.71
filterror = 0.02  0.76    1.74    2.72    3.69    4.67    5.65    5.66

```

Task 4.4

R1:

```
ip tcp synwait-time 5
```

Task 4.4 Breakdown

The TCP SYN wait time is the time the router will wait after sending a TCP SYN packet for a SYN/ACK to come back. If the SYN/ACK response has not been received before the timer expires, the connection is reset.

Task 4.5

R6:

```

interface Serial0/0
 ip accounting output-packets
!
ip accounting-threshold 1000

```

Task 4.5 Verification

Verify IP accounting:

```
Rack1SW2#ping 117.0.0.1
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 117.0.0.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

```
Rack1SW2#ping 118.0.0.1
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 118.0.0.1, timeout is 2 seconds:
```

```
!!!!
```

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
 Rack1SW2#**ping 119.0.0.1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 119.0.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms

Rack1R6#**show ip accounting**

Source	Destination	Packets	Bytes
148.1.68.8	117.0.0.1	5	500
148.1.68.8	118.0.0.1	5	500
148.1.68.8	119.0.0.1	5	500

Task 4.6

R3:

```
interface FastEthernet0/0
 ip nat inside
!
interface FastEthernet0/1
 ip nat outside
!
interface Serial1/0.302 point-to-point
 ip nat outside
!
interface Serial1/1
 ip nat outside
!
ip nat pool REAL_SERVERS 148.1.3.110 148.1.3.112 prefix-length 24 type
rotary
ip nat inside destination list OLD_WEB_SERVER pool REAL_SERVERS
!
ip access-list extended OLD_WEB_SERVER
 permit tcp any host 148.1.3.100 eq www
 permit tcp any host 148.1.3.100 eq 8080
 permit tcp any host 148.1.3.100 eq 443
```

Task 4.6 Verification

Rack1R1#**telnet 148.1.3.100 80**

Trying 148.1.3.100, 80 ...

Rack1AS>3

[Resuming connection 3 to r3 ...]

Rack1R3#**show ip nat translations**

Pro	Inside global	Inside local	Outside local	Outside global
tcp	148.1.3.100:80	148.1.3.110:80	148.1.0.1:11004	148.1.0.1:11004

Rack1AS>6

[Resuming connection 6 to r6 ...]

Rack1R6#**telnet 148.1.3.100 80**

Trying 148.1.3.100, 80 ...

Rack1AS>3

```
[Resuming connection 3 to r3 ... ]
```

```
Rack1R3#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	148.1.3.100:80	148.1.3.110:80	148.1.0.1:11004	148.1.0.1:11004
tcp	148.1.3.100:80	148.1.3.111:80	148.1.68.6:21387	148.1.68.6:21387

```
Rack1AS>8
```

```
[Resuming connection 8 to SW2 ... ]
```

```
Rack1SW2#telnet 148.1.3.100 80
```

```
Trying 148.1.3.100, 80 ...
```

```
Rack1AS>3
```

```
[Resuming connection 3 to r3 ... ]
```

```
Rack1R3#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
tcp	148.1.3.100:80	148.1.3.110:80	148.1.0.1:11004	148.1.0.1:11004
tcp	148.1.3.100:80	148.1.3.111:80	148.1.68.6:21387	148.1.68.6:21387
tcp	148.1.3.100:80	148.1.3.112:80	148.1.18.8:11000	148.1.18.8:11000

5. IP Multicast

Task 5.1

R1 and R2:

```
ip multicast-routing
!
interface FastEthernet0/0
 ip pim dense-mode
!
interface Serial0/0
 ip pim dense-mode
```

R3:

```
ip multicast-routing
!
interface FastEthernet0/0
 ip pim dense-mode
!
interface Serial1/0.302
 ip pim dense-mode
```

R6:

```
ip multicast-routing
!
interface FastEthernet0/0
 ip pim dense-mode
!
interface FastEthernet0/1
 ip pim dense-mode
```

SW2:

```
ip multicast-routing distributed
```

```

!
interface Vlan 18
 ip pim dense-mode
!
interface Vlan 68
 ip pim dense-mode

```

Task 5.1 Verification

Verify the PIM neighbors:

```
Rack1R2#show ip pim neighbor
```

```

PIM Neighbor Table
Neighbor      Interface      Uptime/Expires  Ver  DR
Address
148.1.0.3     Serial0/0      00:01:28/00:01:16 v2   1 / DR S
148.1.0.1     Serial0/0      00:02:32/00:01:39 v2   1 / S

```

```
Rack1SW2#show ip pim neighbor
```

```

PIM Neighbor Table
Neighbor      Interface      Uptime/Expires  Ver  DR
Address
148.1.18.1    Vlan18         00:00:44/00:01:29 v2   1 / S
148.1.68.6    Vlan68         00:01:13/00:01:30 v2   1 / S

```

Verify the PIM interfaces:

```
Rack1R3#show ip pim interface
```

```

Address      Interface      Ver/   Nbr   Query  DR     DR
Mode        Count  Intvl  Prior
148.1.3.3    FastEthernet0/0 v2/D   0     30     1
148.1.3.3
148.1.0.3    Serial1/0.302  v2/D   1     30     1     0.0.0.0

```

```
Rack1R2#show ip pim interface
```

```

Address      Interface      Ver/   Nbr   Query  DR     DR
Mode        Count  Intvl  Prior
192.10.1.2   FastEthernet0/0 v2/D   0     30     1     192.10.1.2
148.1.0.2    Serial0/0      v2/D   2     30     1     148.1.0.3

```

```
Rack1R1#show ip pim interface
```

```

Address      Interface      Ver/   Nbr   Query  DR     DR
Mode        Count  Intvl  Prior
148.1.18.1   FastEthernet0/ v2/D   1     30     1     148.1.18.8
148.1.0.1    Serial0/0      v2/D   1     30     1     148.1.0.2

```

```
Rack1SW2#show ip pim interface
```

```

Address      Interface      Ver/   Nbr   Query  DR     DR
Mode        Count  Intvl  Prior
148.1.18.8   Vlan18         v2/D   1     30     1     148.1.18.8

```

```
148.1.68.8   Vlan68                v2/D   1       30      1       148.1.68.8
```

```
Rack1R6#show ip pim interface
```

Address	Interface	Ver/ Mode	Nbr Count	Query Intvl	DR Prior	DR
148.1.6.6	FastEthernet0/0	v2/D	0	30	1	148.1.6.6
148.1.68.6	FastEthernet0/1	v2/D	1	30	1	148.1.68.8

Task 5.2

R1:

```
interface Tunnel0
 ip address 148.1.13.1 255.255.255.0
 ip pim dense-mode
 tunnel source Loopback0
 tunnel destination 150.1.3.3
!
ip mroute 0.0.0.0 0.0.0.0 Tunnel0
```

R3:

```
interface Tunnel0
 ip address 148.1.13.3 255.255.255.0
 ip pim dense-mode
 tunnel source Loopback0
 tunnel destination 150.1.1.1
```

R6:

```
interface FastEthernet0/0
 ip igmp join-group 224.6.6.6
```

Task 5.2 Verification

Verify that R6 may now receive the multicast feed:

```
Rack1R3#ping 224.6.6.6 source fa0/0 repeat 5
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 224.6.6.6, timeout is 2 seconds:

```
Reply to request 0 from 148.1.68.6, 16 ms
Reply to request 1 from 148.1.68.6, 8 ms
Reply to request 2 from 148.1.68.6, 8 ms
Reply to request 3 from 148.1.68.6, 8 ms
Reply to request 4 from 148.1.68.6, 8 ms
```

Verify the multicast routing tables:

```
Rack1R1#show ip mroute
```

```
IP Multicast Routing Table
<output omitted>
```

```
(*, 224.6.6.6), 00:01:14/stopped, RP 0.0.0.0, flags: D
 Incoming interface: Null, RPF nbr 0.0.0.0
 Outgoing interface list:
  Tunnel0, Forward/Dense, 00:01:14/00:00:00
```



```

Serial0/0, Forward/Dense, 00:01:14/00:00:00
FastEthernet0/0, Forward/Dense, 00:01:14/00:00:00

(148.1.13.3, 224.6.6.6), 00:01:14/00:02:01, flags: T
Incoming interface: Tunnel0, RPF nbr 148.1.13.3
Outgoing interface list:
FastEthernet0/0, Forward/Dense, 00:01:16/00:00:00
Serial0/0, Forward/Dense, 00:01:16/00:00:00

(192.10.1.3, 224.6.6.6), 00:01:16/00:01:44, flags:
Incoming interface: Tunnel0, RPF nbr 148.1.13.3, Mroute
Outgoing list:
FastEthernet0/0, Forward/Dense, 00:01:16/00:00:00
Serial0/0, Forward/Dense, 00:01:16/00:00:00

(*, 224.0.1.40), 03:46:58/00:02:17, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel0, Forward/Dense, 00:02:27/00:00:00
Serial0/0, Forward/Dense, 03:46:52/00:00:00
FastEthernet0/0, Forward/Dense, 03:46:58/00:00:00

```

Rack1R3#show ip mroute

IP Multicast Routing Table

<output omitted>

```

(*, 224.6.6.6), 00:00:54/stopped, RP 0.0.0.0, flags: D
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Serial1/0.302, Forward/Dense, 00:00:54/00:00:00
Tunnel0, Forward/Dense, 00:00:54/00:00:00

(192.10.1.3, 224.6.6.6), 00:00:54/00:02:13, flags:
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Serial1/0.302, Forward/Dense, 00:00:55/00:00:00
Tunnel0, Forward/Dense, 00:00:55/00:00:00

(*, 224.0.1.40), 03:45:24/00:02:50, RP 0.0.0.0, flags: DCL
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Serial1/0.302, Forward/Dense, 03:45:23/00:00:00
Tunnel0, Forward/Dense, 00:02:03/00:00:00
FastEthernet0/0, Forward/Dense, 03:45:24/00:00:00

```

Task 5.3**R2:**

ip multicast rpf backoff 300 5000

Task 5.3 Verification*Verify the backoff timers:***Rack1R2#show ip rpf events**

Last 15 triggered multicast RPF check events

```
RPF backoff delay: 300 msec
RPF maximum delay: 5 sec
```

DATE/TIME	BACKOFF	PROTOCOL	EVENT	RPF CHANGES
Mar 1 09:15:40.833	500 msec	RIP	Route UP	0
Mar 1 09:15:18.829	500 msec	RIP	Route UP	0
Mar 1 05:31:12.802	500 msec	Connected	Route UP	0

6. QoS

Task 6.1

```
R6:
policy-map SHAPE
  class class-default
    shape average 5000000 160000 80000
!
interface Serial0/0
  bandwidth 45000
  service-policy output SHAPE
```

Task 6.1 Verification

Verify MQC configuration:

```
Rack1R6#show policy-map interface s0/0
```

```
Serial0/0
```

```
Service-policy output: SHAPE
```

```
Class-map: class-default (match-any)
  1 packets, 13 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
Traffic Shaping
  Target/Average  Byte  Sustain  Excess  Interval  Increment
  Rate           Limit bits/int bits/int (ms)      (bytes)
  5000000/5000000 30000 160000   80000   32        20000

  Adapt Queue  Packets  Bytes  Packets  Bytes  Shaping
  Active Depth          Bytes  Delayed Delayed Active
  -      0            0      0      0      0      no
```

Task 6.1 Breakdown

For a rate of 5 Mbps, 5 kilobits are sent every millisecond. For a time interval of 32 ms, the corresponding number of bits is 160,000 bits. For an excess burst of 50% of the rate, the excess burst value will be 80,000 bits.

Task 6.2

```

R6:
ip cef
!
class-map match-any PEER-TO-PEER
  match protocol fasttrack
  match protocol gnutella
  match protocol kazaa2
!
policy-map QOS_POLICY
  class PEER-TO-PEER
    police cir 8000
!
policy-map SHAPE
  class class-default
    service-policy QOS_POLICY

```



Further Reading

[Network Based Application Recognition \(NBAR\)](#)

Task 6.2 Verification

Verify the new MQC configuration:

```
Rack1R6#show policy-map interface s0/0/0
```

```
Serial0/0
```

```
Service-policy output: SHAPE
```

```

Class-map: class-default (match-any)
 36 packets, 811 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
Match: any
Traffic Shaping
  Target/Average Byte  Sustain   Excess   Interval  Increment
  Rate                Limit bits/int bits/int (ms)      (bytes)
  5000000/5000000 30000 160000   80000    32        20000

Adapt Queue   Packets  Bytes   Packets  Bytes  Shaping
Active Depth                Delayed  Delayed  Active
-      0          8        447    0        0      no

```

```
Service-policy : QOS_POLICY
```

```

Class-map: PEER-TO-PEER (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol fasttrack
 0 packets, 0 bytes
 5 minute rate 0 bps
Match: protocol gnutella
 0 packets, 0 bytes

```

```

    5 minute rate 0 bps
Match: protocol kazaa2
    0 packets, 0 bytes
    5 minute rate 0 bps
police:
    cir 8000 bps, bc 1500 bytes
    conformed 0 packets, 0 bytes; actions:
        transmit
    exceeded 0 packets, 0 bytes; actions:
        drop
    conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)
    4 packets, 133 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
Match: any

```

Task 6.3

R6:

```

class-map match-all HTTP
    match protocol http
class-map match-all SMTP
    match protocol smtp
!
policy-map QOS_POLICY
    class HTTP
        bandwidth 2000
    class SMTP
        bandwidth 1000

```

Task 6.3 Verification

Confirm the MQC configuration changes:

```
Rack1R6#show policy-map interface s0/0 | begin HTTP
```

```

Class-map: HTTP (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol http
Queueing
    Output Queue: Conversation 137
    Bandwidth 2000 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0

Class-map: SMTP (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol smtp
Queueing
    Output Queue: Conversation 138
    Bandwidth 1000 (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)
  21 packets, 516 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any

```

Task 6.4

R6:

```

class-map match-all #CISCO_UNDERNET_IRC
  match access-group name CISCO_UNDERNET_IRC
!
policy-map QOS_POLICY
  class #CISCO_UNDERNET_IRC
    priority 32
!
ip access-list extended CISCO_UNDERNET_IRC
  permit tcp host 148.1.6.10 any eq 6667

```

Task 6.4 Verification

Veriy MQC priority configuration:

```
Rack1R6#show policy-map interface s0/0 | begin UNDERNET
```

```

Class-map: #CISCO_UNDERNET_IRC (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group name CISCO_UNDERNET_IRC
  Queueing
    Strict Priority
    Output Queue: Conversation 136
    Bandwidth 32 (Kbps) Burst 800 (Bytes)
    (pkts matched/bytes matched) 0/0
    (total drops/bytes drops) 0/0

Class-map: class-default (match-any)
  39 packets, 850 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any

```

7. Security

Task 7.1

```
R6:
interface Serial0/0
 ip access-group RFC1918 in
!
ip access-list extended RFC1918
 deny ip 10.0.0.0 0.255.255.255 any
 deny ip 172.16.0.0 0.15.255.255 any
 deny ip 192.168.0.0 0.0.255.255 any
 permit ip any any
```

Task 7.1 Breakdown

RFC 1918 specifies that the address ranges 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 are reserved for private usage. As these addresses are not publicly routable on the Internet, there is no valid reason for traffic to be received on an outside interface which is sourced from a host in this range. Therefore, best security practice dictates that these addresses should be denied from entering the network.



Standard

[RFC 1918: Address Allocation for Private Internets](#)

Task 7.2

```
R5:
ip access-list extended DENY_INTERNET_SURFING
 permit ip any any time-range NON_WORK_HOURS
 permit tcp any host 148.1.3.100 eq www 443
!
time-range NON_WORK_HOURS
 periodic weekend 0:00 to 23:59
 periodic weekdays 0:00 to 8:59
 periodic weekdays 17:00 to 23:59
!
interface FastEthernet0/1
 ip access-group DENY_INTERNET_SURFING in
```

Task 7.2 Breakdown

Newer IOS versions allow multiple non-consecutive ports to be matched on a single ACL line. Here, both HTTP and HTTPS are matched in the same line. Additionally, a second ACL entry is needed to match the time-range and allow during the non-work hours.