Task 1.1

SW1:

```
lacp system-priority 1
!
interface FastEthernet0/19
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode active
!
interface FastEthernet0/20
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode active
!
interface FastEthernet0/21
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode active
!
interface Port-channel3
 switchport mode dynamic desirable
```

SW4:

```
interface FastEthernet0/13
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode passive
!
interface FastEthernet0/14
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode passive
!
interface FastEthernet0/15
 no shutdown
 switchport mode dynamic desirable
 channel-group 3 mode passive
!
interface Port-channel3
 switchport mode dynamic desirable
```

Task 1.1 Verification

Check the port-channel status:

```
Rack1SW1#show etherchannel 3 summary
<output omitted>
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------
3      Po3(SU)         LACP      Fa0/19(P)   Fa0/20(P)   Fa0/21(P)

Rack1SW4#show etherchannel 3 summary
<output omitted>
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------
3      Po3(SU)         LACP      Fa0/13(P)   Fa0/14(P)   Fa0/15(P)
```

Verify the trunk:

```
Rack1SW1#show interface po3 trunk

Port        Mode         Encapsulation  Status        Native vlan
Po3         desirable    n-isl          trunking      1

Port        Vlans allowed on trunk
Po3         1-4094

Port        Vlans allowed and active in management domain
Po3         1,3,5-6,8,10,12-13,26,33,52,255,783

Port        Vlans in spanning tree forwarding state and not pruned
Po3         1,3,5-6,8,10,12-13,26,33,52,255,783

Rack1SW4#show interface po3 trunk

Port        Mode         Encapsulation  Status        Native vlan
Po3         desirable    n-isl          trunking      1

Port      Vlans allowed on trunk
Po3         1-4094

Port        Vlans allowed and active in management domain
Po3         1,3,5-6,8,10,12-13,26,33,52,255,783

Port        Vlans in spanning tree forwarding state and not pruned
Po3         1,3,5-6,8,10,12-13,26,33,52,255,783
Rack1SW4#
```

Verify the dot1q LACP priority:

```
Rack1SW1#show lacp sys-id
1, 0019.55e6.6580
```

Task 1.2

```
SW1:
aaa new-model
aaa authentication login default none
aaa authentication dot1x default group radius
!
dot1x system-auth-control
!
interface FastEthernet0/9
 switchport mode access
 dot1x port-control auto
!
interface FastEthernet0/10
 switchport mode access
 dot1x port-control auto
!
ip radius source-interface Loopback0
!
radius-server host 204.12.1.100
radius-server key CISCO
```

Task 1.2 Breakdown

In order to provide added security at the access layer of the network, 802.1x defines username and password based authentication for Ethernet switches. To enable 802.1x authentication, first issue the global configuration command **`dot1x system-auth-control`** (prior to 12.1(14)EA1 this command is not required). Next, enable dot1x must be enabled on a per interface basis by issuing the interface level command **`dot1x port-control [mode],`** where *mode* is either auto, forced-authorized, or forced-unauthorized. Forced-authorized is the default mode, and indicated that authorization is not required for access into the network. Forced-unauthorized is the opposite, and dictates that clients can never access the network through this port. When the state is set to auto, dot1x is enabled for username and password authentication.

In order to centrally manage users, dot1x integrates with Authentication Authorization and Accounting (AAA) to offload username and password databases to either TACACS or RADIUS. Therefore, to enable dot1x authentication, AAA must be enabled. The first step in enabling AAA is to issue the global command **`aaa new-model`**. This command starts the AAA process. Next, either the TACACS or RADIUS server should be defined, along with its corresponding key value. This is accomplished with the **`radius-server`** or **`tacacs-server`** global configuration command. Additionally, since network devices typically have multiple interfaces running IP, it is common practice to force the router/switch to generate radius or tacacs packets from a single interface instead of relying on what the routing table dictates the outgoing

interface to be. This is accomplished with the **ip [tacacs | radius] source-interface** command.

After AAA is enabled, the authentication policy must be defined. This is accomplished by issuing the **aaa authentication dot1x** command. In the above case, the *default* group is used. The default group applies to all interfaces and lines of the device in question.

Task 1.2 Verification

Verify dot1x port control:

```
Rack1SW1#show dot1x
Sysauthcontrol                  = Enabled
Supplicant Allowed In Guest Vlan  = Disabled
Dot1x Protocol Version          = 1

Rack1SW1#show dot1x all
Dot1x Info for interface FastEthernet0/9
<output omitted>
HostMode          = Single
PortControl       = Auto
ControlDirection  = Both
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
<output omitted>

Dot1x Info for interface FastEthernet0/10
--------------------------------------------------
<output omitted>
HostMode          = Single
PortControl       = Auto
ControlDirection  = Both
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
<output omitted>
```

Check to see if RADIUS is configured:

```
Rack1SW1#show aaa servers

RADIUS: id 1, priority 1, host 204.12.1.100, auth-port 1645, acct-port
1646
     State: current UP, duration 3634s, previous duration 0s
```


Task 1.3

SW1 and SW2:

```
sdm prefer routing
```

> **Note**
> After altering the Switch Database Template (SDM) a reload is required before the new template will take effect.

Task 1.3 Breakdown

The Switch Database Template (SDM) is used to alter the default allocation of resources (unicast routes, MAC addresses, etc) for the 3550 and 3560 series switches. By default the 3560 will support 8,000 unicast routes (6,000 directly connected and 2,000 non-directly connected). Since the new company's network already has 4,000 routes, the SDM will need to be altered to prefer *routing* to allow SW1 and SW2 to contain over 4,000 non-directly connected routes in their routing tables.

Task 1.3 Verification

Default SDM:

```
Rack1SW1#show sdm prefer | begin unicast routes
  number of IPv4 unicast routes:                    8K
    number of directly-connected IPv4 hosts:        6K
    number of indirect IPv4 routes:                 2K
  number of IPv4 policy based routing aces:         0
  number of IPv4/MAC qos aces:                      512
  number of IPv4/MAC security aces:                 1K
```

After the SDM has been changed to prefer routing and reloaded:

```
Rack1SW1#show sdm prefer | begin unicast routes
  number of IPv4 unicast routes:                    11K
    number of directly-connected IPv4 hosts:        3K
    number of indirect IPv4 routes:                 8K
  number of IPv4 policy based routing aces:         512
  number of IPv4/MAC qos aces:                      512
  number of IPv4/MAC security aces:                 1K
```

Task 2.1

R1:
```
interface Serial0/0
 ip ospf network point-to-multipoint
!
interface FastEthernet0/0
 ip ospf authentication-key CISCO
!
router ospf 1
 area 17 authentication
```

R2:
```
interface Serial0/0
 ip ospf network point-to-multipoint
```

R3:
```
interface Serial1/0
 ip ospf network point-to-multipoint
!
```
R4:
```
interface Serial0/0/0
 ip ospf network point-to-multipoint
```

SW1:
```
ip routing
!
interface FastEthernet0/1
 ip ospf authentication-key CISCO
!
router ospf 1
 router-id 150.1.7.7
 area 17 authentication
 network 132.1.17.7 0.0.0.0 area 17
```

Task 2.1 Verification

Verify the OSPF neighbors. For instance on R1:

```
Rack1R1#show ip ospf neighbor

Neighbor ID  Pri   State       Dead Time   Address         Interface
150.1.4.4      0   FULL/  -    00:01:58    132.1.0.4       Serial0/0
150.1.3.3      0   FULL/  -    00:01:58    132.1.0.3       Serial0/0
150.1.2.2      0   FULL/  -    00:01:58    132.1.0.2       Serial0/0
```

Verify the area and network type of the interface:

```
Rack1R1#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 132.1.0.1/24, Area 0
  Process ID 1, Router ID 150.1.1.1, Network Type POINT_TO_MULTIPOINT,
Cost: 64
```

```
  <output omitted>
```

Verify that the OSPF adjacencies in area 17 are being authenticated:

```
Rack1R1#show ip ospf | begin Area 17
    Area 17
        Number of interfaces in this area is 1
        Area has simple password authentication
```

Check in the interface is configured for authentication:

```
Rack1R1#show ip ospf interface fa0/0 | inc auth
  Simple password authentication enabled
```

V*erify that the adjacency is up:*

```
Rack1R1#show ip ospf neighbor | inc 132.1.17.7
150.1.7.7    1   FULL/DR    00:00:32    132.1.17.7      FastEthernet0/0
```

Task 2.2

R2:

```
router eigrp 10
 network 132.1.26.2 0.0.0.0
 neighbor 132.1.26.6 FastEthernet0/0
```

R6:

```
router eigrp 10
 network 132.1.26.6 0.0.0.0
 neighbor 132.1.26.2 FastEthernet0/0.26
!
interface FastEthernet0/0.26
 encapsulation dot1Q 26
 ip address 132.1.26.6 255.255.255.0
 ip summary-address eigrp 10 200.0.0.0 255.255.252.0 5
```

Task 2.2 Verification

Verify that the EIGRP packets are being sent to the unicast address (protocol 88 is EIGRP):

```
Rack1R2#debug interface fa0/0
Rack1R2#debug ip packet detail
IP: s=132.1.26.6 (FastEthernet0/0), d=132.1.26.2 (FastEthernet0/0), len
60, rcvd 3, proto=88
IP: s=132.1.26.2 (local), d=132.1.26.6 (FastEthernet0/0), len 60,
sending, proto=88
Rack1R2#undebug all
Rack1R2#no debug interface fa0/0
```

Verify that we have formed the appropriate EIGRP adjacencies:

```
Rack1R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 10
```

```
H   Address        Interface        Hold Uptime   SRTT   RTO  Q  Seq Type
                                    (sec)         (ms)       Cnt Num
1   132.1.26.6     Fa0/0              14 13:42:44    1   200  0  25   S
0   132.1.23.3     Se0/1              14 13:43:08   43   258  0  61
```

Verify that the EIGRP summary is generated on R6:

```
Rack1R6#show ip route | include Null0
D    200.0.0.0/22 is a summary, 00:00:30, Null0
```

Check that the other EIGRP enabled routers see the summary:

```
Rack1R2#show ip route eigrp | include 200.0
D    200.0.0.0/22 [90/2300416] via 132.1.26.6,00:01:38, FastEthernet0/0
```

Task 2.3

SW1:

```
router rip
 offset-list EVEN_SECOND_OCTET in 16 Vlan783
!
ip access-list standard EVEN_SECOND_OCTET
 permit 0.0.0.0 255.254.255.255
```

Task 2.3 Breakdown

The least significant bit of a binary number determines whether the number is even or odd. If the least significant bit is not set the number must be even. If the least significant bit is set the number must be odd. This always holds true since all other places in the binary table are even numbers, and any combination of even numbers plus an odd number results in an odd number. Likewise any combination of even numbers results in an even number.

Place	128	64	32	16	8	4	2	1
Even	X	X	X	X	X	X	X	0
Odd	X	X	X	X	X	X	X	1

Where “X” is either 0 or 1.

Since only the least significant bit determines whether a number is even or odd it is the only bit that needs to be checked. Therefore the resulting wildcard mask is 254, or in binary as follows:

Place	128	64	32	16	8	4	2	1
Wildcard	1	1	1	1	1	1	1	0

Where “0” is check and “1” is ignore.

The most common way to filter off a routing prefix in a distance vector protocol is to use the **distribute-list** command. A distribute-list is a way to apply an access-list to routing protocol updates. A routing prefix may also be filtered out by poisoning the metric or distance of the route.

To change the metric of a distance vector prefix use the routing process level command **offset-list.** In RIP a metric of 16 is "infinite". When a prefix has a metric of 16 it is considered unreachable, and cannot be installed in the routing table. The first solution to this task adds a metric of 16 to the incoming prefixes, hence invalidating them.

The second solution is to use the distance command. A distance of 255 is infinite. Any prefix with a distance of 255 is considered unreachable, and cannot be installed in the routing table. To change the distance of a prefix use the **distance [distance] [neighbor] [wildcard] [access-list]** where *distance* is the desired distance, *neighbor* is the originating address of the prefix, *wildcard* is a wildcard mask used to check the *neighbor* field, and *access-list* is a standard access-list number.

Task 2.3 Verification

Verify that the RIP networks with an even second octet are being filtered:

```
Rack1SW1#debug ip rip
<output omitted>
RIP: received v2 update from 204.12.1.254 on Vlan783
      30.0.0.0/16 via 0.0.0.0 in 17 hops  (inaccessible)
      30.1.0.0/16 via 0.0.0.0 in 1 hops
      30.2.0.0/16 via 0.0.0.0 in 17 hops  (inaccessible)
      30.3.0.0/16 via 0.0.0.0 in 1 hops
      31.0.0.0/16 via 0.0.0.0 in 17 hops  (inaccessible)
      31.1.0.0/16 via 0.0.0.0 in 1 hops
      31.2.0.0/16 via 0.0.0.0 in 17 hops  (inaccessible)
      31.3.0.0/16
```

Task 2.4

SW1:

```
router ospf 1
 redistribute rip subnets
 network 132.1.17.7 0.0.0.0 area 17
!
router rip
 redistribute ospf 1 metric 1
 distance 109
```

R2:

```
router eigrp 10
 redistribute ospf 1 metric 1 1 1 1 1
!
router ospf 1
 redistribute eigrp 10 subnets metric 20
 distance ospf external 171
 distance 110 0.0.0.0 255.255.255.255 EXTERNAL_VIA_OSPF
!
ip access-list standard EXTERNAL_VIA_OSPF
 remark == External prefixes that should be reachable via OSPF
 permit 132.1.8.0
 permit 150.1.7.0
 permit 150.1.8.0
 permit 204.12.1.0
 permit 31.0.0.0 0.255.255.255
 permit 30.0.0.0 0.255.255.255
```

R3:

```
router eigrp 10
 redistribute ospf 1 metric 1 1 1 1 1
!
router ospf 1
 redistribute eigrp 10 subnets metric 30
 distance ospf external 171
 distance 110 0.0.0.0 255.255.255.255 EXTERNAL_VIA_OSPF
!
ip access-list standard EXTERNAL_VIA_OSPF
 remark == External prefixes that should be reachable via OSPF
 permit 132.1.8.0
 permit 150.1.7.0
 permit 150.1.8.0
 remark == VLAN6 is here for multicast & PBR sections
 permit 132.1.6.0
 permit 204.12.1.0
 permit 31.0.0.0 0.255.255.255
 permit 30.0.0.0 0.255.255.255
```

☞ Note
The metric values used for redistribution are arbitrary. If the lab doesn't specify or imply a certain value should be used, then any value can be used.

R4:

```
router eigrp 10
 redistribute ospf 1 metric 1 1 1 1 1
!
router ospf 1
 redistribute eigrp 10 subnets metric 40
 distance ospf external 171
 distance 110 0.0.0.0 255.255.255.255 EXTERNAL_VIA_OSPF
!
ip access-list standard EXTERNAL_VIA_OSPF
 remark == External prefixes that should be reachable via OSPF
 permit 132.1.8.0
 permit 150.1.7.0
 permit 150.1.8.0
 permit 204.12.1.0
 permit 31.0.0.0 0.255.255.255
 permit 30.0.0.0 0.255.255.255
```

Task 2.4 Verification

Verify that redistribution is active:

```
Rack1SW1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  <output omitted>
  It is an autonomous system boundary router
  Redistributing External Routes from,
    rip, includes subnets in redistribution
<output omitted>
Routing Protocol is "rip"
  <output omitted>
  Redistributing: ospf 1, rip
 <output omitted>
```

Verify full IGP reachability with TCL script:

```
tclsh
foreach i {
132.1.0.1
132.1.17.1
150.1.1.1
132.1.0.2
132.1.23.2
150.1.2.2
132.1.26.2
132.1.3.3
132.1.0.3
132.1.23.3
150.1.3.3
132.1.35.3
132.1.33.3
132.1.0.4
150.1.4.4
```

```
132.1.255.4
132.1.45.4
132.1.5.5
150.1.5.5
132.1.35.5
132.1.45.5
192.10.1.5
54.1.2.6
132.1.6.6
150.1.6.6
132.1.26.6
132.1.17.7
150.1.7.7
204.12.1.7
132.1.8.8
150.1.8.8
204.12.1.8
132.1.255.9
132.1.255.10

} { puts [ exec "ping $i" ] }
```

Do not worry if you can not ping local unmapped IP addresses on Frame Relay multipoint and physical interfaces. If you are uncertain as to the requirement for your particular lab ask the proctor for clarification. Also for now ignore the 132.1.45.0/24 subnet as it's the backup link between R4 and R5.

As additional verification bring down the Frame Relay link between R3 and R5 by removing the DLCI from either side's subinterface. Once the backup interface is out of the standby state, rerun the ping script.

Although the 3550's and 3560's do not support the TCL shell they do support macros. The macro below can be used for testing from the switches.

```
conf t
macro name ping_internal
do ping 132.1.0.1
do ping 132.1.17.1
do ping 150.1.1.1
do ping 132.1.0.2
do ping 132.1.23.2
do ping 150.1.2.2
do ping 132.1.26.2
do ping 132.1.3.3
do ping 132.1.0.3
do ping 132.1.23.3
do ping 150.1.3.3
do ping 132.1.35.3
do ping 132.1.33.3
do ping 132.1.0.4
do ping 132.1.255.4
do ping 150.1.4.4
do ping 132.1.5.5
do ping 150.1.5.5
do ping 132.1.35.5
```

```
do ping 192.10.1.5
do ping 54.1.2.6
do ping 150.1.6.6
do ping 132.1.26.6
do ping 132.1.17.7
do ping 150.1.7.7
do ping 204.12.1.7
do ping 132.1.8.8
do ping 150.1.8.8
do ping 204.12.1.8
do ping 132.1.255.9
do ping 132.1.255.10
@
macro global apply ping_internal
```

Although points are not taken away for additional configuration it is advisable to remove the macros from the configuration prior to leaving the lab.

Lastly, verify reachability to the backbone IGP networks with following TCL script and macro:

```
tclsh
foreach i {
200.0.0.1
200.0.1.1
200.0.2.1
200.0.3.1
31.3.0.1
31.1.0.1
30.3.0.1
30.1.0.1
} { puts [ exec "ping $i" ] }
```

```
SW1 and SW2:
conf t
macro name ping_external
do ping 200.0.0.1
do ping 200.0.1.1
do ping 200.0.2.1
do ping 200.0.3.1
do ping 31.3.0.1
do ping 31.1.0.1
do ping 30.3.0.1
do ping 30.1.0.1
@
macro global apply ping_external
```

Task 2.5

R5:

```
router bgp 200
 neighbor 192.10.1.254 remote-as 254
 neighbor 192.10.1.254 password CISCO
```


SW1:

```
router bgp 400
 neighbor 204.12.1.254 remote-as 54
 neighbor 204.12.1.254 local-as 100 no-prepend
```

Task 2.5 Verification

Try setting wrong password and see results:

```
Rack1R5#conf t
Rack1R5(config)#router bgp 200
Rack1R5(config-router)#no neighbor 192.10.1.254 password CISCO
Rack1R5(config-router)#neighbor 192.10.1.254 password CISCO1
Rack1R5(config-router)#do clear ip bgp 192.10.1.254
%BGP-5-ADJCHANGE: neighbor 192.10.1.254 Down User reset
%TCP-6-BADAUTH: Invalid MD5 digest from 192.10.1.254(179) to
192.10.1.5(49258)
%TCP-6-BADAUTH: Invalid MD5 digest from 192.10.1.254(179) to
192.10.1.5(49258)
```

Verify that local-AS is configured:

```
Rack1SW1#show ip bgp neighbors 204.12.1.254 | inc local
BGP neighbor is 204.12.1.254,  remote AS 54,  local AS 100 no-prepend,
external link
```

Verify that the local-AS is not prepended on iBGP peering session:

```
Rack1SW1#show ip bgp neighbors 204.12.1.8 advertised-routes
<output omitted>

   Network          Next Hop            Metric LocPrf Weight Path
*> 28.119.16.0/24   204.12.1.254             0             0 54 i
*> 28.119.17.0/24   204.12.1.254             0             0 54 i
*> 112.0.0.0        204.12.1.254                           0 54 50 60 i
*> 113.0.0.0        204.12.1.254                           0 54 50 60 i
```

Remove the "no-prepend" keyword to see the difference:

```
Rack1SW1#conf t
Rack1SW1(config)#router bgp 400
Rack1SW1(config-router)#no neighbor 204.12.1.254 local-as 100 no-
prepend
Rack1SW1(config-router)#neighbor 204.12.1.254 local-as 100 no-prepend
%BGP-5-ADJCHANGE: neighbor 204.12.1.254 Down Local AS change
Rack1SW1(config-router)#neighbor 204.12.1.254 local-as 100
Rack1SW1#show ip bgp neighbors 204.12.1.8 advertised-routes
<output omitted>
   Network          Next Hop       Metric LocPrf Weight Path
*> 28.119.16.0/24   204.12.1.254        0             0 100 54 i
*> 28.119.17.0/24   204.12.1.254        0             0 100 54 i
*> 112.0.0.0        204.12.1.254                      0 100 54 50 60 i
*> 113.0.0.0        204.12.1.254                      0 100 54 50 60 i
*> 114.0.0.0        204.12.1.254                      0 100 54 i
<output omitted>
```

Task 2.6

SW1:

```
router bgp 400
 neighbor 204.12.1.254 route-map STOP_TRANSIT_TO_AS_254 out
!
ip as-path access-list 1 permit _254$
!
route-map STOP_TRANSIT_TO_AS_254 deny 10
 match as-path 1
!
route-map STOP_TRANSIT_TO_AS_254 permit 20
```

Task 2.6 Verification

Check if we have AS 254 originated routes in BGP table:

```
Rack1SW1#show ip bgp regexp _254$
<output omitted>
   Network     Next Hop            Metric LocPrf Weight Path
*> 205.90.31.0 132.1.17.1                             0 300 200 254 ?
*> 220.20.3.0  132.1.17.1                             0 300 200 254 ?
*> 222.22.2.0  132.1.17.1                             0 300 200 254 ?
```

Make sure they are not advertised to AS 54

```
Rack1SW1#show ip bgp neighbor 204.12.1.254 advertised-routes

Rack1SW1#
```

Task 2.7

R5:

```
router bgp 200
 network 132.1.5.0 mask 255.255.255.0
 aggregate-address 132.1.0.0 255.255.0.0 summary-only
 neighbor 132.1.35.3 route-map DENY_AGGREGATE out
 neighbor 132.1.45.4 route-map DENY_AGGREGATE out
!
ip prefix-list DENY_AGGREGATE seq 5 permit 132.1.0.0/16
!
route-map DENY_AGGREGATE deny 10
 match ip address prefix-list DENY_AGGREGATE
!
route-map DENY_AGGREGATE permit 20
```

Task 2.7 Verification

Verify that the aggregate is being generated:

```
Rack1R5#show ip bgp
```



```
<output omitted>
*> 132.1.0.0        0.0.0.0                            32768 i
s> 132.1.5.0/24     0.0.0.0                  0         32768 i
```

Check if we send only the summary route to BB2:

```
Rack1R5#show ip bgp neig 192.10.1.254 advertised-routes | inc 0.0.0.0
*> 132.1.0.0        0.0.0.0                            32768 i
```

Check if we don't send the summary to R3 and R4:

```
Rack1R5#show ip bgp neighbors 132.1.35.3 advertised-routes | inc
132.1.0.0

Rack1R5#
```

Task 2.8

R5:

```
router bgp 200
 bgp nexthop trigger delay 15
 neighbor 192.10.1.254 advertisement-interval 3
```

Task 2.8 Verification

```
RSRack1R5#sh ip bgp neighbors 192.10.1.254 | include advertisement
  Default minimum time between advertisement runs is 30 seconds
  Minimum time between advertisement runs is 3 seconds
```

Task 3.1

R2:

```
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2001:CC1E:1::2/128
!
interface Serial0/0
 ipv6 address 2001:CC1E:1:2323::2/64
 frame-relay map ipv6 2001:CC1E:1:2323::3 203 broadcast
!
ipv6 route 2001:CC1E:1::3/128 Serial0/0 2001:CC1E:1:2323::3
```

R3:

```
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address 2001:CC1E:1::3/128
!
interface Serial1/0
 ipv6 address 2001:CC1E:1:2323::3/64
 frame-relay map ipv6 2001:CC1E:1:2323::2 302 broadcast
!
ipv6 route 2001:CC1E:1::2/128 Serial1/0 2001:CC1E:1:2323::2
```

Task 3.1 Breakdown

Frame Relay is a non-broadcast multi-access (NBMA) media. This implies that for multipoint configurations layer 3 to layer 2 resolution must be obtained. Since only static routing is used, a mapping is not required to the remote link-local address. If dynamic IPv6 routing were configured a mapping for the remote link-local address would be required.

Task 3.1 Verification

```
Verify the Frame Relay IPv6 layer 3 to layer 2 mappings:

Rack1R3#show frame-relay map
<output omitted>
Serial1/0 (up): ipv6 2001:CC1E:1:2323::2 dlci 302(0x12E,0x48E0),
static,
              broadcast,
              CISCO, status defined, active

Verify L3 reachability:

Rack1R3#ping 2001:CC1E:1::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:CC1E:1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
```

Task 4.1

```
R4:
mpls ip
!
interface Loopback100
 ip address 150.1.44.44 255.255.255.255
!
mpls ldp route-id Loopback100 force
!
! The tunnel link is needed to traverse the non-MPLS cloud
! MPLS should be enabled to accept tagged packets
!
interface Tunnel 46
 tunnel source Loopback0
 tunnel destination 150.1.6.6
 ip address 132.1.46.4 255.255.255.0
 mpls ip

!
! We use static routes as the simplest solution
```

```
!
ip route 150.1.66.66 255.255.255.255 Tunnel46

!
! VC types must match on both ends, i.e. both should be VLAN
!
interface FastEthernet 0/0.4
 encapsulation dot1q 4 native
 xconnect 150.1.66.66 46 encapsulation mpls
```

R6:

```
mpls ip
!
interface Loopback100
 ip address 150.1.66.66 255.255.255.255
!
mpls ldp route-id Loopback100 force
!
interface Tunnel 46
 tunnel source Loopback0
 tunnel destination 150.1.4.4
 ip address 132.1.46.6 255.255.255.0
 mpls ip
!
ip route 150.1.44.44 255.255.255.255 Tunnel46

!
interface FastEthernet 0/0.6
 xconnect 150.1.44.44 46 encapsulation mpls
```

Task 4.1 Verification

Check LDP neighbors – AToM uses LDP for signaling.

```
RSRack1R4#show mpls ldp neighbor
    Peer LDP Ident: 150.1.66.66:0; Local LDP Ident 150.1.44.44:0
        TCP connection: 150.1.66.66.16608 - 150.1.44.44.646
        State: Oper; Msgs sent/rcvd: 38/42; Downstream
        Up time: 00:05:09
        LDP discovery sources:
          Targeted Hello 150.1.44.44 -> 150.1.66.66, active, passive
        Addresses bound to peer LDP Ident:
          132.1.26.6      54.1.2.6        150.1.6.6       150.1.66.66
          132.1.46.6

RSRack1R4#show mpls l2transport binding
  Destination Address: 150.1.66.66,  VC ID: 46
    Local Label:  65
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 43
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
```

```
              CV Type: LSPV [2]

RSRack1R4#show mpls l2transport vc detail
Local interface: Fa0/0.4 up, line protocol up, Eth VLAN 4 up
  Destination address: 150.1.66.66, VC ID: 46, VC status: up
    Output interface: Tu46, imposed label stack {43}
    Preferred path: not configured
    Default path: active
    Next hop: point2point
  Create time: 00:05:26, last status change time: 00:05:25
  Signaling protocol: LDP, peer 150.1.66.66:0 up
    MPLS VC labels: local 65, remote 43
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 275, send 211
    byte totals:   receive 16000, send 18321
    packet drops:  receive 0, seq error 0, send 0
```

Task 5.1

R1:

```
ip pim rp-address 150.1.2.2
```

R2:

```
!
ip pim rp-address 150.1.2.2
```

R3:

```
!
ip pim rp-address 150.1.2.2
```

R6:

```
!
ip pim rp-address 150.1.2.2
```

SW1:

```
ip pim rp-address 150.1.2.2
!
interface Vlan783
 ip igmp join-group 228.28.28.28
```

Task 5.1 Verification

Verify the joined groups and multicast routes:

```
Rack1SW1#show ip igmp groups
IGMP Connected Group Membership
Group Address    Interface         Uptime    Expires   Last Reporter
228.28.28.28     Vlan783           00:00:32  00:02:27  204.12.1.7
```

```
224.0.1.40       FastEthernet0/1   00:04:35  00:02:04  132.1.17.7

Rack1SW1#show ip mroute
<output omitted>
(*, 228.28.28.28), 00:00:41/00:02:18, RP 150.1.2.2, flags: SJCL
  Incoming interface: FastEthernet0/1, RPF nbr 132.1.17.1
  Outgoing interface list:
    Vlan783, Forward/Sparse, 00:00:41/00:02:18, H
```

Use m*trace to see how the packets should flow through the network:*

```
Rack1SW1#mtrace 132.1.26.6 228.28.28.28
Type escape sequence to abort.
Mtrace from 132.1.6.6 to 132.1.17.7 via group 228.28.28.28
From source (?) to destination (?)
Querying full reverse path...
 0  132.1.17.7
-1  132.1.17.7 PIM  [132.1.6.0/24]
-2  132.1.17.1 PIM  [132.1.6.0/24]
-3  132.1.0.2 PIM Reached RP/Core [132.1.6.0/24]
-4  132.1.26.6 PIM  [132.1.6.0/24]
```

U*se ping to verify the configuration:*

```
Rack1R6#debug ip mpacket
Rack1R6#ping 228.28.28.28

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 228.28.28.28, timeout is 2 seconds:

IP(0): s=132.1.26.6 (FastEthernet0/0.26) d=228.28.28.28 id=498,
ttl=254, prot=1, len=114(100), mroute olist null
Reply to request 0 from 132.1.17.7, 8 ms
Reply to request 0 from 132.1.17.7, 12 ms
```

Finally look at the output of the multicast routing table:

```
Rack1R6#show ip mroute
<output omitted>

(132.1.26.6, 228.28.28.28), 00:01:09/00:02:24, flags: FT
  Incoming interface: FastEthernet0/0.26, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    FastEthernet0/0.26, Forward/Sparse, 00:01:09/00:03:17
```

Task 5.2

R2:

```
interface Serial0/0
 ip pim nbma-mode
```

Task 6.1

R5:

```
no ip source-route
```

```
no ip bootp server
!
interface FastEthernet0/1
 no ip proxy-arp
 no cdp enable
!
banner login "Access to this device or the attached networks is
prohibited without express written permission.  Violators will be shot
on sight."
```

Task 6.1 Verification

Verify that CDP is disabled on interface FastEthernet0/1 – compare the outputs for the two interfaces:

```
Rack1R5#show cdp interface Fa0/0
FastEthernet0/0 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds

Rack1R5#show cdp interface Fa0/1

Rack1R5#
```

Verify that the commands are in the configuration:

```
Rack1R5#show run | include (source-route|bootp)
no ip source-route
no ip bootp server
```

If you really want to see how source-routing works, try the following command from R4:

```
Rack1R4#traceroute
Protocol [ip]:
Target IP address: 222.22.2.1
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]: L
Source route: 132.1.0.1 132.1.0.2 132.1.23.3 132.1.35.5 192.10.1.254
Loose, Strict, Record, Timestamp, Verbose[LV]:
```

Now try it with source-routing enabled on R5.

```
Rack1R5#show running-config interface Fa0/1

interface FastEthernet0/1
 ip address 192.10.1.5 255.255.255.0
 no ip proxy-arp
 half-duplex
```

```
 no cdp enable
end
```

To be sure that Proxy-ARP is disabled, issue the following command:

```
Rack1R5#sh ip interface Fa0/1 | include Proxy
  Proxy ARP is disabled
  Local Proxy ARP is disabled
```

Verify the login banner:

```
Rack1R3#telnet 150.1.5.5
Trying 150.1.5.5 ... Open
Access to this device or the attached networks is
prohibited without express written permission. Violators will be shot
on sight.

User Access Verification

Password:
```

Task 6.2

R5:

```
!
! ACL for SNMP classification
!
ip access-list extended ACL_SNMP
 permit udp any any eq SNMP

!
! Class-map for SNMP traffic
!
class-map type inspect CMAP_SNMP
 match access-group name ACL_SNMP

!
! Inspection policy for Outside to Inside Traffic
!
policy-map type inspect PMAP_FROM_OUTSIDE_TO_INSIDE
 class type inspect CMAP_SNMP
  drop
 class class-default
  pass

!
! Inspection policy for Inside to Outside Traffic
!
policy-map type inspect PMAP_FROM_INSIDE_TO_OUTSIDE
 class class-default
  pass
!
zone security OUTSIDE
zone security INSIDE
```

```
zone-pair security ZP_OUTSIDE_TO_INSIDE source OUTSIDE destination
INSIDE
  service-policy type inspect PMAP_FROM_OUTSIDE_TO_INSIDE
!
! Zone-Pair for Inside to Outside Traffic
!
zone-pair security ZP_INSIDE_TO_OUTSIDE source INSIDE destination
OUTSIDE
  service-policy type inspect PMAP_FROM_INSIDE_TO_OUTSIDE

!
interface FastEthernet0/1
 zone-member security OUTSIDE
!
interface FastEthernet0/0
 zone-member security INSIDE
!
interface Serial0/1/0
 zone-member security INSIDE

interface Serial0/0/0.1
 zone-member security INSIDE
```

R6:

```
!
! ACL for SNMP classification
!
ip access-list extended ACL_SNMP
 permit udp any any eq SNMP

!
! Class-map for SNMP traffic
!
class-map type inspect CMAP_SNMP
 match access-group name ACL_SNMP

!
! Inspection policy for Outside to Inside Traffic
!
policy-map type inspect PMAP_FROM_OUTSIDE_TO_INSIDE
 class type inspect CMAP_SNMP
  drop
 class class-default
  pass

!
! Inspection policy for Inside to Outside Traffic
!
policy-map type inspect PMAP_FROM_INSIDE_TO_OUTSIDE
 class class-default
  pass
!
zone security OUTSIDE
zone security INSIDE

zone-pair security ZP_OUTSIDE_TO_INSIDE source OUTSIDE destination
INSIDE
```

```
  service-policy type inspect PMAP_FROM_OUTSIDE_TO_INSIDE

!
!
!
zone-pair security ZP_INSIDE_TO_OUTSIDE source INSIDE destination
OUTSIDE
  service-policy type inspect PMAP_FROM_INSIDE_TO_OUTSIDE

!
interface Serial0/0/0
 zone-member security OUTSIDE
!
interface FastEthernet0/0.26
 zone-member security INSIDE
```

Task 6.3

R2 and R4:

```
snmp-server community public RO 1
access-list 1 deny any log
logging 132.1.33.100
```

Task 6.3 Breakdown

The key to this section is to create an access-list that denies all IP address and includes the *log* keyword. The access-list is then bound to the RO community string of *public*. This is a useful technique to track down the source of a host attempting to poll a device.

Task 6.4

R5:

```
interface FastEthernet0/1
 ip access-group DENY_SNMP in
 ip access-group EVALUATE_ICMP out
!
ip access-list extended DENY_SNMP
 deny   udp any any eq snmp
 permit icmp any any time-exceeded
 permit icmp any any port-unreachable
 evaluate ICMP
 deny   icmp any any
 permit ip any any
!
ip access-list extended EVALUATE_ICMP
 permit icmp any any reflect ICMP
 permit ip any any
```

Task 6.4 Verification

To verify our reflective ACL ping from R3 to BB2:

```
Rack1R3#ping 192.10.1.254 repeat 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 192.10.1.254, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!
```

```
Rack1R5#show ip access-lists ICMP
Reflexive IP access list ICMP
     permit icmp host 192.10.1.254 host 132.1.35.3  (400 matches) (time
left 297)
```

Task 7.1

R5 and R6:

```
rmon event 1 trap IETRAP description "Five Minute CPU Average Above
      75%"
rmon event 2 trap IETRAP description "Five Minute CPU Average Below
      40%"
rmon alarm 1 lsystem.58.0 60 absolute rising-threshold 75 1 falling-
      threshold 40 2
!
snmp-server host 132.1.33.100 IETRAP
```

Task 7.1 Verification

Verify RMON configuration:

```
Rack1R6#show rmon alarms
Alarm 1 is active, owned by config
 Monitors lsystem.58.0 every 60 second(s)
 Taking absolute samples, last value was 0
 Rising threshold is 75, assigned to event 1
 Falling threshold is 40, assigned to event 2
 On startup enable rising or falling alarm
```

```
Rack1R6#show rmon events
Event 1 is active, owned by config
 Description is Five Minute CPU Average Above 75%
 Event firing causes trap to community IETRAP,
 last event fired at  0y0w0d,00:00:00,
 Current uptime       0y0w0d,18:12:47
Event 2 is active, owned by config
 Description is Five Minute CPU Average Below 40%
 Event firing causes trap to community IETRAP,
 last event fired at  0y0w0d,18:12:04,
 Current uptime       0y0w0d,18:12:47
```

Task 7.2

R4:
```
username NOC password 0 CISCO
!
line vty 0 4
 exec-timeout 5 0
 logout-warning 60
 absolute-timeout 15
 login local
```

Task 7.3

R4:
```
no username NOC password CISCO
username NOC secret CISCO
```

Task 7.4

R3:
```
interface Serial1/0
 no logging event link-status
 logging event dlci-status-change
!
logging 132.1.33.100
logging trap debugging
```

Task 7.4 Verification

*To verify the logging configuration, use the **show logging** exec command.*

```
Rack1R3#show logging
Syslog logging: enabled (0 messages dropped, 1 messages rate-limited, 0
flushes, 0 overruns, xml disabled)
    Console logging: level debugging, 26 messages logged, xml disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: disabled, xml disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level debugging, 31 message lines logged
        Logging to 132.1.33.100, 2 message lines logged, xml disabled
```

Task 7.5

R5:
```
interface FastEthernet0/1
 ip accounting access-violations
!
ip accounting-threshold 2500
```

```
R6:
interface Serial0/0/0
 ip accounting access-violations
!
ip accounting-threshold 2500
```

Task 7.5 Verification

Access-violation accounting appears to only work with numbered ACLs in the IOS versions used:

```
Rack1R5#show ip access-lists 100
Extended IP access list 100
    10 deny udp any any eq snmp
    20 permit icmp any any time-exceeded
    30 permit icmp any any port-unreachable
    40 deny icmp any any (30 matches)
    50 permit ip any any (8 matches)

Rack1R5#show run interface FastEthernet 0/1
!
interface FastEthernet0/1
 ip address 192.10.1.5 255.255.255.0
 ip access-group 100 in
<output omitted>
```

> ☞ **Note**
> You will not have access to the backbone routers in the real.

```
FRS-BB2>ping 132.1.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 132.1.0.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
FRS-BB2>ping 132.1.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 132.1.3.3, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
FRS-BB2>ping 132.1.33.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 132.1.33.3, timeout is 2 seconds:
U.U.U

Rack1R5#show ip accounting access-violations
   Source           Destination    Packets               Bytes   ACL
 192.10.1.254     132.1.33.3             5                 500   100
 192.10.1.254     132.1.3.3              5                 500   100
 192.10.1.254     132.1.0.4              5                 500   100
```

Task 7.6

SW1, SW2, SW3, SW4:
```
no setup express
```

Task 7.6 Verification

```
RSRack1SW1#show setup express
express setup mode is not active
```

Task 8.1

R3:
```
class-map match-all SMTP_FROM_SERVER
 match access-group name SMTP_FROM_SERVER
!
policy-map CBWFQ
 class SMTP_FROM_SERVER
  bandwidth 256
!
interface Serial1/1
 bandwidth 512
 service-policy output CBWFQ
!
ip access-list extended SMTP_FROM_SERVER
 permit tcp host 132.1.3.100 eq smtp any
```

R5:
```
class-map match-all SMTP_TO_SERVER
 match access-group name SMTP_TO_SERVER
!
policy-map CBWFQ
 class SMTP_TO_SERVER
  bandwidth 256
!
interface Serial0/0/0
 bandwidth 512
 service-policy output CBWFQ
!
ip access-list extended SMTP_TO_SERVER
 permit tcp any host 132.1.3.100 eq smtp
```

Task 8.1 Verification

Verify that the policy-map is configured, applied, and working.

Simulate SMTP traffic from R5:

```
Rack1R5#telnet 132.1.3.100 25 /source-interface Fa0/1
Trying 132.1.3.100, 25 ...
```

Check out policy-map status:

```
Rack1R5#show policy-map interface s0/0/0

 Serial0/0/0

  Service-policy output: CBWFQ

    Class-map: SMTP_TO_SERVER (match-all)
      4 packets, 192 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name SMTP_TO_SERVER
      Queueing
        Output Queue: Conversation 137
        Bandwidth 256 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 4/192
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: class-default (match-any)
      51 packets, 3292 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 8.2

R2:

```
interface FastEthernet0/0
 ip policy route-map POLICY-ROUTE
!
ip access-list extended FTP_FROM_VLAN6
 permit tcp 132.1.26.0 0.0.0.255 host 132.1.33.33 eq ftp
 permit tcp 132.1.26.0 0.0.0.255 host 132.1.33.33 eq ftp-data
!
route-map POLICY-ROUTE permit 10
 match ip address FTP_FROM_VLAN6
 set ip next-hop 132.1.23.3
```

R3:

```
interface FastEthernet0/1
 ip policy route-map POLICY-ROUTE
!
ip access-list extended FTP_FROM_SERVER
 permit tcp host 132.1.33.33 eq ftp 132.1.26.0 0.0.0.255
 permit tcp host 132.1.33.33 eq ftp-data 132.1.26.0 0.0.0.255
!
route-map POLICY-ROUTE permit 10
 match ip address FTP_FROM_SERVER
 set ip next-hop 132.1.23.2
```

Task 8.3

R2:

```
class-map match-all FTP_FROM_VLAN6
  match access-group name FTP_FROM_VLAN6
!
```

```
policy-map RESERVE_FTP
 class FTP_FROM_VLAN6
  bandwidth 256
!
interface Serial0/1
 bandwidth 1536
 service-policy output RESERVE_FTP
```

R3:

```
class-map match-all FTP_FROM_SERVER
  match access-group name FTP_FROM_SERVER
!
policy-map RESERVE_FTP
  class FTP_FROM_SERVER
   bandwidth 256
!
interface Serial1/3
 bandwidth 1536
 service-policy output RESERVE_FTP
```

Task 8.3 Verification

Verify the policy-map configuration:

```
Rack1R3#show policy-map interface s1/3
 Serial1/3

  Service-policy output: RESERVE_FTP

    Class-map: FTP_FROM_SERVER (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name FTP_FROM_SERVER
      Queueing
        Output Queue: Conversation 265
        Bandwidth 256 (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)
      2 packets, 88 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 8.4

R2:

```
interface Serial0/0
 frame-relay traffic-shaping
 frame-relay class REMAINING_BW
 frame-relay interface-dlci 204
  class DLCI_204
!
map-class frame-relay DLCI_204
```



```
 frame-relay cir 128000
 frame-relay bc 1280
!
map-class frame-relay REMAINING_BW
 frame-relay cir 192000
 frame-relay bc 24000
```

R4:

```
interface Serial0/0/0
 frame-relay class REMAINING_BW
 frame-relay traffic-shaping
 frame-relay interface-dlci 402
  class DLCI_402
!
map-class frame-relay DLCI_402
 frame-relay cir 128000
 frame-relay bc 1280
!
map-class frame-relay REMAINING_BW
 frame-relay cir 192000
 frame-relay bc 24000
```

Task 8.4 Verification

Verify the FRTS parameters:

```
Rack1R4#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
413           192000 3000   24000     0        125       3000      -
401           192000 3000   24000     0        125       3000      -
402           128000 160    1280      0        10        160       -
403           192000 3000   24000     0        125       3000      -
405           192000 3000   24000     0        125       3000      -
```