Advanced Networking 2 Case Study

Final Report

Noah Morin, Blake Whiting, Shriji Shah

Ontario Tech University

Oshawa, Canada

noah.morin@ontariotechu.net,

blake.whiting@ontariotechu.net,

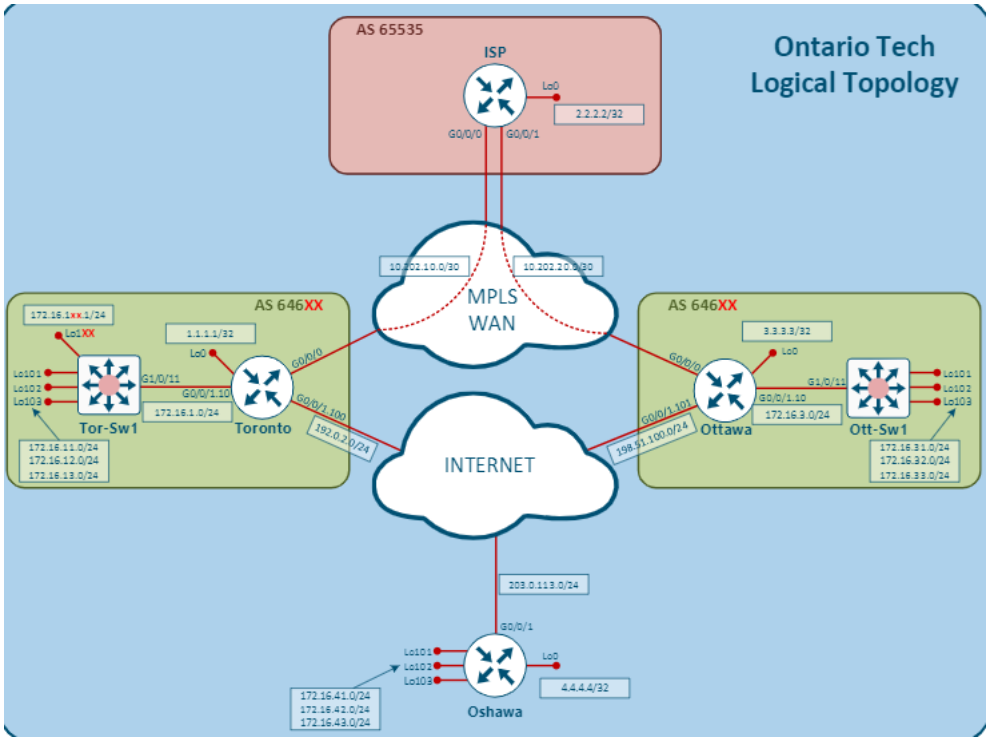
shriji.shah@ontariotechu.net

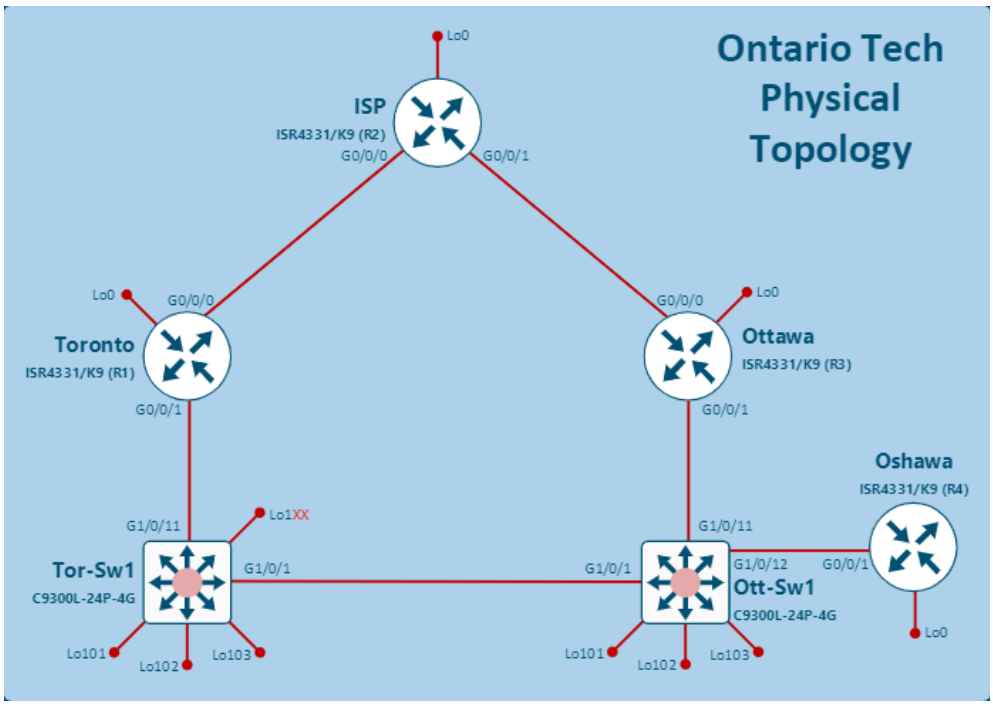
I. table of contents

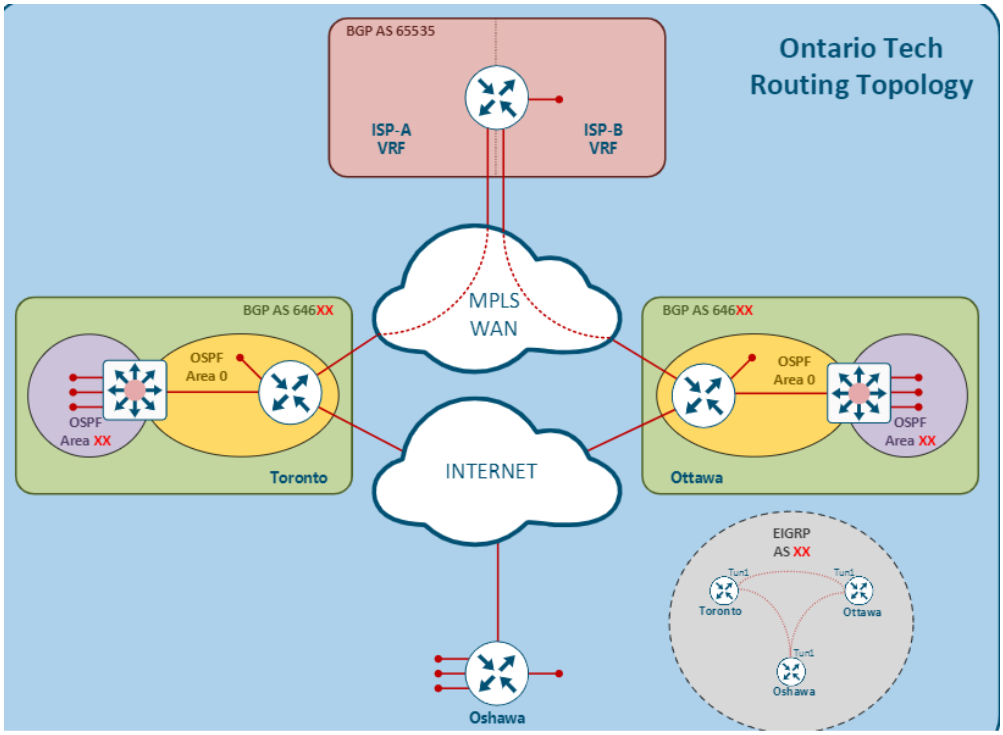
II. abstract

Our group is being tasked with assisting, designing, and deploying new WAN infrastructure for Ontario Tech and it's affiliates. In this scenario, we are implementing networks on three different campuses under Ontario Tech, north Oshawa, Toronto, and Ottawa. This network incorporates concepts learned in the Advanced Networking 2 course, these included the use of protocols such as DMVPN, MPLS, and PBR, along with concepts including route redistribution and route filtering

III. topology diagrams






Ontario Tech
Routing Topology
BGP AS 65535
ISP-A
VRF
ISP-B
VRF
MPLS
WAN
INTERNET
BGP AS 646XX
OSPF
Area 0
OSPF
Area XX
Toronto
BGP AS 646XX
OSPF
Area 0
OSPF
Area XX
Ottawa
Oshawa
EIGRP
AS XX
Tun1
Tun1
Tun1
Toronto
Ottawa
Oshawa

IV. configuration steps

task 1: addressing

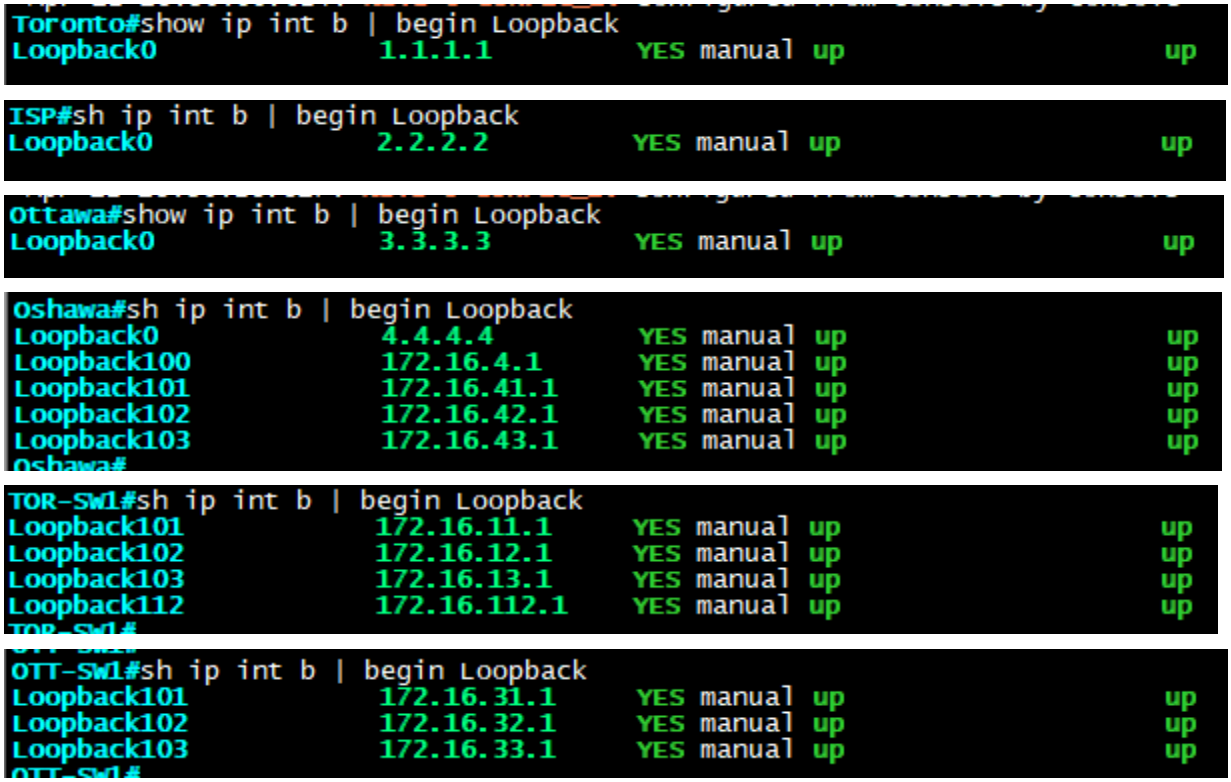
Task	Commands Used

Create the loopback interfaces on the routers and switches as indicated in the logical topology diagram. Note that the tunnel interfaces will be created and addressed in a later step.	Toronto:
	`Interface Loopback0`
	`Ip address 1.1.1.1 255.255.255.255`
	
	ISP:
	`Interface Loopback0`
	`Ip address 2.2.2.2 255.255.255.255`
	
	Ottawa:
	`Interface Loopback0`
	`Ip address 3.3.3.3 255.255.255.255`
	
	Oshawa:
	`Interface Loopback0`
	`Ip address 4.4.4.4 255.255.255.255`
	`Interface Loopback101`
	`Ip address 172.16.41.1 255.255.255.0`
	`Interface Loopback102`
	`Ip address 172.16.42.1 255.255.255.0`
	`Interface Loopback103`
	`Ip address 172.16.43.1 255.255.255.0`
	
	Tor-Sw1:
	`Interface vlan 10`
	`Ip address 172.16.1.2 255.255.255.0`
	`Interface Lo101`
	`Ip address 172.16.11.1 255.255.255.0`
	`Interface Lo102`
	`Ip address 172.16.12.1 255.255.255.0`
	`Interface Lo103`
	`Ip address 172.16.13.1 255.255.255.0`
	`Interface Lo112`
	`Ip address 172.16.112.1 255.255.255.0`
	
	Ott-Sw1:
	`Interface loopback 101`
	`Ip address 172.16.31.1 255.255.255.0`

	Interface loopback 102 Ip address 172.16.32.1 255.255.255.0 Interface loopback 103 Ip address 172.16.33.1 255.255.255.0

Summary:
Using the IP address command, we configured addresses on all the loopback interfaces

Screenshot:



Task	Commands Used
Using the addressing table below, assign IP addresses to each of the interfaces. Note that some interfaces have already been pre-configured in Task 0.	Toronto: Interface g0/0/0 Ip address 10.202.10.2 255.255.255.252 Interface g0/0/1.10 Encapsulation dot1q 10 Ip address 172.16.1.1 255.255.255.0 Ip nat inside Interface g0/0/1.100 Encapsulation dot1q 100

	Ip address 192.0.2.12 255.255.255.0 Ip nat outside ISP: interface GigabitEthernet0/0/0 vrf forwarding ISP-A ip address 10.202.10.1 255.255.255.252 interface GigabitEthernet0/0/1 vrf forwarding ISP-B ip address 10.202.20.1 255.255.255.252 Ottawa: interface GigabitEthernet0/0/0 ip address 10.202.20.2 255.255.255.252 interface GigabitEthernet0/0/1 no ip address interface GigabitEthernet0/0/1.10 encapsulation dot1Q 10 ip address 172.16.3.1 255.255.255.0 ip nat inside interface GigabitEthernet0/0/1.101 encapsulation dot1Q 101 Ip address 198.51.100.12 255.255.255.0 Oshawa: interface GigabitEthernet0/0/1 ip address 203.0.113.12 255.255.255.0 ip nat outside

Summary:
Using the IP address command, we configured addresses on multiple interfaces

Screenshot:

```
Toronto#sh ip int b
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   10.202.10.2     YES manual up                    up
GigabitEthernet0/0/1   unassigned      YES unset  up                    up
Gi0/0/1.10             172.16.1.1      YES manual up                    up
Gi0/0/1.100            192.0.2.12      YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
Serial0/1/0            unassigned      YES manual administratively down down
Serial0/1/1            unassigned      YES manual administratively down down
GigabitEthernet0       unassigned      YES DHCP   administratively down down
Loopback0              1.1.1.1         YES manual up                    up
Toronto#
```

```
ISP#sh ip int b
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0       10.202.10.1     YES manual up                    up
GigabitEthernet0/0/1       10.202.20.1     YES manual up                    up
GigabitEthernet0/0/2       unassigned      YES unset  administratively down down
GigabitEthernet0           unassigned      YES DHCP   administratively down down
Loopback0                  2.2.2.2         YES manual up                    up
```

```
Ottawa#sh ip int b
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0       10.202.20.2     YES manual up                    up
GigabitEthernet0/0/1       unassigned      YES unset  up                    up
Gi0/0/1.10                 172.16.3.1      YES manual up                    up
Gi0/0/1.100                unassigned      YES unset  deleted               down
Gi0/0/1.101                198.51.100.12   YES manual up                    up
GigabitEthernet0/0/2       unassigned      YES unset  administratively down down
Serial0/1/0                unassigned      YES manual administratively down down
Serial0/1/1                unassigned      YES manual administratively down down
GigabitEthernet0           unassigned      YES DHCP   administratively down down
Loopback0                  3.3.3.3         YES manual up                    up
```

```
Oshawa#sh ip int b
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0       unassigned      YES unset  administratively down down
GigabitEthernet0/0/1       203.0.113.12    YES manual up                    up
GigabitEthernet0/0/2       unassigned      YES unset  administratively down down
GigabitEthernet0           unassigned      YES DHCP   administratively down down
Loopback0                  4.4.4.4         YES manual up                    up
Loopback100                172.16.4.1      YES manual up                    up
Loopback101                172.16.41.1     YES manual up                    up
Loopback102                172.16.42.1     YES manual up                    up
Loopback103                172.16.43.1     YES manual up                    up
Oshawa#
```

```
TOR-SW1(config-if)#do sh ip int b | incl Vlan
Vlan1                  unassigned      YES unset  up                    up
Vlan10                 172.16.1.2      YES manual up                    up
TOR-SW1(config-if)#
```

```
OTT-SW1#sh ip int b | incl Vlan
Vlan1                  unassigned      YES unset  up                    up
Vlan10                 172.16.3.2      YES manual up                    up
Vlan100                192.0.2.254     YES manual up                    up
Vlan101                198.51.100.254  YES manual up                    up
Vlan102                203.0.113.254   YES manual up                    up
OTT-SW1#
```

task 2: configure OSPF

Task	Commands Used
Enable IPv6 unicast routing on **Toronto**, **Tor-Sw1**, **Ottawa**, and **Ott-Sw1**.	`Ipv6 unicast-routing`

Summary:
Using the Ipv6 unicast-routing command we Enabled IPv6 unicast routing on multiple devices

Screenshot:

```
Toronto#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "application"
IPv6 Routing Protocol is "ospf 12"
IPv6 Routing Protocol is "ND"
```

```
Ottawa#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "application"
IPv6 Routing Protocol is "ospf 12"
IPv6 Routing Protocol is "ND"
```

```
TOR-SW1#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ospf 12"
IPv6 Routing Protocol is "ND"
```

```
OTT-SW1#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ospf 12"
IPv6 Routing Protocol is "ND"
```

Task	Commands Used
Enable OSPFv3 on **Toronto**, **Tor-Sw1**, **Ottawa**, and **Ott-Sw1** for the IPv4 address family.	`Address-family ipv4 unicast`

Summary:
Enable address families for IPv4 on all of the switches in the OSPFv3 configuration

Screenshot:

```
Toronto#show ospfv3 12 ipv4 | incl OSPFv3
 OSPFv3 12 address-family ipv4
```

```
Ottawa#show ospfv3 12 ipv4 | incl OSPFv3
 OSPFv3 12 address-family ipv4
```

```
TOR-SW1#show ospfv3 12 ipv4 | incl OSPFv3
 OSPFv3 12 address-family ipv4
```

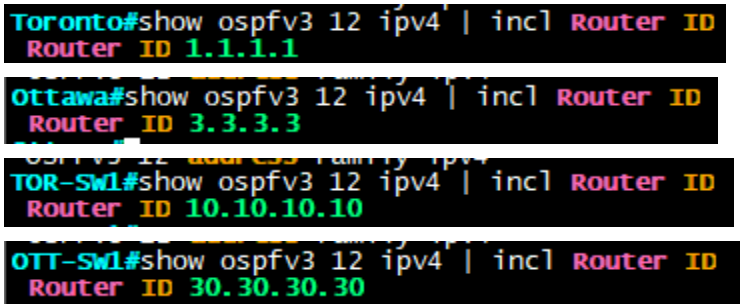
```
OTT-SW1#show ospfv3 12 ipv4 | incl OSPFv3
 OSPFv3 12 address-family ipv4
```

Task	Commands Used

Configure the following router IDs on each device within the IPv4 address family:
o Toronto: **1.1.1.1**
o Tor-Sw1: **10.10.10.10**
o Ottawa: **3.3.3.3**
o Ott-Sw1: **30.30.30.30**
	Toronto
	`Router-id 1.1.1.1`
	
	Tor-Switch1
	`Router-id 10.10.10.10`
	
	Ottawa
	`Router-id 3.3.3.3`
	
	Ott-Sw1
	`Router-id 30.30.30.30`

Summary:
Using the router-id command, we configured router-ID's for each of the specified routers.

Screenshot:



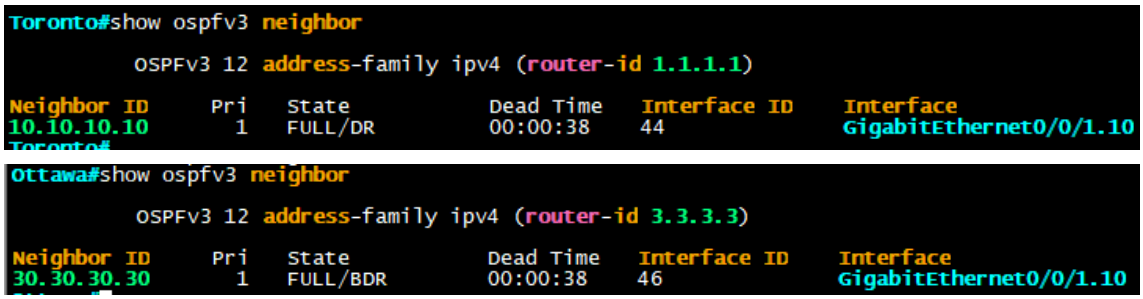
Task	Commands Used

• Enable OSPFv3 on the following links:
o Toronto to Tor-Sw1 in area 0
o All Tor-Sw1 Loopbacks in **area xx**
o Ottawa to Ott-Sw1 in **area 0**
o All Ott-Sw1 Loopbacks in **area xx**
	Toronto
	`interface g0/0/1.10`
	`ipv6 enable`
	`ospfv3 12 ipv4 area 0`
	
	Ottawa
	`interface g0/0/1.10`
	`Ipv6 enable`
	`ospfv3 12 ipv4 area 0`
	
	Tor-Sw1
	`interface Lo101`
	`ipv6 enable`

	ospfv3 12 ipv4 area 12 interface Lo102 ipv6 enable ospfv3 12 ipv4 area 12 interface Lo103 ipv6 enable ospfv3 12 ipv4 area 12 interface Lo112 ipv6 enable ospfv3 12 ipv4 area 12 Interface vlan 10 Ipv6 enable Ospfv3 12 ipv4 area 0 Ott-Sw1 interface Lo101 ipv6 enable ospfv3 12 ipv4 area 12 interface Lo102 ipv6 enable ospfv3 12 ipv4 area 12 interface Lo103 ipv6 enable ospfv3 12 ipv4 area 12 Interface vlan 10 Ipv6 enable Ospfv3 12 ipv4 area 0

Summary:
To enable OSPFv3 operation within the network, we configured ipv6 on the specified interface and used the ospfv3 12 ipv4 area 12 command to enable OSPF

Screenshot:



```
TOR-SW1#show ospfv3 neighbor

          OSPFv3 12 address-family ipv4 (router-id 10.10.10.10)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           1   FULL/BDR        00:00:35    13              Vlan10
```

```
OTT-SW1#show ospfv3 neighbor

          OSPFv3 12 address-family ipv4 (router-id 30.30.30.30)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           1   FULL/DR         00:00:33    12              Vlan10
```

Task	Commands Used
• Change the OSPF reference bandwidth to **100Gbps**.	`Auto-cost reference-bandwidth 100000`

Summary:
Using the Auto-cost reference-bandwidth command, we set the bandwidth to 100000, due to referenced bandwidth is 100 Mbps..

Screenshot:

```
Toronto#show ospfv3 | incl Ref
 Reference bandwidth unit is 100000 mbps
```

```
Ottawa#show ospfv3 | incl Ref
 Reference bandwidth unit is 100000 mbps
```

```
TOR-SW1#show ospfv3 | incl Ref
 Reference bandwidth unit is 100000 mbps
```

```
OTT-SW1#show ospfv3 | incl Ref
 Reference bandwidth unit is 100000 mbps
```

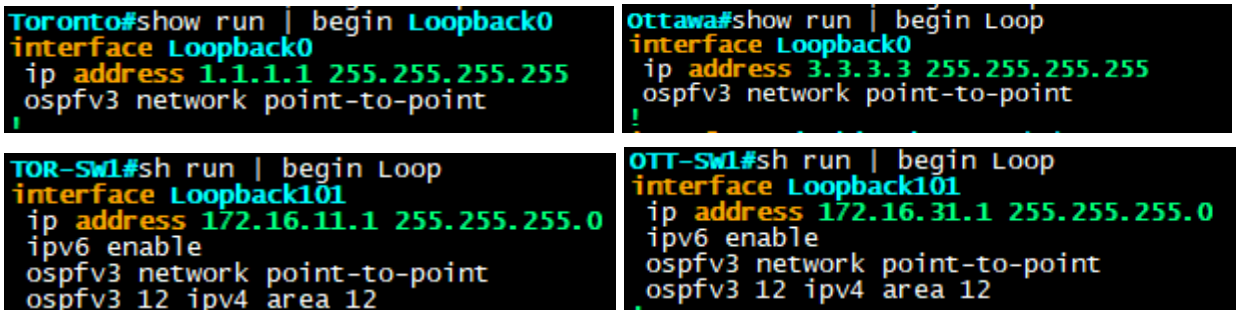
Task	Commands Used

• Change the network type on the loopback interfaces to **point-to-point** so that the routes are advertised with the correct subnet mask rather than /32.	Toronto
	`Interface lo0`
	`Ospf network point-to-point`
	
	Ottawa
	`Interface lo0`
	`Ospf network point-to-point`
	
	Tor-Sw1
	`interface Lo101`
	`ospf network point-to-point`
	`interface Lo102`
	`ospf network point-to-point`
	`interface Lo103`
	`ospf network point-to-point`

	interface Lo112 ospf network point-to-point Ott-Sw1 interface Lo101 ospf network point-to-point interface Lo102 ospf network point-to-point interface Lo103 ospf network point-to-point

Summary:
On all of the loopback interfaces, we used the ospf network point-to-point command to enable point-to-point routing rules.

Screenshot:

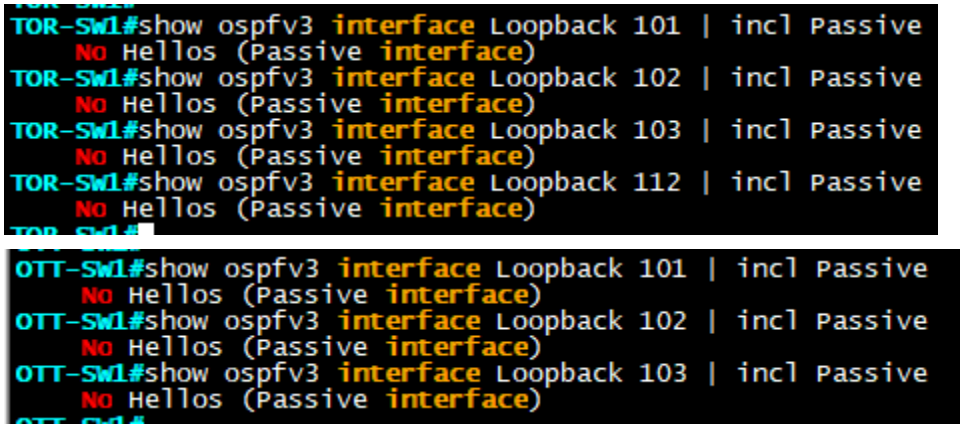


Task	Commands Used
• Configure all Loopback interfaces as passive.	Toronto: Address-family ipv4 unicast passive-interface Lo0 Ottawa: Address-family ipv4 unicast passive-interface Lo0 Tor-Sw1: Address-family ipv4 unicast passive-interface Lo101 passive-interface Lo102 passive-interface Lo103 passive-interface Lo112

	Ott-Sw1:
	`Address-family ipv4 unicast`
	`passive-interface Lo101`
	`passive-interface Lo102`
	`passive-interface Lo103`

Summary:
Using the passive-interface command, we enabled each loopback interface as passive

Screenshot:



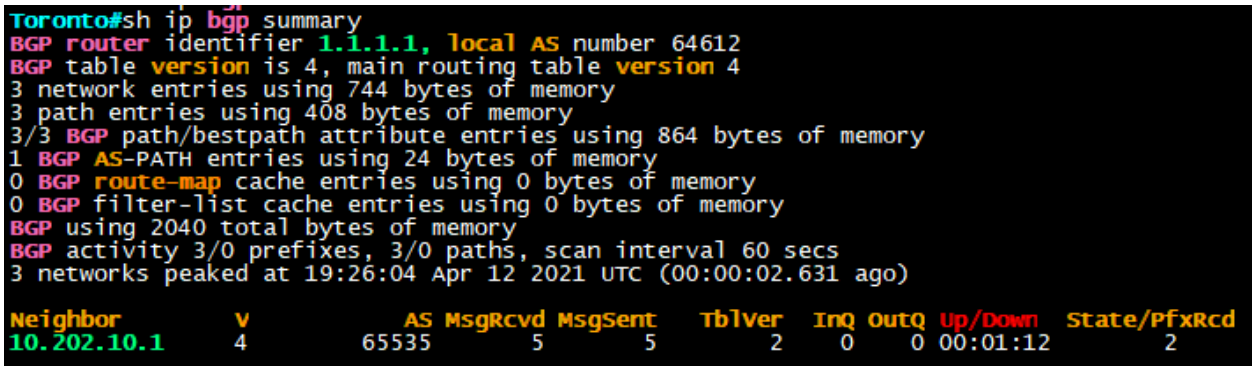
task 3: configure BGP

Task	Commands Used

Configure BGP on **Toronto**, **ISP**, and **Ottawa**
• **Toronto** and **Ottawa** are in AS **646xx**.
• **ISP** is in AS **65535**.
• On **ISP** configure the **Toronto** neighbor relationship in the VRF **ISP-A** address family and the **Ottawa** neighbor relationship in the VRF **ISP-B** address family.
	Toronto:
	`Router bgp 64612`
	`Address-family ipv4`
	
	ISP:
	`Router bgp 65535`
	`Address-family ipv4 vrf ISP-A`
	`Neighbor 10.202.10.2 remote-as 64612`
	`Address-family ipv4 vrf ISP-B`
	`Neighbor 10.202.20.2 remote-as 64612`
	
	Ottawa:
	`Router bgp 64612`
	`Address-family ipv4`

Summary:
Beginning the configuration with the router bgp command, we configured BGP using the AS 65535. We then configured a neighbour address on both vrf ISP-A and ISP-B using the address-family command.

Screenshot:



```
ISP#show ip bgp vpnv4 vrf ISP-B summary
BGP router identifier 2.2.2.2, local AS number 65535
BGP table version is 9, main routing table version 9
4 network entries using 1024 bytes of memory
4 path entries using 544 bytes of memory
3/2 BGP path/bestpath attribute entries using 912 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2528 total bytes of memory
BGP activity 8/0 prefixes, 8/0 paths, scan interval 60 secs
8 networks peaked at 19:25:53 Apr 12 2021 UTC (00:01:01.766 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.202.20.2     4        64612       6       7        9    0    0 00:02:11        1
```

```
Ottawa#show ip bgp summary
BGP router identifier 3.3.3.3, local AS number 64612
BGP table version is 5, main routing table version 5
4 network entries using 992 bytes of memory
4 path entries using 544 bytes of memory
4/4 BGP path/bestpath attribute entries using 1152 bytes of memory
2 BGP AS-PATH entries using 64 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2752 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs
4 networks peaked at 19:25:59 Apr 12 2021 UTC (00:01:52.874 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.202.20.1     4        65535       8       6        5    0    0 00:02:37        3
```

Task	Commands Used

Use router ID **1.1.1.1** for **Toronto**, **2.2.2.2** for **ISP**, and **3.3.3.3** for **Ottawa**.	Toronto:
	`bgp router-id 1.1.1.1`
	
	ISP:
	`bgp router-id 2.2.2.2`
	
	Ottawa:
	`bgp router-id 3.3.3.3`

Summary:
Using the bgp-router id <ip-address> command, we used the respective router ids for each router to set the router ids.

Screenshot:

```
Toronto#sh ip bgp summary
BGP router identifier 1.1.1.1, local AS number 64612
```

```
ISP#show ip bgp vpnv4 vrf ISP-B summary
BGP router identifier 2.2.2.2, local AS number 65535
```

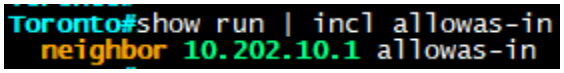
```
Ottawa#show ip bgp summary
BGP router identifier 3.3.3.3, local AS number 64612
```

Task	Commands Used

Configure **Toronto** and **Ottawa** with the *allowas-in parameter* for their neighbor relationships with **ISP**.
• Because **Toronto** and **Ottawa** are in the same AS, separated by the ISP, the **`allowas-in`** parameter ensures they do not discard the route if their own AS is in the AS Path. Be very careful using this in the real world as it could introduce routing loops.
	Toronto:
	`Toronto(config-router-af)#`
	`Neighbor 10.202.10.1 remote-as 65535`
	`Neighbor 10.202.10.1 allowas-in`
	
	Ottawa:
	`Ottawa(config-router-af)#`
	`Neighbor 10.202.20.1 remote-as 65535`
	`Neighbor 10.202.20.1 allowas-in`

Summary:
Further in the BGP configuration, we added another neighbour with the allowas-in parameter to ensure the router does not discard the route within their own AS.

Screenshot:

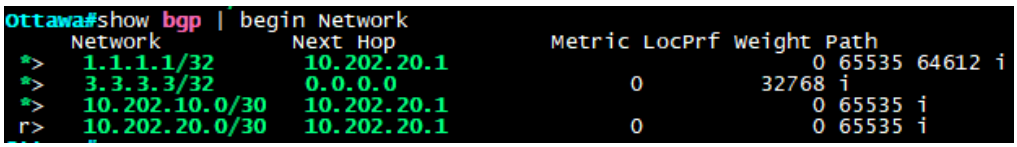


Task	Commands Used

Advertise the Loopback 0 networks into BGP on **Toronto** and **Ottawa**.	Toronto:
	`Toronto(config-router-af)#`
	`Network 1.1.1.1 mask 255.255.255.255`
	
	Ottawa:
	`Ottawa(config-router-af)#`
	`Network 3.3.3.3 mask 255.255.255.255`

Summary:
Using the network command with the loopback interface IP address and mask, we advertised the loopback0 into BGP for each respective device.

Screenshot:



```
Toronto#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>   1.1.1.1/32       0.0.0.0                  0         32768 i
 *>   3.3.3.3/32       10.202.10.1                            0 65535 64612 i
 r>   10.202.10.0/30   10.202.10.1              0             0 65535 i
 *>   10.202.20.0/30   10.202.10.1                            0 65535 i
```

Task	Commands Used

On **ISP**, advertise the **10.202.10.0/30** subnet into the **ISP-A** VRF, and the **10.202.20.0/30** subnet into the **ISP-B** VRF	ISP:
	`ISP(config-router-af)#`
	`Address-family ipv4 vrf ISP-A`
	`Network 10.202.10.0 mask 255.255.255.252`
	`Address-family ipv4 vrf ISP-B`
	`Network 10.202.20.0 mask 255.255.255.252`

Summary:
On ISP in each corresponding VRF address family, we used the network command to advertise each network and mask.

Screenshot:

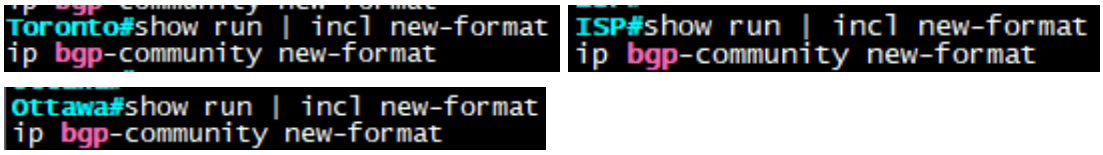
```
ISP#show bgp vpnv4 unicast vrf ISP-B | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:2 (default for vrf ISP-B)
 *>   1.1.1.1/32       10.202.10.2              0             0 64612 i
 *>   3.3.3.3/32       10.202.20.2              0             0 64612 i
 *>   10.202.10.0/30   0.0.0.0                  0         32768 i
 *>   10.202.20.0/30   0.0.0.0                  0         32768 i
ISP#show bgp vpnv4 unicast vrf ISP-A | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf ISP-A)
 *>   1.1.1.1/32       10.202.10.2              0             0 64612 i
 *>   3.3.3.3/32       10.202.20.2              0             0 64612 i
 *>   10.202.10.0/30   0.0.0.0                  0         32768 i
 *>   10.202.20.0/30   0.0.0.0                  0         32768 i
```

Task	Commands Used

Configure **Toronto**, **Ottawa**, and **ISP** to use the new format of BGP communities	Toronto:
	`Toronto(config)#`
	`ip bgp-community new-format`
	
	ISP:
	`ISP(config)#`
	`ip bgp-community new-format`
	
	Ottawa:
	`Ottawa(config)#`
	`ip bgp-community new-format`

Summary:
On each router, we used the global configuration command to configure the new bgp community format.

Screenshot:

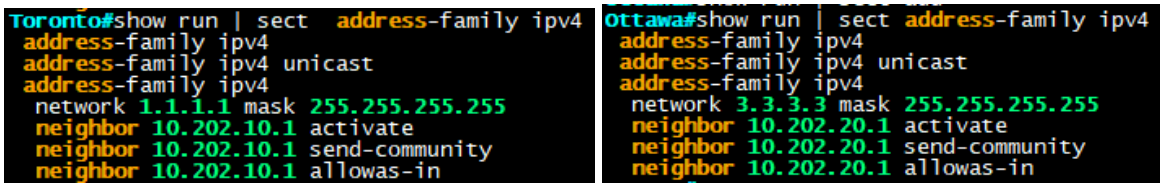


Task	Commands Used

Configure **Toronto**, **Ottawa**, and **ISP** to send communities to all BGP neighbors.	Toronto:
	`Toronto(config-router-af)#`
	`Neighbor 10.202.10.1 activate`
	`Neighbor 10.202.10.1 send-community`
	
	ISP:
	`ISP(config-router-af)#`
	`Address-family ipv4 vrf ISP-A`
	`Neighbor 10.202.10.2 activate`
	`Neighbor 10.202.10.2 send-community`
	`Address-family ipv4 vrf ISP-B`
	`Neighbor 10.202.20.2 activate`
	`Neighbor 10.202.20.2 send-community`
	
	Ottawa:
	`Ottawa(config-router-af)#`
	`Neighbor 10.202.20.1 activate`
	`Neighbor 10.202.20.1 send-community`

Summary:
Using the send-community parameter on the neighbour configuration, we are specifying why information the neighbor will share

Screenshot:



```
ISP#show run | section address-family ipv4
 address-family ipv4
 address-family ipv4
 address-family ipv4
 address-family ipv4 vrf ISP-A
  network 2.2.2.2 mask 255.255.255.255
  network 10.202.10.0 mask 255.255.255.252
  neighbor 10.202.10.2 remote-as 64612
  neighbor 10.202.10.2 activate
  neighbor 10.202.10.2 send-community
 address-family ipv4 vrf ISP-B
  network 2.2.2.2 mask 255.255.255.255
  network 10.202.20.0 mask 255.255.255.252
  neighbor 10.202.20.2 remote-as 64612
  neighbor 10.202.20.2 activate
  neighbor 10.202.20.2 send-community
```

Task	Commands Used

On **Ottawa**, set a BGP community of **646xx:xx** on the **172.16.32.0** and **172.16.33.0** routes being advertised to **ISP-B** via BGP. Remember there is an implicit deny at the end of every route map!	Ottawa:
	`Ottawa(config)#`
	`ip prefix-list ALLOW-32-33 seq 5 permit 172.16.32.0/24`
	`ip prefix-list ALLOW-32-33 seq 10 permit 172.16.33.0/24`
	`route-map ISP-B-IN permit 10`
	`Match ip address prefix-list ALLOW-32-33`
	`set community 64612:12`

Summary:
Using a prefix list inside of a route map, we specified permit statements which apply to a small range of addresses. We further applied the community 64612:12 on these addresses

Screenshot:

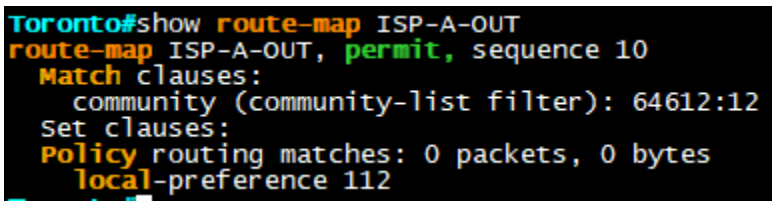
```
Ottawa#show ip prefix-list ALLOW-32-33
ip prefix-list ALLOW-32-33: 2 entries
   seq 5 permit 172.16.32.0/24
   seq 10 permit 172.16.33.0/24
Ottawa#show route-map ISP-B-IN
route-map ISP-B-IN, permit, sequence 10
  Match clauses:
    ip address prefix-lists: ALLOW-32-33
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
    community 64612:12
```

Task	Commands Used

On **Toronto**, match any routes that have the BGP community **646xx:xx** and set a Local Preference value of **1xx** on those routes.Remember there is an implicit deny at the end of every route map!	Toronto:
	`Toronto(config)#`
	`route-map ISP-A-OUT permit 10`
	`Match community 64612:12`
	`Set local-preference 112`

Summary:
Using a route map, we specified which BGP community would be accepted and further applied a local preference of 112.

Screenshot:



task 4: configure MPLS

Task	Commands Used

Enable MPLS on the link between **Toronto** and **ISP**, and the link between **Ottawa** and **ISP**.	Toronto:
	`Interface g0/0/0`
	`Mpls ip`
	
	ISP:
	`Interface g0/0/0`
	`Mpls ip`
	`Interface g0/0/1`
	`Mpls ip`
	
	Ottawa:
	`Interface g0/0/0`
	`Mpls ip`

Summary:
To configure MPLS, we used the command *mpls ip* in interface configuration mode.

Screenshot:

```
Toronto#show mpls interfaces
Interface              IP           Tunnel   BGP Static Operational
GigabitEthernet0/0/0   Yes (ldp)    No       No  No     Yes
```

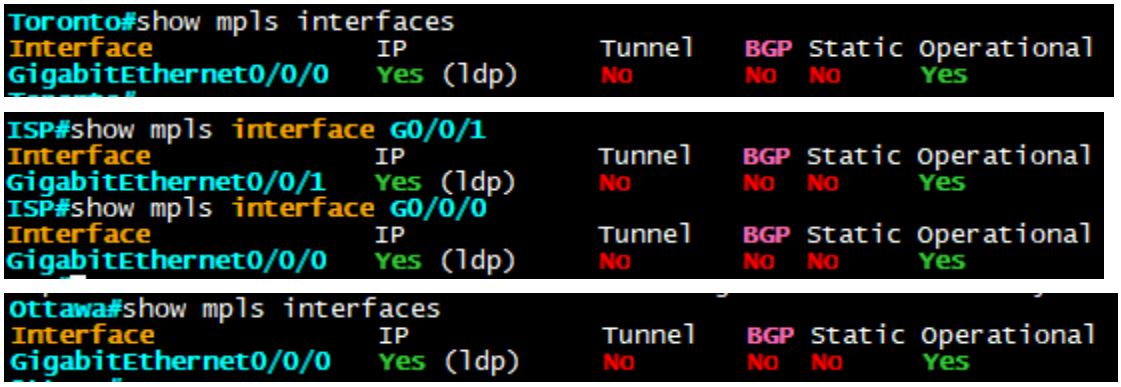
```
ISP#show mpls interface G0/0/1
Interface              IP           Tunnel   BGP Static Operational
GigabitEthernet0/0/1   Yes (ldp)    No       No  No     Yes
ISP#show mpls interface G0/0/0
Interface              IP           Tunnel   BGP Static Operational
GigabitEthernet0/0/0   Yes (ldp)    No       No  No     Yes
```

```
Ottawa#show mpls interfaces
Interface              IP           Tunnel   BGP Static Operational
GigabitEthernet0/0/0   Yes (ldp)    No       No  No     Yes
```

Task	Commands Used
Set the label protocol to LDP.	`router(config-if)#` `mpls label protocol ldp`

Summary:
To configure LDP as the label protocol for MPLS, the command `mpls label protocol ldp` was used. This forces the label distribution protocol to be LDP.

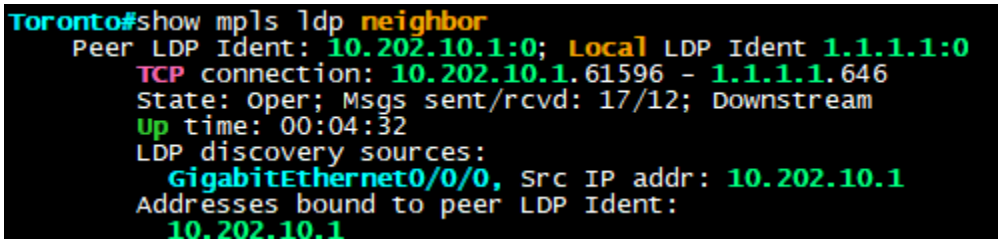
Screenshot:



Task	Commands Used
Force the routers to use their Loopback 0 interfaces as the LDP router ID.	`router(config)#` `mpls ldp router-id lo0 force`

Summary:
To force the routers to use their Loopback 0 interfaces as the LDP router id, we used the command `mpls ldp router-id lo0 force` in global configuration mode.

Screenshot:



```
ISP#show mpls ldp neighbor vrf ISP-A
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 10.202.10.1:0
        TCP connection: 1.1.1.1.646 - 10.202.10.1.61596
        State: Oper; Msgs sent/rcvd: 13/18; Downstream
        Up time: 00:05:32
        LDP discovery sources:
          GigabitEthernet0/0/0, Src IP addr: 10.202.10.2
        Addresses bound to peer LDP Ident:
          10.202.10.2     172.16.1.1      192.0.2.12      1.1.1.1
ISP#show mpls ldp neighbor vrf ISP-B
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 10.202.20.1:0
        TCP connection: 3.3.3.3.646 - 10.202.20.1.48893
        State: Oper; Msgs sent/rcvd: 13/17; Downstream
        Up time: 00:05:26
        LDP discovery sources:
          GigabitEthernet0/0/1, Src IP addr: 10.202.20.2
        Addresses bound to peer LDP Ident:
          10.202.20.2     172.16.3.1      198.51.100.12   3.3.3.3
```

```
Ottawa#show mpls ldp neighbor
    Peer LDP Ident: 10.202.20.1:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 10.202.20.1.48893 - 3.3.3.3.646
        State: Oper; Msgs sent/rcvd: 17/13; Downstream
        Up time: 00:05:45
        LDP discovery sources:
          GigabitEthernet0/0/0, Src IP addr: 10.202.20.1
        Addresses bound to peer LDP Ident:
          10.202.20.1
```

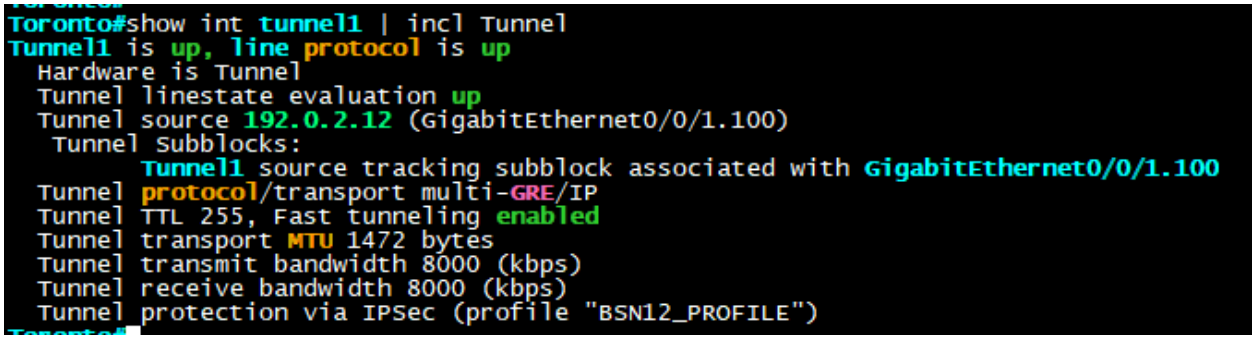
task 5: configure DMVPN phase 3

Task	Commands Used

Configure a Tunnel1 interface on **Toronto**, **Ottawa**, and **Oshawa**.
● Set the tunnel interfaces on all three routers to use multipoint GRE.
● Set the tunnel source on all three routers to be the interface connecting to the Internet.
● Set the tunnel key on all three routers to be your group number (xx).
● Set the IP address of the tunnel interfaces as follows:
○ Toronto: **10.1.xx.1/24**
○ Ottawa: **10.1.xx.2/24**
○ Oshawa: **10.1.xx.3/24**
	Toronto:
	`Interface tunnel 1`
	`Tunnel mode gre multipoint`
	`Tunnel source g0/0/1.100`
	`Tunnel key 12`
	`Ip address 10.1.12.1 255.255.255.0`
	
	Ottawa:
	`Interface tunnel 1`
	`Tunnel mode gre multipoint`
	`Tunnel source g0/0/1.101`
	`Tunnel key 12`
	`Ip address 10.1.12.2 255.255.255.0`
	
	Oshawa:
	`Interface tunnel 1`
	`Tunnel mode gre multipoint`
	`Tunnel source g0/0/1`
	`Tunnel key 12`
	`Ip address 10.1.12.3 255.255.255.0`

Summary:
Configuring a tunnel interface on the switches with GRE multipoint enabled as to enable the DMVPN interface without specifying destination addresses.

Screenshot:



```
Ottawa#show int tunnel1 | include Tunnel
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel linestate evaluation up
  Tunnel source 198.51.100.12 (GigabitEthernet0/0/1.101)
   Tunnel Subblocks:
         Tunnel1 source tracking subblock associated with GigabitEthernet0/0/1.101
  Tunnel protocol/transport multi-GRE/IP
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1472 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "BSN12_PROFILE")
```

```
Oshawa#show int tunnel1 | incl Tunnel
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel linestate evaluation up
  Tunnel source 203.0.113.12 (GigabitEthernet0/0/1)
   Tunnel Subblocks:
         Tunnel1 source tracking subblock associated with GigabitEthernet0/0/1
  Tunnel protocol/transport multi-GRE/IP
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1472 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "BSN12_PROFILE")
```

Task	Commands Used

Set the bandwidth of the tunnel interface to **1,000,000** and the delay to **10**.	`router(config-if)#`
	`Bandwidth 1000000`
	`Delay 10`

Summary:
In order to set bandwidth and delay to the needed numbers, we used the bandwidth and delay commands respectively.

Screenshot:

```
Toronto#show int tunnel1 | incl BW
  MTU 9972 bytes, BW 1000000 Kbit/sec, DLY 100 usec,
```

```
Ottawa#show int tunnel1 | incl BW
  MTU 9972 bytes, BW 1000000 Kbit/sec, DLY 100 usec,
```

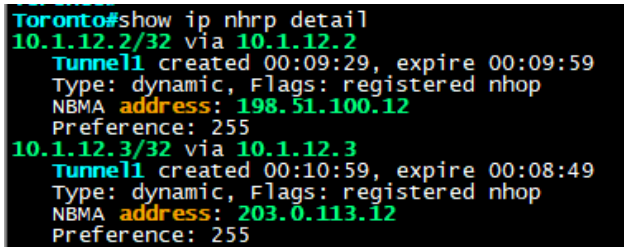
```
Oshawa#show int tunnel1 | incl BW
  MTU 9972 bytes, BW 1000000 Kbit/sec, DLY 100 usec,
```

Task	Commands Used

Configure NHRP in a hub-and-spoke topology, where **Toronto** is the hub:
• Use a network ID of xx.
• Set the NHRP authentication value as the first letter of each of your group member's names, in all capitals (for example, John, Mary, and Luke would use **JML** as the authentication password).
• On **Toronto**, configure NHRP to dynamically map multicast traffic for the tunnel endpoints.
• On **Ottawa** and **Oshawa**, configure **Toronto's** tunnel IP as the next hop server.
• On **Ottawa** and **Oshawa**, statically map **Toronto's** tunnel IP address to its Internet IP Address.
• On **Ottawa** and **Oshawa**, statically map multicast addresses to **Toronto's** Internet IP address.
• Configure **Toronto** to send NHRP redirects, and **Ottawa** and **Oshawa** to use NHRP shortcuts to enable Phase 3 DMVPN.
	Toronto:
	`Ip nhrp network-id 12`
	`Ip nhrp authentication BSN`
	`Ip nhrp map multicast dynamic`
	`Ip nhrp redirect`
	
	Ottawa:
	`Ip nhrp network-id 12`
	`Ip nhrp authentication BSN`
	`Ip nhrp nhs 10.1.12.1`
	`Ip nhrp map multicast 192.0.2.12`
	`Ip nhrp map 10.1.12.1 192.0.2.12`
	`Ip nhrp shortcut`
	
	Oshawa:
	`Ip nhrp network-id 12`
	`Ip nhrp authentication BSN`
	`Ip nhrp nhs 10.1.12.1`
	`Ip nhrp map multicast 192.0.2.12`
	`Ip nhrp map 10.1.12.1 192.0.2.12`
	`Ip nhrp shortcut`

Summary:
Configure NHRP with a network ID, BSN authentication, multicast dynamic, and redirect. NHRP is used alongside DMVPN to resolve the destination of the tunnel dynamically.

Screenshot:



Task	Commands Used

Secure the DMVPN tunnels using IPSec:
● Configure the following IKE policy:
○ ISAKMP policy number: xx
○ Hash: **SHA 512**
○ Encryption: **AES 256**
○ DH group number: **14**
○ Authentication: **Pre-shared Key**
○ Pre-shared key: Group member first initials and group number (e.g., **JML50**) for all addresses (**0.0.0.0**)
● Configure the following IPSec transform set:
○ Transform set name: Group member first initials and group number followed by "_TRANS" (e.g., **JML50_TRANS**)
○ Encryption: **AES 256**
○ Hash: **SHA 512 HMAC**
○ Use **Transport** mode
● Configure the following IPSec profile:
○ Profile name: Group member first initials and group number followed by "_PROFILE" (e.g., **JML50_PROFILE**).
	Toronto:
	`Crypto isakmp policy 12`
	`Hash sha512`
	`Encryption aes 256`
	`Group 14`
	`Authentication pre-share`
	`Exit`
	`Crypto isakmp key BSN12 address 0.0.0.0`
	`Crypto ipsec transform-set BSN12_TRANS esp-aes 256 esp-sha512-hmac`
	`Mode transport`
	`Exit`
	`Crypto ipsec profile BSN12_PROFILE`
	`Set transform-set BSN12_TRANS`
	`Exit`
	`Interface tunnel 1`
	`Tunnel protection ipsec profile BSN12_PROFILE`
	
	Ottawa:
	`Crypto isakmp policy 12`
	`Hash sha512`
	`Encryption aes 256`
	`Group 14`
	`Authentication pre-share`
	`Exit`
	`Crypto isakmp key BSN12 address 0.0.0.0`
	`Crypto ipsec transform-set BSN12_TRANS esp-aes 256 esp-sha512-hmac`
	`Mode transport`
	`Exit`
	`Crypto ipsec profile BSN12_PROFILE`
	`Set transform-set BSN12_TRANS`
	`Exit`
	`Interface tunnel 1`
	`Tunnel protection ipsec profile BSN12_PROFILE`
	
	Oshawa:
	`Crypto isakmp policy 12`
	`Hash sha512`
	`Encryption aes 256`

○ Use the transform set created previously. ○ Assign this profile as tunnel protection for interface Tunnel 1 on all three routers	Group 14 Authentication pre-share Exit Crypto isakmp key BSN12 address 0.0.0.0 Crypto ipsec transform-set BSN12_TRANS esp-aes 256 esp-sha512-hmac Mode transport Exit Crypto ipsec profile BSN12_PROFILE Set transform-set BSN12_TRANS Exit Interface tunnel 1 Tunnel protection ipsec profile BSN12_PROFILE

Summary:
In order to secure our DMVPN configuration, we used a variety of cryptography methods including IPSEC, SHA 256, and isakmp keys.

Screenshot:

```
Toronto#show crypto isakmp key
Keyring      Hostname/Address                          Preshared Key

default      0.0.0.0        [0.0.0.0]                  BSN12
Toronto#show crypto isakmp policy

Global IKE policy
Protection suite of priority 12
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard 2 (512 bit)
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #14 (2048 bit)
        lifetime:               86400 seconds, no volume limit
Toronto#show interfaces Tunnel1 | incl protection
  Tunnel protection via IPSec (profile "BSN12_PROFILE")
Toronto#show crypto ipsec profile
IPSEC profile BSN12_PROFILE
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                BSN12_TRANS:  { esp-256-aes esp-sha512-hmac  } ,
        }
```

```
Ottawa#show crypto isakmp key
Keyring      Hostname/Address                          Preshared Key

default      0.0.0.0        [0.0.0.0]                  BSN12
Ottawa#show crypto isakmp policy

Global IKE policy
Protection suite of priority 12
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard 2 (512 bit)
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #14 (2048 bit)
        lifetime:               86400 seconds, no volume limit
Ottawa#show interfaces Tunnel1 | incl protection
  Tunnel protection via IPSec (profile "BSN12_PROFILE")
Ottawa#show crypto ipsec profile
IPSEC profile BSN12_PROFILE
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                BSN12_TRANS:  { esp-256-aes esp-sha512-hmac  } ,
        }
```

```
Oshawa#show crypto isakmp key
Keyring      Hostname/Address                          Preshared Key

default      0.0.0.0        [0.0.0.0]                  BSN12
Oshawa#show crypto isakmp policy

Global IKE policy
Protection suite of priority 12
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard 2 (512 bit)
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #14 (2048 bit)
        lifetime:               86400 seconds, no volume limit
Oshawa#show interfaces Tunnel1 | incl protection
  Tunnel protection via IPSec (profile "BSN12_PROFILE")
Oshawa#show crypto ipsec profile
IPSEC profile BSN12_PROFILE
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                BSN12_TRANS:  { esp-256-aes esp-sha512-hmac  } ,
        }
```

task 6: configure EIGRP

Task	Commands Used
Enable EIGRP Named Mode on **Toronto**, **Ottawa**, and **Oshawa**. Name your EIGRP process **OntarioTechxx**.	`router eigrp OntarioTech12` `address-family ipv4 autonomous-system 12`

Summary:
To begin our EIGRP named mode configuration, we assigned the name OntarioTech12 with an AF-IPv4 AS number of 12

Screenshot:

```
Toronto#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(1.1.1.1)
```

```
Ottawa#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(3.3.3.3)
```

```
Oshawa#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(4.4.4.4)
```

Task	Commands Used

Use the following router IDs on each device:
• Toronto: **1.1.1.1**
• Ottawa: **3.3.3.3**
• Oshawa: **4.4.4.4**
	Toronto:
	`eigrp router-id 1.1.1.1`
	
	Ottawa:
	`eigrp router-id 3.3.3.3`
	
	Oshawa:
	`eigrp router-id 4.4.4.4`

Summary:
On each individual router, we configured a router-ID

Screenshot:

```
Toronto#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(1.1.1.1)
```

```
Ottawa#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(3.3.3.3)
```

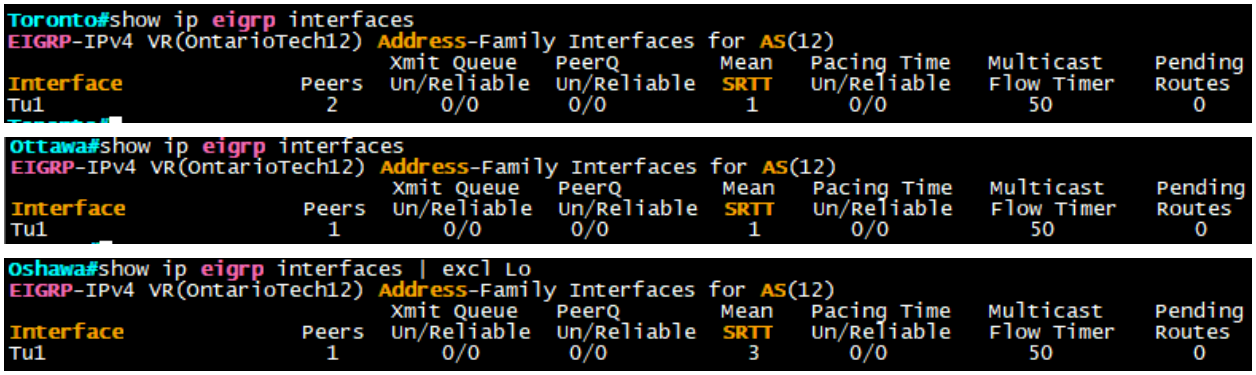
```
Oshawa#show ip eigrp topology | incl EIGRP
EIGRP-IPv4 VR(OntarioTech12) Topology Table for AS(12)/ID(4.4.4.4)
```

Task	Commands Used

On all three routers, enable EIGRP on the DMVPN tunnel interfaces.	Toronto:
	`Network 10.1.12.0 255.255.255.0`
	
	Ottawa:
	`network 10.1.12.0 255.255.255.0`
	
	Oshawa:
	`Network 10.1.12.0 255.255.255.0`

Summary:
On each respective router, in order to enable EIGRP on the DMVPN tunnel interfaces,

Screenshot:



Task	Commands Used

On **Oshawa**, enable EIGRP on loopbacks 100-103.	`network 172.16.4.1`
	`network 172.16.41.1`
	`network 172.16.42.1`
	`network 172.16.43.1`

Summary:
In order to enable EIGRP on the loopbacks on Oshawa, the command network <ip-address> was used with the respective loopback ip addresses.

Screenshot:

```
Oshawa#show ip eigrp interfaces | excl Tu
EIGRP-IPv4 VR(OntarioTech12) Address-Family Interfaces for AS(12)
                              Xmit Queue   PeerQ        Mean   Pacing Time   Multicast    Pending
Interface              Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Lo100                    0        0/0          0/0        0        0/0           0           0
Lo101                    0        0/0          0/0        0        0/0           0           0
Lo102                    0        0/0          0/0        0        0/0           0           0
Lo103                    0        0/0          0/0        0        0/0           0           0
```

Task	Commands Used
On **Toronto**, configure a summary route for **172.16.0.0/16** on the tunnel interface to trigger the DMVPN spokes to perform next-hop resolution for any addresses in the LAN subnets.	`summary-address 172.16.0.0 255.255.0.0`

Summary:
In the routing table, we configured a static summary route to the 172.16.0.0/16 network to trigger the DMVPN spokes to perform NH resolution.

Screenshot:

```
Toronto#show ip route eigrp | begin Gate
Gateway of last resort is 192.0.2.254 to network 0.0.0.0

      172.16.0.0/16 is variably subnetted, 11 subnets, 3 masks
D        172.16.0.0/16 is a summary, 00:10:43, Null0
D        172.16.4.0/24 [90/56960] via 10.1.12.3, 00:10:43, Tunnel1
D        172.16.41.0/24 [90/56960] via 10.1.12.3, 00:10:43, Tunnel1
D        172.16.42.0/24 [90/56960] via 10.1.12.3, 00:10:43, Tunnel1
D        172.16.43.0/24 [90/56960] via 10.1.12.3, 00:10:43, Tunnel1
```

task 7: configure redistribution

Task	Commands Used

Perform **mutual redistribution** between EIGRP and OSPF on both **Toronto** and **Ottawa**. For EIGRP metrics use the following values:
• Bandwidth: **1,000,000 Kbps**
• Delay: **10 µsec**
• Reliability: **255/255**
• Load: **1/255**
• MTU: **1500**
	Toronto:
	`router eigrp OntarioTech12`
	`address-family ipv4 autonomous-system 12`
	`topology base`
	`redistribute ospfv3 12 metric 1000000 10 255 1 1500`
	
	`router ospfv3 12`
	`address-family ipv4`
	`redistribute eigrp 12 metric-type 1`
	
	Ottawa:
	`router eigrp OntarioTech12`
	`address-family ipv4 autonomous-system 12`
	`topology base`
	`redistribute ospfv3 12 metric 1000000 10 255 1 1500`
	
	`router ospfv3 12`
	`address-family ipv4`
	`redistribute eigrp 12 metric-type 1`
	`redistribute bgp 64612`

Summary:
In both the EIGRP and OSPF configuration modes, using the *`redistribute ospfv3 [process ID]`* command, with provided metrics to dynamically redistribute routes between EIGRP named mode and OSPFv3

Screenshot:

```
Ottawa#show ip route | begin Gateway
Gateway of last resort is 198.51.100.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 198.51.100.254
      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [20/0] via 10.202.20.1, 00:01:06
      3.0.0.0/32 is subnetted, 1 subnets
C        3.3.3.3 is directly connected, Loopback0
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C        10.1.12.0/24 is directly connected, Tunnel1
L        10.1.12.2/32 is directly connected, Tunnel1
B        10.202.10.0/30 [20/0] via 10.202.20.1, 00:01:06
C        10.202.20.0/30 is directly connected, GigabitEthernet0/0/0
L        10.202.20.2/32 is directly connected, GigabitEthernet0/0/0
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
D        172.16.0.0/16 [90/107520] via 10.1.12.1, 00:01:47, Tunnel1
C        172.16.3.0/24 is directly connected, GigabitEthernet0/0/1.10
L        172.16.3.1/32 is directly connected, GigabitEthernet0/0/1.10
O IA     172.16.31.0/24
           [110/101] via 172.16.3.2, 00:01:06, GigabitEthernet0/0/1.10
O IA     172.16.32.0/24
           [110/101] via 172.16.3.2, 00:01:06, GigabitEthernet0/0/1.10
O IA     172.16.33.0/24
           [110/101] via 172.16.3.2, 00:01:06, GigabitEthernet0/0/1.10
      198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.51.100.0/24 is directly connected, GigabitEthernet0/0/1.101
L        198.51.100.12/32 is directly connected, GigabitEthernet0/0/1.101
```

```
Toronto#show ip route | begin Gate
Gateway of last resort is 192.0.2.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.0.2.254
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      3.0.0.0/32 is subnetted, 1 subnets
B        3.3.3.3 [20/0] via 10.202.10.1, 00:00:50
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C        10.1.12.0/24 is directly connected, Tunnel1
L        10.1.12.1/32 is directly connected, Tunnel1
C        10.202.10.0/30 is directly connected, GigabitEthernet0/0/0
L        10.202.10.2/32 is directly connected, GigabitEthernet0/0/0
B        10.202.20.0/30 [20/0] via 10.202.10.1, 00:00:50
      172.16.0.0/16 is variably subnetted, 15 subnets, 3 masks
D        172.16.0.0/16 is a summary, 00:00:51, Null0
C        172.16.1.0/24 is directly connected, GigabitEthernet0/0/1.10
L        172.16.1.1/32 is directly connected, GigabitEthernet0/0/1.10
B        172.16.3.0/24 [20/0] via 10.202.10.1, 00:00:10
D        172.16.4.0/24 [90/56960] via 10.1.12.3, 00:21:02, Tunnel1
O IA     172.16.11.0/24
           [110/101] via 172.16.1.2, 00:00:50, GigabitEthernet0/0/1.10
O IA     172.16.12.0/24
           [110/101] via 172.16.1.2, 00:00:50, GigabitEthernet0/0/1.10
O IA     172.16.13.0/24
           [110/101] via 172.16.1.2, 00:00:50, GigabitEthernet0/0/1.10
B        172.16.31.0/24 [20/0] via 10.202.10.1, 00:00:10
B        172.16.32.0/24 [20/0] via 10.202.10.1, 00:00:10
B        172.16.33.0/24 [20/0] via 10.202.10.1, 00:00:10
D        172.16.41.0/24 [90/56960] via 10.1.12.3, 00:21:02, Tunnel1
D        172.16.42.0/24 [90/56960] via 10.1.12.3, 00:21:02, Tunnel1
D        172.16.43.0/24 [90/56960] via 10.1.12.3, 00:21:02, Tunnel1
O IA     172.16.112.0/24
           [110/101] via 172.16.1.2, 00:00:50, GigabitEthernet0/0/1.10
      192.0.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.0.2.0/24 is directly connected, GigabitEthernet0/0/1.100
L        192.0.2.12/32 is directly connected, GigabitEthernet0/0/1.100
```

```
Ottawa#show ospfv3 database

          OSPFv3 12 address-family ipv4 (router-id 3.3.3.3)

              Router Link States (Area 0)

ADV Router      Age         Seq#       Fragment ID  Link count  Bits
 3.3.3.3        209         0x80000006 0            1           E
 30.30.30.30    299         0x80000004 0            1           B

              Net Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Rtr count
 30.30.30.30    313         0x80000001 46         2

              Inter Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Prefix
 30.30.30.30    353         0x80000001 172.16.33.0/24
 30.30.30.30    353         0x80000001 172.16.32.0/24
 30.30.30.30    353         0x80000001 172.16.31.0/24

              Link (Type-8) Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Interface
 3.3.3.3        375         0x80000001 12         Gi0/0/1.10
 30.30.30.30    353         0x80000001 46         Gi0/0/1.10

              Intra Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Ref-lstype  Ref-LSID
 30.30.30.30    313         0x80000001 47104      0x2002      46

              Type-5 AS External Link States

ADV Router      Age         Seq#       Prefix
 3.3.3.3        209         0x80000001 1.1.1.1/32
 3.3.3.3        209         0x80000001 10.202.10.0/30
 3.3.3.3        208         0x80000001 3.3.3.3/32
 3.3.3.3        208         0x80000001 10.1.12.0/24
 3.3.3.3        208         0x80000001 172.16.0.0/16
```

```
Toronto#show ospfv3 dat

          OSPFv3 12 address-family ipv4 (router-id 1.1.1.1)

              Router Link States (Area 0)

ADV Router      Age         Seq#       Fragment ID  Link count  Bits
 1.1.1.1        224         0x80000006 0            1           E
 10.10.10.10    312         0x80000004 0            1           B

              Net Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Rtr count
 10.10.10.10    326         0x80000001 44         2

              Inter Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Prefix
 10.10.10.10    366         0x80000001 172.16.112.0/24
 10.10.10.10    366         0x80000001 172.16.13.0/24
 10.10.10.10    366         0x80000001 172.16.12.0/24
 10.10.10.10    366         0x80000001 172.16.11.0/24

              Link (Type-8) Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Interface
 1.1.1.1        388         0x80000001 13         Gi0/0/1.10
 10.10.10.10    366         0x80000001 44         Gi0/0/1.10

              Intra Area Prefix Link States (Area 0)

ADV Router      Age         Seq#       Link ID    Ref-lstype  Ref-LSID
 10.10.10.10    326         0x80000001 45056      0x2002      44

              Type-5 AS External Link States

ADV Router      Age         Seq#       Prefix
 1.1.1.1        223         0x80000001 10.1.12.0/24
 1.1.1.1        223         0x80000001 172.16.0.0/16
 1.1.1.1        223         0x80000001 172.16.4.0/24
 1.1.1.1        223         0x80000001 172.16.41.0/24
 1.1.1.1        223         0x80000001 172.16.42.0/24
 1.1.1.1        223         0x80000001 172.16.43.0/24
```

Task	Commands Used

Perform **mutual redistribution** between BGP and OSPF on both **Toronto** and **Ottawa**.	Toronto
	`router bgp 64612`
	`address-family ipv4`
	`redistribute ospf 12`
	`bgp redistribute-internal`
	
	Ottawa:
	`router bgp 64612`
	`address-family ipv4`
	`redistribute ospfv3 12`
	`bgp redistribute-internal`

Summary:
Similar to the previous step, we used the *`redistribute ospfv3 [process ID]`* command within the BGP router configuration to redistribute routes between both protocols.

Screenshot:

refer to step above for screenshots of `show ip route`, `show ospfv3 database`
refer to appendix for screenshots of `show ip protocols`

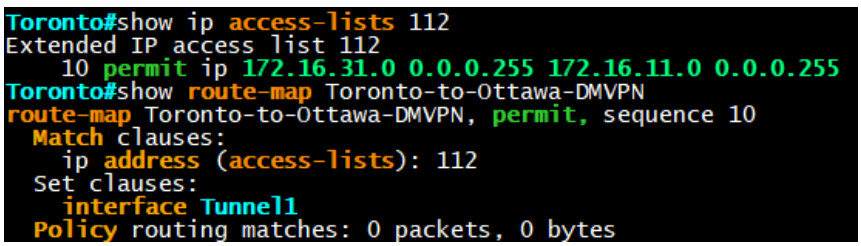
task 8: configure policy-based routing (PBR)

Task	Commands Used

Configure policy-based routing (PBR)
- Packets from **172.16.11.0/24** going to **172.16.31.0/24** should always use the DMVPN tunnel to **Ottawa**, rather than crossing the MPLS WAN. Configure this on **Toronto**.
- Packets from **172.16.31.0/24** going to **172.16.11.0/24** should also always use the DMVPN tunnel to **Toronto** rather than crossing the MPLS WAN. Configure this on **Ottawa**.
	Toronto:
	`Ip access-list extended 112`
	`Remark ACL matches TOR-SW1 Lo101 traffic`
	`Permit ip 172.16.31.0 0.0.0.255 172.16.11.0 0.0.0.255`
	`route-map Toronto-to-Ottawa-DMVPN`
	`Match ip address 112`
	`Set interface tunnel1`
	`Interface g0/0/1.10`
	`Ip policy route-map Toronto-to-Ottawa-DMVPN`
	
	Ottawa:
	`Ip access-list extended 112`
	`Remark ACL matches OTT-SW1 Lo101 traffic`
	`Permit ip 172.16.11.0 0.0.0.255 172.16.31.0 0.0.0.255`
	`exit`
	`Route-map Ottawa-to-Toronto-DMVPN`
	`Match ip address 112`
	`Set interface tunnel1`
	`exit`
	`Int g0/0/1.10`
	`Ip policy route-map Ottawa-to-Toronto-DMVPN`

Summary:
Using the access list “Toronto-to-Ottawa-DMVPN”, applied with a route map on the Tunnel1 interface, we perform policy based routing which permits traffic depending on its source and destination addresses.

Screenshot:



task 9: route filtering

Task	Commands Used

Filter the Loopback **1xx** subnet from being advertised via BGP using the following parameters:	`ip prefix-list TORONTO permit 172.16.0.0/20 ge 24`
	`ip prefix-list TORONTO permit 1.1.1.1/32`
	`router bgp 64612`
	`address-family ipv4`
	`neighbor 10.202.10.1 distribute-list TORONTO out`

Summary:
Using the prefix list “Toronto”, we configured two permit statements for the address range 172.16.0.0/20 ge 24 and 1.1.1.1/32. We then configured neighbor 10.202.10.1 to apply this prefix list on outbound traffic.

Screenshot:

```
Toronto#show ip prefix-list TORONTO
ip prefix-list TORONTO: 2 entries
   seq 5 permit 172.16.0.0/20 ge 24
   seq 10 permit 1.1.1.1/32
Toronto#ping 172.16.11.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Toronto#ping 172.16.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Toronto#ping 172.16.13.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Toronto#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Toronto#ping 172.16.112.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Toronto#ping 172.16.113.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.113.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

(172.16.113.1 is out of the scope of the prefix list, therefore, it will not have connectivity)

task 10: testing

Task	Commands Used

To verify connectivity, execute the following TCL script on all devices **except ISP**	`Tclsh`
	`foreach address {`
	`172.16.1.1`
	`10.1.12.1`
	`172.16.3.1`
	`10.1.12.2`
	`172.16.4.1`
	`172.16.41.1`
	`172.16.42.1`
	`172.16.43.1`
	`10.1.12.3`
	`172.16.1.2`
	`172.16.11.1`
	`172.16.12.1`
	`172.16.13.1`
	`172.16.112.1`
	`172.16.3.2`
	`172.16.31.1`
	`172.16.32.1`
	`172.16.33.1`
	`} { ping $address }`
	`tclquit`

Summary:
Using a TCL script, we individually pinged each destination address to test connectivity between all the links in the network

Screenshot:

```
Toronto#Tclsh
Toronto(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Ottawa(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/48 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Ottawa(tcl)#
```

```
Oshawa#Tclsh
Oshawa(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Oshawa(tcl)#
```

```
TOR-SW1#Tclsh
TOR-SW1(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
TOR-SW1(tcl)#
```

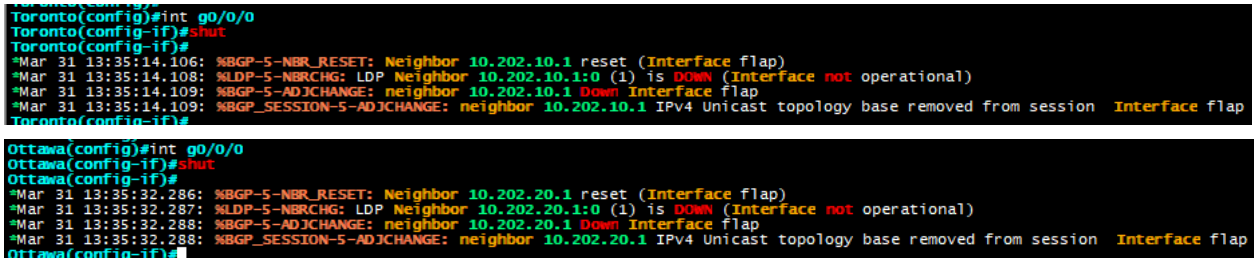
```
OTT-SW1(tcl)#tclsh
OTT-SW1(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
OTT-SW1(tcl)#
```

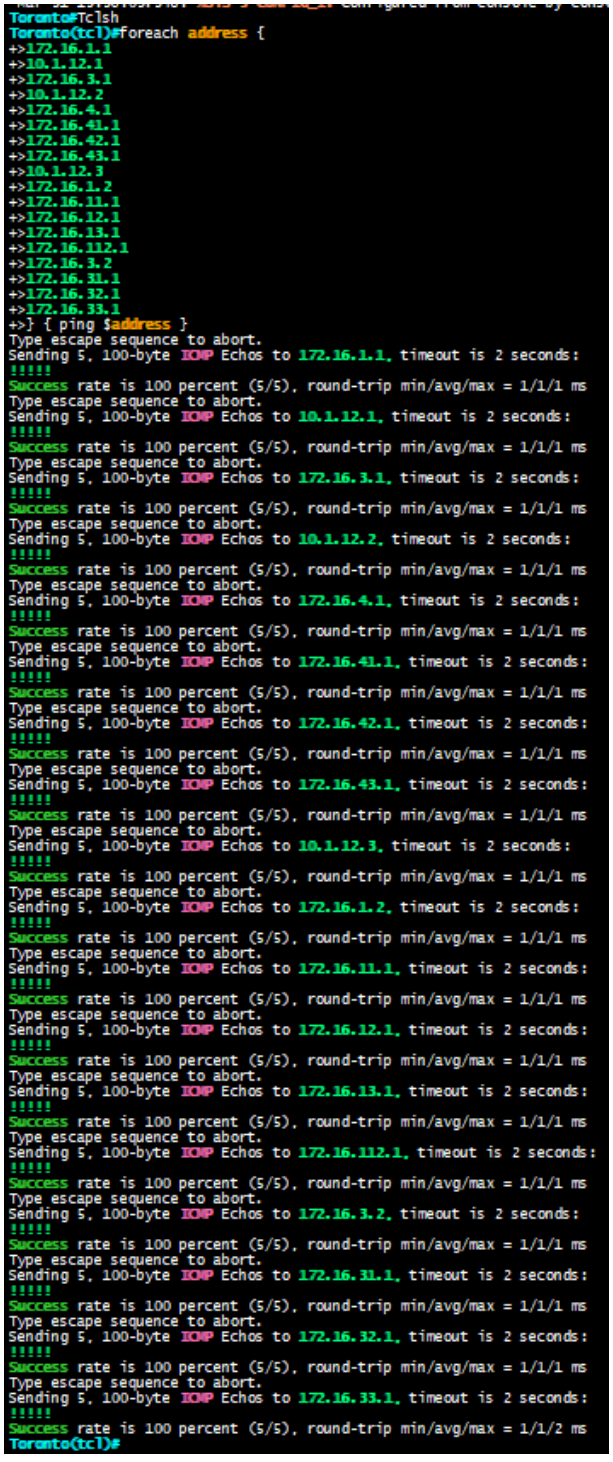
Task	Commands Used

Shut down the MPLS WAN interfaces on **Toronto** and **Ottawa** and run the TCL script again on all devices **except ISP**. The pings should still all be successful.	Toronto: Shut down MPLS WAN
	`Interface g0/0/0`
	`shut`
	Ottawa: Shut down MPLS WAN
	`Interface g0/0/0`
	`Shut`
	
	Redo traceroute commands again[refer to commands here/hyperlink]

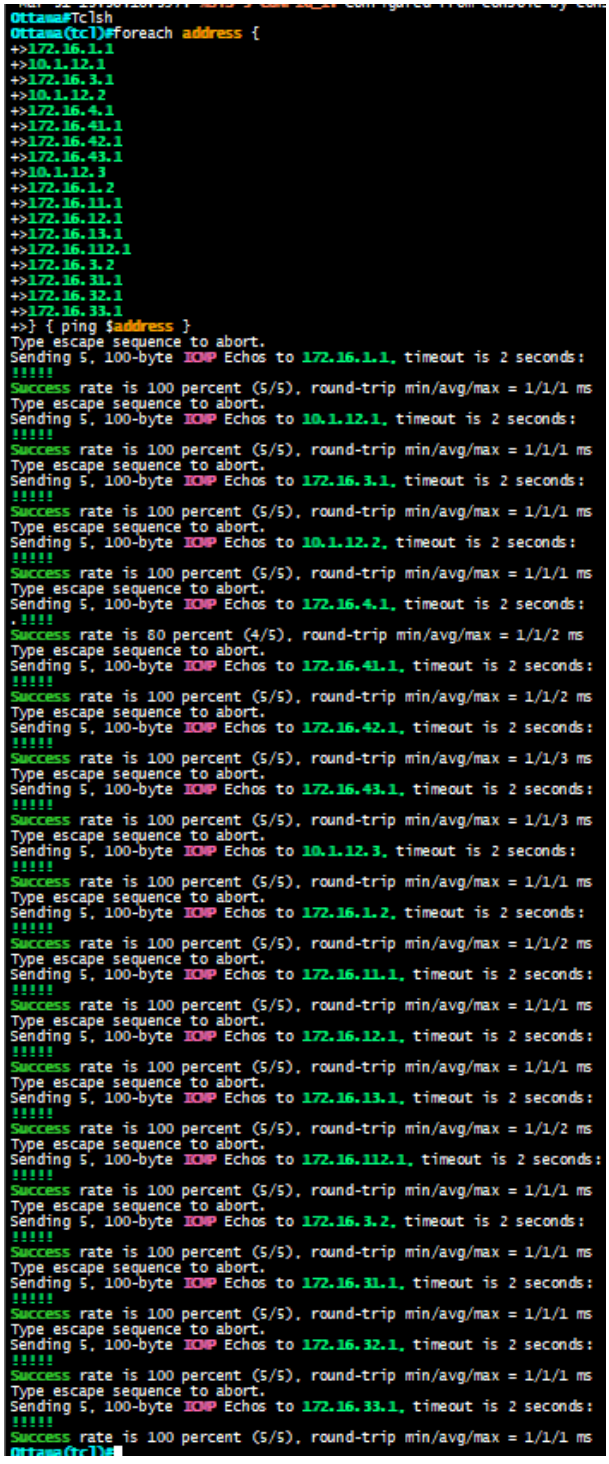
Summary:
To further test the connectivity of the links, we shut down interfaces within the MPLS WAN and ran the same TCP script as last.

Screenshot:





```
Toronto#Tclsh
Toronto(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Toronto(tcl)#
```



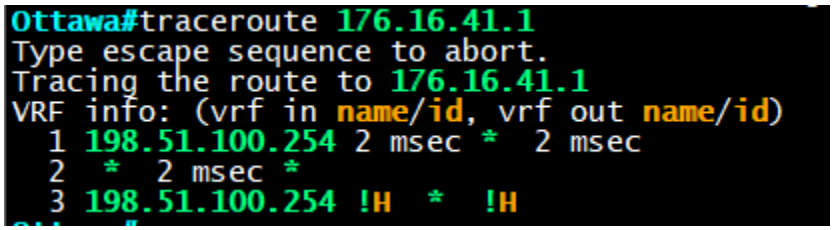
```
Ottawa#Tclsh
Ottawa(tcl)#foreach address {
+>172.16.1.1
+>10.1.12.1
+>172.16.3.1
+>10.1.12.2
+>172.16.4.1
+>172.16.41.1
+>172.16.42.1
+>172.16.43.1
+>10.1.12.3
+>172.16.1.2
+>172.16.11.1
+>172.16.12.1
+>172.16.13.1
+>172.16.112.1
+>172.16.3.2
+>172.16.31.1
+>172.16.32.1
+>172.16.33.1
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.41.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.42.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.43.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.112.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.31.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.33.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Ottawa(tcl)#
```

Task	Commands Used

Perform a traceroute from **Ottawa** to the **172.16.41.1** IP on **Oshawa** to show that the packets go directly to **Oshawa**, rather than through the DMVPN hub at **Toronto**	Toronto: Enable MPLS WAN
	`Interface g0/0/0`
	`No shut`
	Ottawa: Enable MPLS WAN
	`Interface g0/0/0`
	`No Shut`
	
	Ottawa: Traceroute
	`traceroute 176.16.41.1`

Summary:
To test the DMVPN configuration, we issued a traceroute from Ottawa to Oshawa using the DMVPN tunnel.

Screenshot:



Task	Commands Used

Perform a traceroute from **Tor-Sw1** to **172.16.31.1** using a source address of **172.16.11.1** to show that these packets are policy routed across the DMVPN. Perform a traceroute from **Tor-Sw1** to **172.16.32.1** using a source address of **172.16.11.1** to show that these packets are routed normally across the MPLS WAN.	TOR-SW1: Traceroute
	`Tclsh`
	`foreach address {`
	`172.16.31.1`
	`172.16.32.1`
	`} { traceroute $address source`
	`172.16.11.1 }`
	`tclquit`

Summary:
To further test the DMVPN configuration by issuing a traceroute from Toronto Switch 1 to 172.16.31.1 using the source address 172.16.11.1.

Screenshot:

```
TOR-SW1#Tclsh
TOR-SW1(tcl)#foreach address {
+>172.16.31.1
+>172.16.32.1
+>} { traceroute $address source 172.16.11.1 }
Type escape sequence to abort.
Tracing the route to OTT-SW1 (172.16.31.1)
VRF info: (vrf in name/id, vrf out name/id)
  1 TORONTO (172.16.1.1) 1 msec 0 msec 1 msec
  2  *  *  *
  3 OTTAWA (10.202.20.2) 2 msec 1 msec 2 msec
  4 OTT-SW1 (172.16.3.2) 1 msec *  2 msec
Type escape sequence to abort.
Tracing the route to OTT-SW1 (172.16.32.1)
VRF info: (vrf in name/id, vrf out name/id)
  1 TORONTO (172.16.1.1) 1 msec 1 msec 1 msec
  2  *  *  *
  3 OTTAWA (10.202.20.2) 1 msec 2 msec 1 msec
  4 OTT-SW1 (172.16.3.2) 2 msec *  2 msec
```

Task	Commands Used

Perform a traceroute from **Ott-Sw1** to **172.16.11.1** using a source address of **172.16.31.1** to show that these packets are policy routed across the DMVPN. Perform a traceroute from **Ott-Sw1** to **172.16.12.1** using a source address of **172.16.31.1** to show that these packets are routed normally across the MPLS WAN.	OTT-SW1: Traceroute
	`Tclsh`
	`foreach address {`
	`172.16.11.1`
	`172.16.12.1`
	`} { traceroute $address source`
	`172.16.31.1 }`
	`Tclquit`

Summary:
As with the last step, we repeated the test on Ottawa Switch 1

Screenshot:

```
OTT-SW1(tcl)#Tclsh
OTT-SW1(tcl)#foreach address {
+>172.16.11.1
+>172.16.12.1
+>} { traceroute $address source 172.16.31.1 }
Type escape sequence to abort.
Tracing the route to TOR-SW1 (172.16.11.1)
VRF info: (vrf in name/id, vrf out name/id)
  1 OTTAWA (172.16.3.1) 1 msec 0 msec 0 msec
  2 10.1.12.1 1 msec 1 msec 1 msec
  3 TOR-SW1 (172.16.1.2) 2 msec *  2 msec
Type escape sequence to abort.
Tracing the route to TOR-SW1 (172.16.12.1)
VRF info: (vrf in name/id, vrf out name/id)
  1 OTTAWA (172.16.3.1) 1 msec 1 msec 1 msec
  2 10.1.12.1 1 msec 1 msec 1 msec
  3  *
    TOR-SW1 (172.16.1.2) 2 msec *
```

V. Additional Changes

1. **Configure non-cisco proprietary routing protocol on DMVPN enabled devices**
   For scalability reasons, EIGRP should not be the protocol of choice for the DMVPN network. Since EIGRP is Cisco-proprietary, the scalability of the network is limited to only Cisco devices. For this reason, OSPF or BGP should be used instead of EIGRP on a multi-campus network as it would be easier to add other vendor devices to the network, such as Juniper, Aruba or Arista.

2. **Configure NTP (Network Time Protocol) server**
   Configure it to go to the government NRC stratum server. This would allow for accurate clock synchronization to simplify network monitoring and accounting.

3. **Implement agent-based automation system in the form of Puppet**
   Puppet offers an automated way to manage,configure networks and it comes with its own modules for many different vedoors and devices, in our case puppet offers Cisco Catalyst switches a module called "`cisco_ios`", which contains many different manifests to configure the switch. For scalability reasons, network automation should be used to help limit the room for human error in the configuration. In the future when the network needs to expand, or change the routing protocol used on the DMVPN tunnel from EIGRP to OSPF. Furthermore, an agent-based solution would be more useful in a large-scale network such as this one since it relies on a dedicated appliance. Using a dedicated appliance would ensure that configuration changes could be made by any network administrator.

4. **Add a redundant link between the Toronto and Ottawa branch routers**
   Adding a redundant link on the large Toronto and Ottawa branch connections between the switch and the router would help to prevent unnecessary network downtime.

5. **Disable CDP and LLDP**
   For security reasons, Cisco discovery protocol, or CDP, should be disabled as the information transferred by CDP could expose a network topology. The information exposed includes, but is not limited to, hostname, device model number and IP address. This information can be used to attack the network using many known tools built-in to the Kali Linux suite. Furthermore, LLDP should also be disabled for the same reason.

6. **AAA model**
   Implementing AAA would allow the campus network access to be controlled securely and efficiently. Implementing AAA would also simplify network monitoring and access-control. Secure authentication can be done on the network by using a RADIUS or TACACS+ server.

VI. Appendix

```
ISP#show ip route | begin Gateway
Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
C        2.2.2.2 is directly connected, Loopback0
ISP#show ip route vrf ISP-A | begin Gateway
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [20/0] via 10.202.10.2, 00:06:09
      3.0.0.0/32 is subnetted, 1 subnets
B        3.3.3.3 [20/0] via 10.202.20.2 (ISP-B), 00:06:00
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.202.10.0/30 is directly connected, GigabitEthernet0/0/0
L        10.202.10.1/32 is directly connected, GigabitEthernet0/0/0
B        10.202.20.0/30 is directly connected, 00:06:03, GigabitEthernet0/0/1
L        10.202.20.1/32 is directly connected, GigabitEthernet0/0/1
      172.16.0.0/24 is subnetted, 4 subnets
B        172.16.3.0 [20/0] via 10.202.20.2 (ISP-B), 00:06:00
B        172.16.31.0 [20/101] via 10.202.20.2 (ISP-B), 00:06:00
B        172.16.32.0 [20/101] via 10.202.20.2 (ISP-B), 00:06:00
B        172.16.33.0 [20/101] via 10.202.20.2 (ISP-B), 00:06:00
ISP#show ip route vrf ISP-B | begin Gateway
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [20/0] via 10.202.10.2 (ISP-A), 00:06:13
      3.0.0.0/32 is subnetted, 1 subnets
B        3.3.3.3 [20/0] via 10.202.20.2, 00:06:04
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B        10.202.10.0/30 is directly connected, 00:06:17, GigabitEthernet0/0/0
L        10.202.10.1/32 is directly connected, GigabitEthernet0/0/0
C        10.202.20.0/30 is directly connected, GigabitEthernet0/0/1
L        10.202.20.1/32 is directly connected, GigabitEthernet0/0/1
      172.16.0.0/24 is subnetted, 4 subnets
B        172.16.3.0 [20/0] via 10.202.20.2, 00:06:04
B        172.16.31.0 [20/101] via 10.202.20.2, 00:06:04
B        172.16.32.0 [20/101] via 10.202.20.2, 00:06:04
B        172.16.33.0 [20/101] via 10.202.20.2, 00:06:04
```

```
Toronto#show ip route | begin Gateway
Gateway of last resort is 192.0.2.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.0.2.254
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      3.0.0.0/32 is subnetted, 1 subnets
B        3.3.3.3 [20/0] via 10.202.10.1, 00:04:24
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C        10.1.12.0/24 is directly connected, Tunnel1
L        10.1.12.1/32 is directly connected, Tunnel1
C        10.202.10.0/30 is directly connected, GigabitEthernet0/0/0
L        10.202.10.2/32 is directly connected, GigabitEthernet0/0/0
B        10.202.20.0/30 [20/0] via 10.202.10.1, 00:04:27
      172.16.0.0/16 is variably subnetted, 15 subnets, 3 masks
D        172.16.0.0/16 is a summary, 00:39:45, Null0
C        172.16.1.0/24 is directly connected, GigabitEthernet0/0/1.10
L        172.16.1.1/32 is directly connected, GigabitEthernet0/0/1.10
B        172.16.3.0/24 [20/0] via 10.202.10.1, 00:04:24
D        172.16.4.0/24 [90/56960] via 10.1.12.3, 00:34:50, Tunnel1
O IA     172.16.11.0/24
           [110/101] via 172.16.1.2, 00:35:24, GigabitEthernet0/0/1.10
O IA     172.16.12.0/24
           [110/101] via 172.16.1.2, 00:35:24, GigabitEthernet0/0/1.10
O IA     172.16.13.0/24
           [110/101] via 172.16.1.2, 00:35:24, GigabitEthernet0/0/1.10
B        172.16.31.0/24 [20/0] via 10.202.10.1, 00:04:24
B        172.16.32.0/24 [20/0] via 10.202.10.1, 00:04:24
B        172.16.33.0/24 [20/0] via 10.202.10.1, 00:04:24
D        172.16.41.0/24 [90/56960] via 10.1.12.3, 00:34:50, Tunnel1
D        172.16.42.0/24 [90/56960] via 10.1.12.3, 00:34:50, Tunnel1
D        172.16.43.0/24 [90/56960] via 10.1.12.3, 00:34:50, Tunnel1
O IA     172.16.112.0/24
           [110/101] via 172.16.1.2, 00:35:24, GigabitEthernet0/0/1.10
      192.0.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.0.2.0/24 is directly connected, GigabitEthernet0/0/1.100
L        192.0.2.12/32 is directly connected, GigabitEthernet0/0/1.100
```

```
Ottawa#show ip route | begin Gateway
Gateway of last resort is 198.51.100.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 198.51.100.254
      1.0.0.0/32 is subnetted, 1 subnets
B        1.1.1.1 [20/0] via 10.202.20.1, 00:06:26
      3.0.0.0/32 is subnetted, 1 subnets
C        3.3.3.3 is directly connected, Loopback0
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C        10.1.12.0/24 is directly connected, Tunnel1
L        10.1.12.2/32 is directly connected, Tunnel1
B        10.202.10.0/30 [20/0] via 10.202.20.1, 00:06:26
C        10.202.20.0/30 is directly connected, GigabitEthernet0/0/0
L        10.202.20.2/32 is directly connected, GigabitEthernet0/0/0
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
D        172.16.0.0/16 [90/107520] via 10.1.12.1, 00:24:14, Tunnel1
C        172.16.3.0/24 is directly connected, GigabitEthernet0/0/1.10
L        172.16.3.1/32 is directly connected, GigabitEthernet0/0/1.10
O IA     172.16.31.0/24
           [110/101] via 172.16.3.2, 00:23:36, GigabitEthernet0/0/1.10
O IA     172.16.32.0/24
           [110/101] via 172.16.3.2, 00:23:36, GigabitEthernet0/0/1.10
O IA     172.16.33.0/24
           [110/101] via 172.16.3.2, 00:23:36, GigabitEthernet0/0/1.10
      198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.51.100.0/24 is directly connected, GigabitEthernet0/0/1.101
L        198.51.100.12/32 is directly connected, GigabitEthernet0/0/1.101
```

```
Oshawa#show ip route | begin Gateway
Gateway of last resort is 203.0.113.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.254
      4.0.0.0/32 is subnetted, 1 subnets
C        4.4.4.4 is directly connected, Loopback0
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.12.0/24 is directly connected, Tunnel1
L        10.1.12.3/32 is directly connected, Tunnel1
      172.16.0.0/16 is variably subnetted, 9 subnets, 3 masks
D        172.16.0.0/16 [90/107520] via 10.1.12.1, 00:37:30, Tunnel1
C        172.16.4.0/24 is directly connected, Loopback100
L        172.16.4.1/32 is directly connected, Loopback100
C        172.16.41.0/24 is directly connected, Loopback101
L        172.16.41.1/32 is directly connected, Loopback101
C        172.16.42.0/24 is directly connected, Loopback102
L        172.16.42.1/32 is directly connected, Loopback102
C        172.16.43.0/24 is directly connected, Loopback103
L        172.16.43.1/32 is directly connected, Loopback103
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C        203.0.113.0/24 is directly connected, GigabitEthernet0/0/1
L        203.0.113.12/32 is directly connected, GigabitEthernet0/0/1
```

```
TOR-SW1#show ip route | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 1 subnets
O E1     10.1.12.0 [110/120] via 172.16.1.1, 00:38:22, Vlan10
      172.16.0.0/16 is variably subnetted, 15 subnets, 3 masks
O E1     172.16.0.0/16 [110/120] via 172.16.1.1, 00:38:22, Vlan10
C        172.16.1.0/24 is directly connected, Vlan10
L        172.16.1.2/32 is directly connected, Vlan10
O E1     172.16.4.0/24 [110/120] via 172.16.1.1, 00:37:50, Vlan10
C        172.16.11.0/24 is directly connected, Loopback101
L        172.16.11.1/32 is directly connected, Loopback101
C        172.16.12.0/24 is directly connected, Loopback102
L        172.16.12.1/32 is directly connected, Loopback102
C        172.16.13.0/24 is directly connected, Loopback103
L        172.16.13.1/32 is directly connected, Loopback103
O E1     172.16.41.0/24 [110/120] via 172.16.1.1, 00:37:50, Vlan10
O E1     172.16.42.0/24 [110/120] via 172.16.1.1, 00:37:50, Vlan10
O E1     172.16.43.0/24 [110/120] via 172.16.1.1, 00:37:50, Vlan10
C        172.16.112.0/24 is directly connected, Loopback112
L        172.16.112.1/32 is directly connected, Loopback112
```

```
OTT-SW1#show ip route | begin Gateway
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O E1     1.1.1.1 [110/101] via 172.16.3.1, 00:07:46, Vlan10
      3.0.0.0/32 is subnetted, 1 subnets
O E1     3.3.3.3 [110/101] via 172.16.3.1, 00:24:50, Vlan10
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O E1     10.1.12.0/24 [110/120] via 172.16.3.1, 00:24:50, Vlan10
O E1     10.202.10.0/30 [110/101] via 172.16.3.1, 00:07:46, Vlan10
      172.16.0.0/16 is variably subnetted, 9 subnets, 3 masks
O E1     172.16.0.0/16 [110/120] via 172.16.3.1, 00:24:50, Vlan10
C        172.16.3.0/24 is directly connected, Vlan10
L        172.16.3.2/32 is directly connected, Vlan10
C        172.16.31.0/24 is directly connected, Loopback101
L        172.16.31.1/32 is directly connected, Loopback101
C        172.16.32.0/24 is directly connected, Loopback102
L        172.16.32.1/32 is directly connected, Loopback102
C        172.16.33.0/24 is directly connected, Loopback103
L        172.16.33.1/32 is directly connected, Loopback103
```

```
Toronto#show ip int b
Any interface listed with OK? value "NO" does not have a valid configuration

Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   10.202.10.2     YES manual up                    up
GigabitEthernet0/0/1   unassigned      YES unset  up                    up
Gi0/0/1.10             172.16.1.1      YES manual up                    up
Gi0/0/1.100            192.0.2.12      YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
Serial0/1/0            unassigned      NO  unset  up                    down
Serial0/1/1            unassigned      NO  unset  up                    down
GigabitEthernet0       unassigned      YES manual administratively down down
Loopback0              1.1.1.1         YES manual up                    up
Tunnel1                10.1.12.1       YES manual up                    up
```

```
ISP#show ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   10.202.10.1     YES manual up                    up
GigabitEthernet0/0/1   10.202.20.1     YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
GigabitEthernet0       unassigned      YES manual administratively down down
Loopback0              2.2.2.2         YES manual up                    up
ISP#
```

```
Ottawa#show ip int br
Any interface listed with OK? value "NO" does not have a valid configuration

Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   10.202.20.2     YES manual up                    up
GigabitEthernet0/0/1   unassigned      YES unset  up                    up
Gi0/0/1.10             172.16.3.1      YES manual up                    up
Gi0/0/1.101            198.51.100.12   YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
Serial0/1/0            unassigned      NO  unset  up                    down
Serial0/1/1            unassigned      NO  unset  up                    down
GigabitEthernet0       unassigned      YES DHCP   administratively down down
Loopback0              3.3.3.3         YES manual up                    up
Tunnel1                10.1.12.2       YES manual up                    up
```

```
Oshawa#show ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  administratively down down
GigabitEthernet0/0/1   203.0.113.12    YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
GigabitEthernet0       unassigned      YES manual administratively down down
Loopback0              4.4.4.4         YES manual up                    up
Loopback100            172.16.4.1      YES manual up                    up
Loopback101            172.16.41.1     YES manual up                    up
Loopback102            172.16.42.1     YES manual up                    up
Loopback103            172.16.43.1     YES manual up                    up
Tunnel1                10.1.12.3       YES manual up                    up
```

```
TOR-SW1#show ip int br
Interface              IP-Address      OK? Method Status                Protoco
Vlan1                  unassigned      YES unset  up                    up
Vlan10                 172.16.1.2      YES manual up                    up
GigabitEthernet0/0     unassigned      YES DHCP   administratively down down
GigabitEthernet1/0/1   unassigned      YES unset  up                    up
GigabitEthernet1/0/2   unassigned      YES unset  administratively down down
GigabitEthernet1/0/3   unassigned      YES unset  administratively down down
GigabitEthernet1/0/4   unassigned      YES unset  administratively down down
GigabitEthernet1/0/5   unassigned      YES unset  administratively down down
GigabitEthernet1/0/6   unassigned      YES unset  administratively down down
GigabitEthernet1/0/7   unassigned      YES unset  administratively down down
GigabitEthernet1/0/8   unassigned      YES unset  administratively down down
GigabitEthernet1/0/9   unassigned      YES unset  administratively down down
GigabitEthernet1/0/10  unassigned      YES unset  administratively down down
GigabitEthernet1/0/11  unassigned      YES unset  up                    up
GigabitEthernet1/0/12  unassigned      YES unset  administratively down down
GigabitEthernet1/0/13  unassigned      YES unset  administratively down down
GigabitEthernet1/0/14  unassigned      YES unset  administratively down down
GigabitEthernet1/0/15  unassigned      YES unset  administratively down down
GigabitEthernet1/0/16  unassigned      YES unset  administratively down down
GigabitEthernet1/0/17  unassigned      YES unset  administratively down down
GigabitEthernet1/0/18  unassigned      YES unset  administratively down down
GigabitEthernet1/0/19  unassigned      YES unset  administratively down down
GigabitEthernet1/0/20  unassigned      YES unset  administratively down down
GigabitEthernet1/0/21  unassigned      YES unset  administratively down down
GigabitEthernet1/0/22  unassigned      YES unset  administratively down down
GigabitEthernet1/0/23  unassigned      YES unset  administratively down down
GigabitEthernet1/0/24  unassigned      YES unset  administratively down down
GigabitEthernet1/1/1   unassigned      YES unset  administratively down down
GigabitEthernet1/1/2   unassigned      YES unset  administratively down down
GigabitEthernet1/1/3   unassigned      YES unset  administratively down down
GigabitEthernet1/1/4   unassigned      YES unset  administratively down down
Ap1/0/1                unassigned      YES unset  up                    up
Loopback101            172.16.11.1     YES manual up                    up
Loopback102            172.16.12.1     YES manual up                    up
Loopback103            172.16.13.1     YES manual up                    up
Loopback112            172.16.112.1    YES manual up                    up
```

```
OTT-SW1#show ip int br
Interface              IP-Address      OK? Method Status                Protoc
Vlan1                  unassigned      YES unset  up                    up
Vlan10                 172.16.3.2      YES manual up                    up
Vlan100                192.0.2.254     YES manual up                    up
Vlan101                198.51.100.254  YES manual up                    up
Vlan102                203.0.113.254   YES manual up                    up
GigabitEthernet0/0     unassigned      YES DHCP   administratively down down
GigabitEthernet1/0/1   unassigned      YES unset  up                    up
GigabitEthernet1/0/2   unassigned      YES unset  administratively down down
GigabitEthernet1/0/3   unassigned      YES unset  administratively down down
GigabitEthernet1/0/4   unassigned      YES unset  administratively down down
GigabitEthernet1/0/5   unassigned      YES unset  administratively down down
GigabitEthernet1/0/6   unassigned      YES unset  administratively down down
GigabitEthernet1/0/7   unassigned      YES unset  administratively down down
GigabitEthernet1/0/8   unassigned      YES unset  administratively down down
GigabitEthernet1/0/9   unassigned      YES unset  administratively down down
GigabitEthernet1/0/10  unassigned      YES unset  administratively down down
GigabitEthernet1/0/11  unassigned      YES unset  up                    up
GigabitEthernet1/0/12  unassigned      YES unset  up                    up
GigabitEthernet1/0/13  unassigned      YES unset  administratively down down
GigabitEthernet1/0/14  unassigned      YES unset  administratively down down
GigabitEthernet1/0/15  unassigned      YES unset  administratively down down
GigabitEthernet1/0/16  unassigned      YES unset  administratively down down
GigabitEthernet1/0/17  unassigned      YES unset  administratively down down
GigabitEthernet1/0/18  unassigned      YES unset  administratively down down
GigabitEthernet1/0/19  unassigned      YES unset  administratively down down
GigabitEthernet1/0/20  unassigned      YES unset  administratively down down
GigabitEthernet1/0/21  unassigned      YES unset  administratively down down
GigabitEthernet1/0/22  unassigned      YES unset  administratively down down
GigabitEthernet1/0/23  unassigned      YES unset  administratively down down
GigabitEthernet1/0/24  unassigned      YES unset  administratively down down
GigabitEthernet1/1/1   unassigned      YES unset  administratively down down
GigabitEthernet1/1/2   unassigned      YES unset  administratively down down
GigabitEthernet1/1/3   unassigned      YES unset  administratively down down
GigabitEthernet1/1/4   unassigned      YES unset  administratively down down
Ap1/0/1                unassigned      YES unset  up                    up
Loopback101            172.16.31.1     YES manual up                    up
Loopback102            172.16.32.1     YES manual up                    up
Loopback103            172.16.33.1     YES manual up                    up
```

```
Toronto#show ip protocols | excl sec nhrp
*** IP Routing is NSF aware ***

Routing Protocol is "application"
  Sending updates every 0 seconds
  Invalid after 0 seconds, hold down 0, flushed after 0
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Maximum path: 32
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 4)

Routing Protocol is "ospfv3 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Autonomous system boundary router
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0/1.10
  Redistributing: eigrp 12
  Maximum path: 4
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.10.10.10           110     00:07:37
  Distance: (default is 110)

Routing Protocol is "bgp 64612"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: ospf 12 (internal)
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    10.202.10.1
  Maximum path: 1
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.202.10.1           20      00:07:24
  Distance: external 20 internal 200 local 200

Routing Protocol is "nhrp"
  Maximum path: 32
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 250)

Routing Protocol is "eigrp 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: ospfv3 12
  EIGRP-IPv4 VR(OntarioTech12) Address-Family Protocol for AS(12)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
    Metric rib-scale 128
    Metric version 64bit
    Soft SIA disabled
    NSF-aware route hold timer is 240
  EIGRP NSF disabled
     NSF signal timer is 20s
     NSF converge timer is 120s
    Router-ID: 1.1.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1
      Total Prefix Count: 15
      Total Redist Count: 5

  Automatic Summarization: disabled
  Address Summarization:
    172.16.0.0/16 for Tu1
      Summarizing 13 components with metric 7206960
  Maximum path: 4
  Routing for Networks:
    10.1.12.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.12.3               90      00:07:28
    10.1.12.2               90      00:07:28
  Distance: internal 90 external 170
```

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

Routing Protocol is "application"
  Sending updates every 0 seconds
  Invalid after 0 seconds, hold down 0, flushed after 0
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Maximum path: 32
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 4)

Routing Protocol is "ospfv3 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 3.3.3.3
  Autonomous system boundary router
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0/1.10
  Redistributing: bgp 64612, eigrp 12
  Maximum path: 4
  Routing Information Sources:
    Gateway         Distance      Last Update
    30.30.30.30           110     00:08:13
  Distance: (default is 110)

Routing Protocol is "bgp 64612"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: ospfv3 12
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    10.202.20.1
  Maximum path: 1
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.202.20.1           20      00:08:14
  Distance: external 20 internal 200 local 200

Routing Protocol is "nhrp"
  Maximum path: 32
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 250)

Routing Protocol is "eigrp 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: ospfv3 12
  EIGRP-IPv4 VR(OntarioTech12) Address-Family Protocol for AS(12)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
    Metric rib-scale 128
    Metric version 64bit
    Soft SIA disabled
    NSF-aware route hold timer is 240
  EIGRP NSF disabled
     NSF signal timer is 20s
     NSF converge timer is 120s
    Router-ID: 3.3.3.3
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1
      Total Prefix Count: 6
      Total Redist Count: 4

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.1.12.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.12.1               90      00:08:28
  Distance: internal 90 external 170
```

VII. References

“The Cisco Bible”, Edgeworth, B., Rios, R. G., Gooley, J., & Hucaby, D. (2020). CCNP and CCIE enterprise core ENCOR 350-401. San Jose, CA: Cisco Press.

Cisco and/or its affiliates, “Cisco Networking Labs,” Cisco Networking Academy, 2020. .

referencing the entirety of the labs offered through the advanced networking 2 course.

J. Lowe, “Lecture 16 - Network Automation,” in Advanced Networking II, 16-Apr-2021.

Case study document itself provided by Josh, i.e the images of the topology