Lab 3:

Configuring EoMPLS
Basic traffic engineering

Objective: EoMPLS without explicit pseudowire

1. First we need to create VRF-lite on our CEs use the VRF name **vpn-eomplslab** (RD 10:6000) for this lab.
2. We are going to configure VLAN based EoMPLS as follows:

A end Z end
CE CE subint CE subint ip PE PE subint VC PE subint PE CE subint ip CE subint CE

Circuit A End						Circuit Z End				
CE	**CE subint**	**CE Subint IP**	**PE**	**PE Subint**	**VC**	**PE Subint**	**PE**	**CE subint IP**	**CE subint**	**CE**
CE12	Fa0/0.700	192.168.70.1/24	PE1	Fa1/1.700	17004700	Fa1/0.700	PE4	192.168.70.2/24	Fa0/0.700	CE4
CE34	Fa0/0.800	192.168.80.1/24	PE3	Fa1/1.800	38002800	Fa1/0.800	PE2	192.168.80.2/24	Fa0/0.800	CE2
CE34	Fa0/0.810	192.168.81.1/24	PE3	Fa1/1.810	38103810	Fa1/0.810	PE3	192.168.81.2/24	Fa0/0.810	CE3
CE34	Fa1/0.820	192.168.82.1/24	PE4	Fa1/1.820	48203820	Fa1/0.820	PE3	192.168.82.2/24	Fa0/0.820	CE3

In this case we're using a standard VC numbering scheme of [A-PE][VLAN#][B-PE][VLAN#]
Where PE numbers are 1 digit and VLAN# is three digits.
E.g. 17004700 is a VC from PE 1 on subint 700 to PE 4 on subint 700.
This VC numbering scheme only allows for nine PEs, so it is not very scalable!

3. One of the circuits in the above table is invalid. Which one, and why? This highlights a big restriction in providing EoMPLS circuits to customers.
4. Create the CE subinterfaces and add them to the CE VRF vpn-eomplslab.
   Note that no routing is going to be setup for this lab, we will simply be verifying connectivity across the interface addresses between CEs.
5. On each PE create the subinterface and EoMPLS xconnect

   E.g for PE1
   conf t
   int fa1/1.700
   encap dot1q 700
   xconnect 172.16.0.4 17004700 encap mpls

5. Once you have configured the PE EoMPLS subinterfaces, check that the EoMPLS circuits have come up, with 'show mpls l2transport vc' and 'show mpls l2transport vc X detail'
6. Check that you can ping between the CE interfaces at each end of the EoMPLS circuit
7. If you don't have connectivity, debug the circuit from CE-PE-PE-CE using the information in the slides
8. For an EoMPLS circuit does the attachment subinterface VLAN number need to be the same on each PE? Does this have implications for CDP if it is running between the CE?

9. Must the CE be a router? If is a switch, what things need to be considered?
10. If the CE is a switch, and the PE subinterface VLAN is different at each end of the circuit, can you think of any issues? With standard STP? With PVSTP?

Objective: Pseudowire based EoMPLS

1. We are going to build on our configuration from the previous lab here.
2. Currently our EoMPLS xconnect statements contain all information required to build an EoMPLS VC to the far end PE. We can use a pseudo-wire class to configure advanced parameters about how the EoMPLS session will be built
3. We are going to name our pseudowires using the following convention

PE[local PE number]-PE[remote PE number]
e.g. for a pseudowire between PE1 and PE4: PE1-PE4

4. For each PE create the pseudowire, at this point we are only specifying that the pseudowire uses MPLS encapsulation:

   e.g. for PE1 to PE4 pseudowire:

   Rack2-PE1(config)#pseudowire-class PE1-PE4
   Rack2-PE1(config-pw-class)#encapsulation mpls

5. We need to tell all our xconnects to use the appropriate pseudowire class, so for VC17004700 on PE1:

   Rack2-PE1(config)#int fa1/1.700
   Rack2-PE1(config-subif)#xconnect 172.16.0.4 17004700 pw-class PE1-PE4

6. Check that all your VCs are still up, and you can forward traffic over them.
7. Did the use of pseudowires achieve anything useful in this example?
8. If we were to point our traffic down a traffic engineering tunnel, where do you think we would include the configuration to do this?

Objective: Basic Traffic Engineering

Create a basic MPLS-TE tunnel

1. Examine your routing table and check the path between your PE's loopbacks. Use both the routing table and traceroutes between PEs.
2. What has selected this path? How is this path selected?
3. Amongst other thing, MPLS Traffic Engineering (MPLS-TE) allows us to utilise paths through the network that wouldn't ordinarily choose.
4. For this lab we want to move all our EoMPLS traffic away from the P5-P6 and P7-PE3. Which of our EoMPLS VCs currently traverse these links?
5. We are going to create the following MPLS-TE tunnels:

on PE1 - Tunnel100, desc PE1 to PE4, via P5,P7,P6
on PE4 - Tunnel101, desc PE4 to PE1, via P6,P7,P5
on PE2 - Tunnel200, desc PE2 to PE3, via P7,P6
on PE3 - Tunnel201, desc PE3 to PE2, via P6,P7

6. First we will create the basic tunnel, and allow it to use a dynamic path – e.g. follow the IGP best path to get to the destination PE. E.g. on PE1:

```
interface Tunnel100
 description PE1 to PE4, via P5,P7,P6
 ip unnumbered Loopback0
 tunnel destination 172.16.0.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng path-option 10 dynamic
```

7. Now look the route to the destination of your tunnel. Has it changed?
8. Add 'tunnel mpls traffic-eng autoroute announce' to your tunnel, and check again. What has this command inserted in the route table of your PE?
9. What is the best path to your tunnel's destination PE now? How many hops away is it?
10. Have a look at the tunnel itself and find out what path the tunnel itself is taking, using 'show mpls traffic-eng tunnels TunnelX'

Using explicit paths for MPLS-TE tunnels

11. We are going to create an explicit path through our MPLS network to avoid the links P5-P6 and P7-PE3. We will be creating the following explicit paths:

on PE1 – explicit-path PE1-to-PE4-via-P7
on PE4 – explicit-path PE4-to-PE1-via-P7
on PE2 – explicit-path PE2-to-PE3-via-P6
on PE3 – explicit-path PE3-to-PE2-via-P6

12. An explicit path contains a list of IP addresses corresponding to the interface of the next P/PE you wish to reach. E.g. for PE1:

```
Rack2-PE1(config)#ip explicit-path name PE1-to-PE4-via-P7
Rack2-PE1(cfg-ip-expl-path)#next-address [IP address of P5 fa1/0]
Rack2-PE1(cfg-ip-expl-path)#next-address [IP address of P7 fa2/0]
Rack2-PE1(cfg-ip-expl-path)#next-address [IP address of P6 fa0/0]
Rack2-PE1(cfg-ip-expl-path)#next-address [IP address of PE4 fa2/0]
```

13. Configure the explicit paths on each PE. You can view your configured explicit-paths with 'show ip explicit-paths'
14. We need to tell our MPLS-TE tunnel to use the explicit-path we just created. E.g. for PE1:

```
interface Tunnel100
 tunnel mpls traffic-eng path-option 1 explicit name PE1-to-PE4-via-P7
```

15. View the details of your Tunnel now, taking note of the path it is taking. Is this different to what you saw before?

16. What is the significance of having the two path options configured within the tunnel? What would happen if there was a failure of a link somewhere along the explicit path?

Configure pseudowire to use an MPLS-TE tunnel

17. We can configure a pseudo wire to use a MPLS-TE tunnel with the following:

```
conf t
pseudowire-class [name of psuedowire class]
 ! encapsulation mpls, if it is not already configured
 preferred-path interface TunnelXXX
```

Configure EoMPLS VC to use a pseudowire

18. Before we configure our EoMPLS pseudowire to use our MPLS-TE tunnel we are going to have a look at what happens when we have a link failure in our MPLS network.
19. Set a continuous ping running from CE to CE across one of your VCs, leaving the default timeout of 1s.

    For CE12-CE4's VC we are going to break the link between P5 and P6
    For CE34-CE2's VC we are going to break the link between P7 and PE2
    For CE34-CE3's VC we are going to break the link between P6 and PE3

    This can be achieved by shutting down the interface at one end.

    NOTE – you will need to check the status of the MPLS network before you start this, as other groups will be shutting down links also. Make sure only one link is broken at a time.
20. How many pings were lost when you shut down a link? What was happing during this time?
21. Enable all links again once you are finished this part.
22. We will now tell configure our EoMPLS VCs to use the pseudowire classes (and their corresponding MPLS-TE tunnels) we have just defined. This is achieved by reapplying the xconnect statement and referencing a given pseudowire. E.g. for PE1:

```
Rack2-PE1(config)#int fa1/1.700
Rack2-PE1(config-subif)#xconnect 172.16.0.4 17004700 pw-class PE1-PE4
```

23. Check that your VCs are still up. Does your VC stand up when one end is configured to use pw-class and the other isn't? Why or why not?
24. Repeat the breaking of a link as per #19 above now that your VCs are using our MPLS-TE tunnel. How many pings are lost this time?
25. What is different when we are using the MPLS-TE tunnel? Would you expect to see this same bevaiour if our MPLS-TE tunnel only had one path-option configured? Hint: look at the labelled switched paths that are created by the MPLS-TE tunnels.