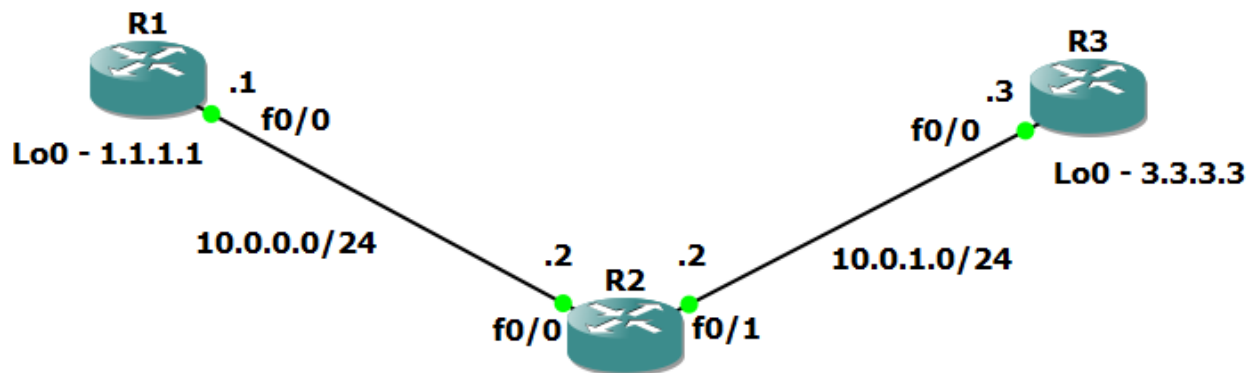


Cisco MPLS Configuration

Step 1 – IP addressing of MPLS Core and OSPF

First bring 3 routers into your topology R1, R2, R3 position them as below. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R3



```
R1
hostname R1
int lo0
ip add 1.1.1.1 255.255.255.255
ip ospf 1 area 0

int f0/0
ip add 10.0.0.1 255.255.255.0
no shut
ip ospf 1 area 0

R2
hostname R2
int lo0
```

```
ip add 2.2.2.2 255.255.255.255
```

```
ip ospf 1 are 0
```

```
int f0/0
```

```
ip add 10.0.0.2 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

```
int f0/1
```

```
ip add 10.0.1.2 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

```
R3
```

```
hostname R3
```

```
int lo0
```

```
ip add 3.3.3.3 255.255.255.255
```

```
ip ospf 1 are 0
```

```
int f0/0
```

```
ip add 10.0.1.3 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

You should now have full ip connectivity between R1, R2, R3 to verify this we need to see if we can ping between the loopbacks of R1 and R3

```
R1#ping 3.3.3.3 source lo0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2
seconds:

Packet sent with a source address of 1.1.1.1

!!!!

Success rate is 100 percent (5/5), round-trip
min/avg/max = 40/52/64 ms

R1#
```

You could show the routing table here, but the fact that you can ping between the loopbacks is verification enough and it is safe to move on.

Step 2 – Configure LDP on all the interfaces in the MPLS Core

In order to run MPLS you need to enable it, there are two ways to do this.

- At each interface enter the **mpls ip** command
- Under the ospf process use the **mpls ldp autoconfig** command

For this tutorial we will be using the second option, so go into the ospf process and enter **mpls ldp autoconfig** – this will enable mpls label distribution protocol on every interface running ospf under that specific process.

```
R1

router ospf 1

mpls ldp autoconfig

R2

router ospf 1
```

```
mpls ldp autoconfig
```

```
R3
```

```
router ospf 1
```

```
mpls ldp autoconfig
```

You should see log messages coming up showing the LDP neighbors are up.

```
R2#
```

```
*Mar  1 00:31:53.643: %SYS-5-CONFIG_I: Configured from console
```

```
*Mar  1 00:31:54.423: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
```

```
R2#
```

```
*Mar  1 00:36:09.951: %LDP-5-NBRCHG: LDP Neighbor 3.3.3.3:0 (2) is UP
```

To verify the mpls interfaces the command is very simple – **sh mpls interface**

This is done on R2 and you can see that both interfaces are running mpls and using LDP

```
R2#sh mpls interface
```

Interface	IP	Tunnel
Operational		
FastEthernet0/0	Yes (ldp)	No Yes
FastEthernet0/1	Yes (ldp)	No Yes

You can also verify the LDP neighbors with the **sh mpls ldp neighbors** command.

```
R2#sh mpls ldp neigh
```

```
Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
```

```
TCP connection: 1.1.1.1.646 - 2.2.2.2.37909
```

```
State: Oper; Msgs sent/rcvd: 16/17; Downstream
Up time: 00:07:46

LDP discovery sources:

FastEthernet0/0, Src IP addr: 10.0.0.1

Addresses bound to peer LDP Ident:

10.0.0.1          1.1.1.1

Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0

TCP connection: 3.3.3.3.22155 - 2.2.2.2.646

State: Oper; Msgs sent/rcvd: 12/11; Downstream
Up time: 00:03:30

LDP discovery sources:

FastEthernet0/1, Src IP addr: 10.0.1.3

Addresses bound to peer LDP Ident:

10.0.1.3          3.3.3.3
```

One more verification to confirm LDP is running ok is to do a trace between R1 and R3 and verify if you get MPLS Labels show up in the trace.

```
R1#trace 3.3.3.3

Type escape sequence to abort.
Tracing the route to 3.3.3.3

  1 10.0.0.2 [MPLS: Label 17 Exp 0] 84 msec 72 msec 44
msec
  2 10.0.1.3 68 msec 60 msec *
```

As you can see the trace to R2 used an MPLS Label in the path, as this is a very small MPLS core only one label was used as R3 was the final hop.

So to review we have now configured IP addresses on the MPLS core, enabled OSPF and full IP connectivity between all routers and finally enabled mpls on all the interfaces in the core and have established ldp neighbors between all routers.

The next step is to configure MP-BGP between R1 and R3

This is when you start to see the layer 3 vpn configuration come to life

Step 3 – MPLS BGP Configuration between R1 and R3

We need to establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

```
R1#
router bgp 1
  neighbor 3.3.3.3 remote-as 1
  neighbor 3.3.3.3 update-source Loopback0
  no auto-summary
  !
  address-family vpnv4
    neighbor 3.3.3.3 activate
```

```
R3#
router bgp 1
  neighbor 1.1.1.1 remote-as 1
  neighbor 1.1.1.1 update-source Loopback0
  no auto-summary
  !
  address-family vpnv4
    neighbor 1.1.1.1 activate
```

```
*Mar  1 00:45:01.047: %BGP-5-ADJCHANGE: neighbor 1.1.1.1
Up
```

You should see log messages showing the BGP sessions coming up.

To verify the BGP session between R1 and R3 issue the command **sh bgp vpnv4 unicast all summary**

```
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor          V    AS MsgRcvd MsgSent   TblVer  InQ
OutQ Up/Down  State/PfxRcd
3.3.3.3           4      1     218     218       1    0
0 03:17:48           0
```

You can see here that we do have a bgp vpnv4 peering to R3 – looking at the PfxRcd you can see it says 0 this is because we have not got any routes in BGP. We are now going to add two more routers to the topology. These will be the customer sites connected to R1 and R3. We will then create a VRF on each router and put the interfaces connected to each site router into that VRF.

Step 4 – Add two more routers, create VRFs

We will add two more routers into the topology so it now looks like the final topology

Router 4 will peer OSPF using process number 2 to a VRF configured on R1. It will use the local site addressing of 192.168.1.0/24.

```
R4
int lo0
ip add 4.4.4.4 255.255.255.255
ip ospf 2 area 2
int f0/0
```

```
ip add 192.168.1.4 255.255.255.0
ip ospf 2 area 2
no shut

R1
int f0/1
no shut
ip add 192.168.1.1 255.255.255.0
```

Now at this point we have R4 peering to R1 but in the global routing table of R1 which is not what we want.

We are now going to start using VRF's

What is a VRF in networking?

Virtual routing and forwarding (**VRF**) is a technology included in IP (Internet Protocol) that allows multiple instances of a routing table to co-exist in a router and work together but not interfere with each other.. This increases functionality by allowing network paths to be segmented without using multiple devices.

As an example if R1 was a PE Provider Edge router of an ISP and it had two customers that were both addressed locally with the 192.168.1.0/24 address space it could accommodate both their routing tables in different VRFs – it distinguishes between the two of them using a Route Distinguisher

So back to the topology – we now need to create a VRF on R1

For this mpls tutorial I will be using VRF RED

```
R1
ip vrf RED
rd 4:4
route-target both 4:4
```


The RD and route-target do not need to be the same – and for a full explanation please read this post on Route Distinguishers [Route Distinguisher vs Route Target](#) before proceeding.

So now we have configured the VRF on R1 we need to move the interface F0/1 into that VRF

```
R1
int f0/1
ip vrf forwarding RED
```

Now notice what happens when you do that – the IP address is removed

```
R1(config-if)#ip vrf fo
R1(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.1.1
removed due to enabling VRF RED
```

You just need to re-apply it

```
R1
int f0/1
ip address 192.168.1.1 255.255.255.0
```

Now if we view the config on R1 int f0/1 you can see the VRF configured.

```
R1
R1#sh run int f0/1
Building configuration...
Current configuration : 119 bytes
!
```

```
interface FastEthernet0/1

  ip vrf forwarding RED

  ip address 192.168.1.1 255.255.255.0

  duplex auto

  speed auto

end

R1#
```

Now we can start to look into VRF's and how they operate – you need to understand now that there are 2 routing tables within R1

- The Global Routing Table
- The Routing Table for VRF RED

If you issue the command **sh ip route** this shows the routes in the global table and you will notice that you do not see 192.168.1.0/24

```
R1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B
- BGP

  D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
inter area

  N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2

  E1 - OSPF external type 1, E2 - OSPF external type 2

  i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 -
IS-IS level-2

  ia - IS-IS inter area, * - candidate default, U - per-
user static route

  o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
1.0.0.0/32 is subnetted, 1 subnets
C 1.1.1.1 is directly connected, Loopback0
  2.0.0.0/32 is subnetted, 1 subnets
O 2.2.2.2 [110/11] via 10.0.0.2, 01:03:48,
FastEthernet0/0
  3.0.0.0/32 is subnetted, 1 subnets
O 3.3.3.3 [110/21] via 10.0.0.2, 01:02:29,
FastEthernet0/0
  10.0.0.0/24 is subnetted, 2 subnets
C 10.0.0.0 is directly connected, FastEthernet0/0
O 10.0.1.0 [110/20] via 10.0.0.2, 01:02:39,
FastEthernet0/0
R1#
```

If you now issue the command `sh ip route vrf red` – this will show the routes in the routing table for VRF RED

```
R1#sh ip route vrf red
% IP routing table red does not exist
```

NOTE: The VRF name is case sensitive!

```
R1#sh ip route vrf RED

Routing Table: RED

Codes: C - connected, S - static, R - RIP, M - mobile, B
- BGP

  D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
inter area

  N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
```

```
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 -  
IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-  
user static route  
o - ODR, P - periodic downloaded static route
```

Gateway of last resort is not set

```
C 192.168.1.0/24 is directly connected, FastEthernet0/1  
R1#
```

We just need to enable OSPF on this interface and get the loopback address for R4 in the VRF RED routing table before proceeding.

```
R1  
  
int f0/1  
  
ip ospf 2 area 2
```

You should see a log message showing the OSPF neighbor come up

```
R1(config-if)#  
  
*Mar 1 01:12:54.323: %OSPF-5-ADJCHG: Process 2, Nbr  
4.4.4.4  
on FastEthernet0/1 from LOADING to FULL, Loading Done
```

If we now check the routes in the VRF RED routing table you should see 4.4.4.4 in there as well.

```
R1#sh ip route vrf RED
```

```
Routing Table: RED
```

```
Codes: C - connected, S - static, R - RIP, M - mobile,  
B - BGP
```

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

4.0.0.0/32 is subnetted, 1 subnets

O 4.4.4.4 [110/11] via 192.168.1.4, 00:00:22, FastEthernet0/1

C 192.168.1.0/24 is directly connected, FastEthernet0/1

R1#

We now need to repeat this process for R3 & R6

Router 6 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing of 192.168.2.0/24.

```
R6
int lo0
ip add 6.6.6.6 255.255.255.255
ip ospf 2 area 2
int f0/0
ip add 192.168.2.6 255.255.255.0
ip ospf 2 area 2
```

```
no shut

R3

int f0/1

no shut

ip add 192.168.2.3 255.255.255.0
```

We also need to configure a VRF onto R3 as well.

```
R3

ip vrf RED

rd 4:4

route-target both 4:4
```

So now we have configured the VRF on R3 we need to move the interface F0/1 into that VRF

```
R3

int f0/1

ip vrf forwarding RED
```

Now notice what happens when you do that – the IP address is removed

```
R3(config-if)#ip vrf forwarding RED

% Interface FastEthernet0/1 IP address 192.168.2.1
removed due to enabling VRF RED
```

You just need to re-apply it

```
R3

int f0/1

ip address 192.168.2.1 255.255.255.0
```

Now if we view the config on R3 int f0/1 you can see the VRF configured.

```
R3
```

```
R3#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
ip vrf forwarding RED
ip address 192.168.2.1 255.255.255.0
duplex auto
speed auto
end
```

Finally we just need to enable OSPF on that interface and verify the routes are in the RED routing table.

```
R3
int f0/1
ip ospf 2 area 2
```

Check the routes in vrf RED

```
R3
R3#sh ip route vrf RED

Routing Table: RED

Codes: C - connected, S - static, R - RIP, M - mobile, B
- BGP

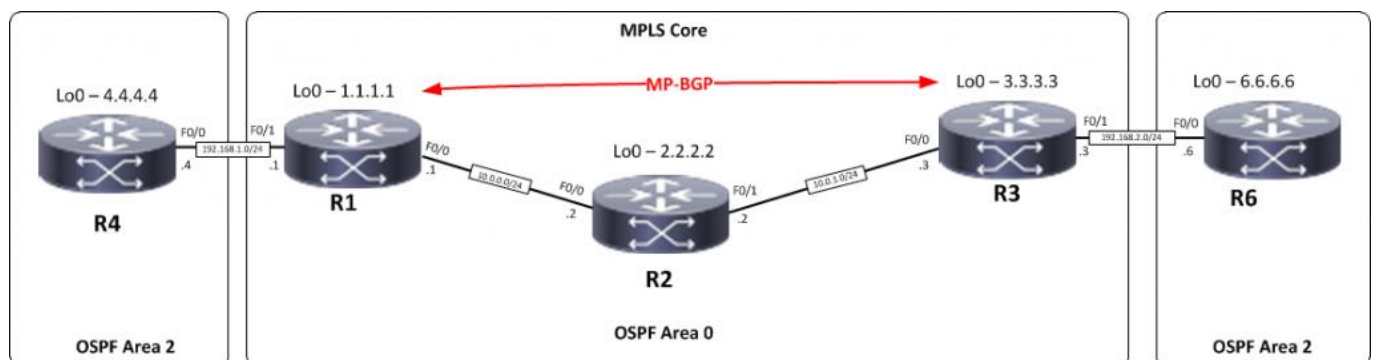
Gateway of last resort is not set
```

```

6.0.0.0/32 is subnetted, 1 subnets
O      6.6.6.6 [110/11] via 192.168.2.6, 00:02:44,
FastEthernet0/1
C      192.168.2.0/24 is directly connected,
FastEthernet0/1
R3#

```

Ok so we have come a long way now let's review the current situation. We now have this setup



R1,R2,R3 form the MPLS Core and are running OSPF with all loopbacks running a /32 address and all have full connectivity. R1 and R3 are peering with MP-BGP. LDP is enabled on all the internal interfaces. The external interfaces of the MPLS core have been placed into a VRF called RED and then a site router has been joined to that VRF on each side of the MPLS core – (These represent a small office)

The final step to get full connectivity across the MPLS core is to redistribute the routes in OSPF on R1 and R3 into MP-BGP and MP-BGP into OSPF, this is what we are going to do now.

We need to redistribute the OSPF routes from R4 into BGP in the VRF on R1, the OSPF routes from R6 into MP-BGP in the VRF on R3 and then the routes in MP-BGP in R1 and R3 back out to OSPF

Before we start lets do some verifications

Check the routes on R4

```

R4#sh ip route

4.0.0.0/32 is subnetted, 1 subnets
C 4.4.4.4 is directly connected, Loopback0

```



```
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

As expected we have the local interface and the loopback address.

When we are done we want to see 6.6.6.6 in there so we can ping across the MPLS

Check the routes on R1

```
R1#sh ip route
```

```
1.0.0.0/32 is subnetted, 1 subnets
```

```
C 1.1.1.1 is directly connected, Loopback0
```

```
2.0.0.0/32 is subnetted, 1 subnets
```

```
O 2.2.2.2 [110/11] via 10.0.0.2, 00:01:04,  
FastEthernet0/0
```

```
3.0.0.0/32 is subnetted, 1 subnets
```

```
O 3.3.3.3 [110/21] via 10.0.0.2, 00:00:54,  
FastEthernet0/0
```

```
10.0.0.0/24 is subnetted, 2 subnets
```

```
C 10.0.0.0 is directly connected, FastEthernet0/0
```

```
O 10.0.1.0 [110/20] via 10.0.0.2, 00:00:54,  
FastEthernet0/0
```

Remember we have a VRF configured on this router so this command will show routes in the global routing table (the MPLS Core) and it will not show the 192.168.1.0/24 route as that is in VRF RED – to see that we run the following command

```
R1#sh ip route vrf RED
```

```
Routing Table: RED
```

```
4.0.0.0/32 is subnetted, 1 subnets
```

```
O 4.4.4.4 [110/11] via 192.168.1.4, 00:02:32,  
FastEthernet0/1  
  
C 192.168.1.0/24 is directly connected, FastEthernet0/1
```

Here you can see Routing Table: RED is shown and the routes to R4 are now visible with 4.4.4.4 being in OSPF.

So we need to do the following;

- Redistribute OSPF into MP-BGP on R1
- Redistribute MP-BGP into OSPF on R1
- Redistribute OSPF into MP-BGP on R3
- Redistribute MP-BGP into OSPF on R3

Redistribute OSPF into MP-BGP on R1

```
R1  
  
router bgp 1  
  
address-family ipv4 vrf RED  
  
redistribute ospf 2
```

Redistribute OSPF into MP-BGP on R3

```
R3  
  
router bgp 1  
  
address-family ipv4 vrf RED  
  
redistribute ospf 2
```

This has enabled redistribution of the OSPF routes into BGP. We can check the routes from R4 and R6 are now showing in the BGP table for their VRF with this command

sh ip bgp vpnv4 vrf RED

```
R1#sh ip bgp vpnv4 vrf RED  
  
BGP table version is 9, local router ID is 1.1.1.1  
  
Status codes: s suppressed, d damped, h history, *  
valid, > best,
```

```

r RIB-failure, S Stale

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

Network Next Hop Metric LocPrf Weight Path

Route Distinguisher: 4:4 (default for vrf RED)

*> 4.4.4.4/32 192.168.1.4 11 32768 ?

*>i6.6.6.6/32 3.3.3.3 11 100 0 ?

*> 192.168.1.0 0.0.0.0 0 32768 ?

*>i192.168.2.0 3.3.3.3 0 100 0 ?

```

Here we can see that 4.4.4.4 is now in the BGP table in VRF RED on R1 with a next hop of 192.168.1.4 (R4) and also 6.6.6.6 is in there as well with a next hop of 3.3.3.3 (which is the loopback of R3 – showing that it is going over the MPLS and R1 is not in the picture)

The same should be true on R3

```

R3#sh ip bgp vpnv4 vrf RED

BGP table version is 9, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, *
valid, > best, i - internal,

r RIB-failure, S Stale

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

Network Next Hop Metric LocPrf Weight Path

Route Distinguisher: 4:4 (default for vrf RED)

*>i4.4.4.4/32 1.1.1.1 11 100 0 ?

*> 6.6.6.6/32 192.168.2.6 11 32768 ?

*>i192.168.1.0 1.1.1.1 0 100 0 ?

```

```
*> 192.168.2.0 0.0.0.0 0 32768 ?
```

Which it is! 6.6.6.6 is now in the BGP table in VRF RED on R3 with a next hop of 192.168.2.6 (R6) and also 4.4.4 is in there as well with a next hop of 1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLS and R2 is not in the picture)

The final step is to get the routes that have come across the MPLS back into OSPF and then we can get end to end connectivity

R1

```
router ospf 2
redistribute bgp 1 subnets
```

R3

```
router ospf 2
redistribute bgp 1 subnets
```

If all has worked we should be now able to ping 6.6.6.6 from R4

Before we do let's see what the routing table looks like on R4

```
R4#sh ip route
```

```
4.0.0.0/32 is subnetted, 1 subnets
```

```
C 4.4.4.4 is directly connected, Loopback0
```

```
6.0.0.0/32 is subnetted, 1 subnets
```

```
O IA 6.6.6.6 [110/21] via 192.168.1.1, 00:01:31,
FastEthernet0/0
```

```
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

```
O E2 192.168.2.0/24 [110/1] via 192.168.1.1, 00:01:31,
FastEthernet0/0
```

Great we have 6.6.6.6 in there

Also check the routing table on R6

```
R6#sh ip route
```

```
4.0.0.0/32 is subnetted, 1 subnets
```

```
O IA 4.4.4.4 [110/21] via 192.168.2.1, 00:01:22,
FastEthernet0/0
```

```
6.0.0.0/32 is subnetted, 1 subnets
```

```
C 6.6.6.6 is directly connected, Loopback0
```

```
O IA 192.168.1.0/24 [110/11] via
192.168.2.1, 00:01:22, FastEthernet0/0
```

```
C 192.168.2.0/24 is directly connected, FastEthernet0/0
```

Brilliant we have 4.4.4.4 in there so we should be able to ping across the MPLS

```
R4#ping 6.6.6.6
```

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

```
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2
seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip
min/avg/max= 40/48/52ms
```

Which we can – to prove this is going over the MPLS and be label switched and not routed, lets do a trace

```
R4#trace 6.6.6.6
```

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

```
Tracing the route to 6.6.6.6
```

```
1 192.168.1.1 20 msec 8 msec 8 msec
```

```
2 10.0.0.2 [MPLS: Labels 17/20 Exp 0] 36 msec 40 msec  
36 msec
```

```
3 192.168.2.1 [MPLS: Label 20 Exp 0] 16 msec 40 msec 16  
msec
```

```
4 192.168.2.6 44 msec 40 msec 56 msec
```

```
R4#
```

What is MPLS network and how does it work?

Multiprotocol Label Switching (**MPLS**) is a way of routing traffic within a telecommunications **network** that directs data from one node to the next based path labels rather than long **network** addresses, It also allows the sharing of address space for clients as it is labels that are being routed not prefixes.

Is MPLS a routing protocol?

No, MPLS is a method to route networks across a service provider network, routing protocols like OSPF and BGP are used to make MPLS work. MPLS operates using BGP and typically uses OSPF to exchange routes with the customer.

Why is MPLS used?

MPLS was designed to work in a multiple protocol environment.

Today, **MPLS** is **used** to support metro-Ethernet services & mobile communications back-haul it's main benefit is the ability to have two clients using the same address space and routing over the service provider network as they are routing using labels and not prefixes.

For reference

https://www.youtube.com/watch?time_continue=2&v=V9ij7se6VDw&feature=emb_title