

Title: Configuring a Basic MPLS VPN - Cisco

Content courtesy of: <https://www.cisco.com/c/en/us/support/docs/multi-protocol-label-switching-mpls/mpls/13733-mpls-vpn-basic.html>

This document provides a sample configuration of a Multiprotocol Label Switching (MPLS) VPN when Border Gateway Protocol (BGP) or Routing Information Protocol (RIP) is present on the customer's site.

When used with MPLS, the VPN feature allows several sites to interconnect transparently through a service provider's network. One service provider network can support several different IP VPNs. Each of these appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN.

Each VPN is associated with one or more VPN routing or forwarding instances (VRFs). A VRF consists of an IP routing table, a derived Cisco express forwarding (CEF) table, and a set of interfaces that use this forwarding table.

The router maintains a separate routing and CEF table for each VRF. This prevents information being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems.

The router using Multiprotocol BGP (MP-BGP) distributes the VPN routing information using the MP-BGP extended communities.

For more information about the propagation of updates through a VPN, refer to these documents:

There are no specific requirements for this document.

The information in this document is based on these software and hardware versions:

P and PE Routers

- Cisco IOS® Software Release 12.2(6h) includes the MPLS VPN feature.
- Any Cisco router from the 7200 series or higher supports P functionality. The Cisco 2691, as well as any 3640 series or higher router supports PE functionality.

C and CE Routers

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

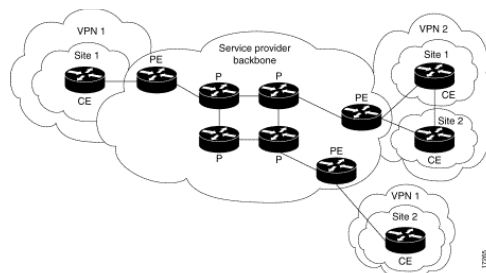
To implement the MPLS feature, you must have a router from the range of Cisco 2600 or higher. To select the required Cisco IOS with MPLS feature, use the Software Advisor (<http://tools.cisco.com/Support/Fusion/FusionHome.do>) (registered (<http://tools.cisco.com/RPF/register/register.do>) customers only). Also check for the additional RAM and Flash memory required to run the MPLS feature in the routers. WIC-1T, WIC-2T, and serial interfaces can be used.

Refer to Cisco Technical Tips Conventions (http://www.cisco.com/en/US/tech/tk801/tk36/technologies_tech_note09186a0080121ac5.shtml) for more information on document conventions.

The letters below represent the different types of routers and switches used.

- **P**—Provider's core router.
- **PE**—Provider's edge router.
- **CE**—Customer's edge router.
- **C**—Customer's router.

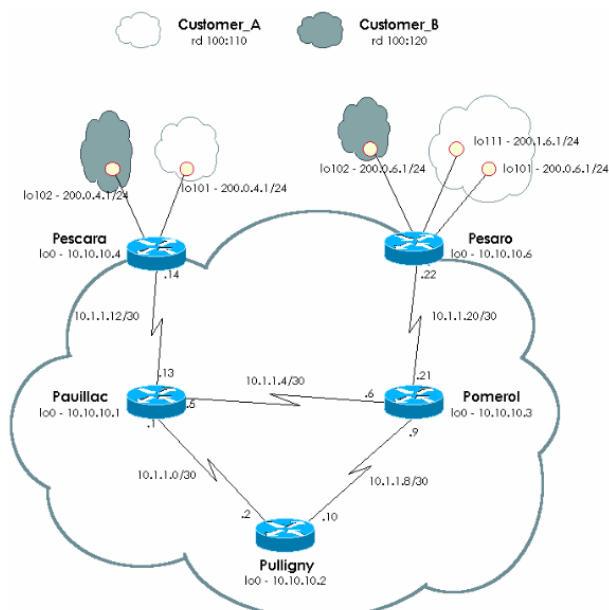
This diagram shows a typical configuration illustrating the conventions outlined above.



In this section, you are presented with the information to configure the features described in this document.

Note: Use the Command Lookup Tool (<http://tools.cisco.com/Support/CLILookup/CLISearchAction.do>) (registered (<http://tools.cisco.com/RPF/register/register.do>) customers only) to find more information on the commands used in this document.

This document uses this network setup:



Refer to MPLS Virtual Private Networks (http://www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/VPN.html) for more information.

Use this procedure in order to enable **ip cef** (http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_e1.html#wp1071616). For improved performance, use **ip cef distributed** (http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_e1.html#wp1071616) (where available). Complete these steps on the PEs after MPLS has been set up (configuring **tag-switching ip** (http://www.cisco.com/en/US/docs/ios/12_2/switch/command/reference/xrscmd9.html#wp1050292) on the interfaces).

1. Create one VRF for each VPN connected using the **ip vrf** (http://www.cisco.com/en/US/docs/ios/12_1/switch/command/reference/xrscmd2.html#wp1030105) **<VPN routing/forwarding instance name>** command.

When doing this:

- Specify the correct route distinguisher used for that VPN. This is used to extend the IP address so that you can identify which VPN it belongs to.

```
rd (http://www.cisco.com/en/US/docs/ios/12\_3/switch/command/reference/swi\_n1.htm
```

- Set up the import and export properties for the MP-BGP extended communities. These are used for filtering the import and export process.

```
route-target (http://www.cisco.com/en/US/docs/ios/12\_3/switch/command/reference/
```

2. Configure the forwarding details for the respective interfaces using the **ip vrf forwarding** (http://www.cisco.com/en/US/docs/ios/12_1/switch/command/reference/xrscmd2.html#wp1030051) **<VPN routing/forwarding instance name>** command and remember to set up the IP address after doing this.
3. Depending on the PE-CE routing protocol you are using, you can configure static routes or routing protocols (RIP, Open Shortest Path First [OSPF], or BGP) between PE and CE. Detailed configurations are available on the MPLS over ATM Support (http://www.cisco.com/en/US/tech/tk436/tk798/tech_configuration_examples_list.html) page.

Configure MP-BGP between the PE routers. There are several ways to configure BGP, such as using the route reflector or confederation methods. The method used here—direct neighbor configuration—is the simplest and the least scalable.

1. Declare the different neighbors.
2. Enter the **address-family ipv4 vrf** (http://www.cisco.com/en/US/docs/ios/12_2/iproute/command/reference/1rfmbgp.html#wp1024841) **<VPN routing/forwarding instance name>** command for each VPN present at this PE router.

Carry out one or more of the following steps, as necessary:

- Redistribute the static routing, RIP, or OSPF information.
- Redistribute connected routing information.
- Activate BGP neighboring with the CE routers.

3. Enter the **address-family vpnv4** (http://www.cisco.com/en/US/docs/ios/12_2/iproute/command/reference/1rfmbgp.html#wp1026442) mode, and complete the following steps:

This document uses these configurations:

Pescara

```

Current configuration:
!
version 12.2
!
hostname Pescara
!
ip cef
!

!--- Customer A commands.

ip vrf Customer_A

!--- Enables the VPN routing and forwarding (VRF) routing table. !--- This command can be used in global or !--- router confi

rd 100:110

!--- Route distinguisher creates routing and forwarding !--- tables for a VRF.

route-target export 100:1000

!--- Creates lists of import and export route-target extended !--- communities for the specified VRF.

route-target import 100:1000
!

!--- Customer B commands.

ip vrf Customer_B
rd 100:120
route-target export 100:2000
route-target import 100:2000
!
interface Loopback0
ip address 10.10.10.4 255.255.255.255
ip router isis

!--- Customer A commands.

interface Loopback101
ip vrf forwarding Customer_A

!--- Associates a VRF instance with an interface or subinterface.

ip address 200.0.4.1 255.255.255.0

!--- Loopback101 and 102 use the same IP address, 200.0.4.1. !--- This is allowed because they belong to two !--- different

no ip directed-broadcast
!

!--- Customer B commands.

interface Loopback102
ip vrf forwarding Customer_B
ip address 200.0.4.1 255.255.255.0

!--- Loopback101 and 102 use the same IP address, 200.0.4.1. !--- This is allowed because they belong to two !--- different

no ip directed-broadcast
!
interface Serial2/0
no ip address
no ip directed-broadcast
encapsulation frame-relay
no fair-queue
!
interface Serial2/0.1 point-to-point
description link to Pauillac
bandwidth 512
ip address 10.1.1.14 255.255.255.252
no ip directed-broadcast
ip router isis
tag-switching ip
frame-relay interface-dlci 401
!
router isis
net 49.0001.0000.0000.0004.00
is-type level-1
!
router bgp 100
bgp log-neighbor-changes

!--- Enables logging of BGP neighbor resets.

neighbor 10.10.10.6 remote-as 100

!--- Adds an entry to the BGP or multiprotocol BGP neighbor table.

neighbor 10.10.10.6 update-source Loopback0

```

```
!--- Enables BGP sessions to use a specific operational !--- interface for TCP connections.
!

!--- Customer A and B commands.

address-family vpnv4

!--- To enter address family configuration mode !--- for configuring routing sessions, such as BGP, !--- that use standard VPI

neighbor 10.10.10.6 activate
neighbor 10.10.10.6 send-community both

!--- Sends the community attribute to a BGP neighbor.

exit-address-family
!

!--- Customer B commands.

address-family ipv4 vrf Customer_B

!--- To enter address family configuration mode !--- for configuring routing sessions, such as BGP, !--- that use standard VPI

redistribute connected
no auto-summary
no synchronization
exit-address-family
!

!--- Customer A commands.

address-family ipv4 vrf Customer_A
redistribute connected
no auto-summary
no synchronization
exit-address-family
!
ip classless
!
end
```

Pesaro

```

Current configuration:
!
version 12.1
!
hostname Pesaro
!

!--- Customer A commands.

ip vrf Customer_A
rd 100:110
route-target export 100:1000
route-target import 100:1000
!

!--- Customer B commands.

ip vrf Customer_B
rd 100:120
route-target export 100:2000
route-target import 100:2000
!
ip cef

!
interface Loopback0
ip address 10.10.10.6 255.255.255.255
ip router isis

!--- Customer A commands.

interface Loopback101
ip vrf forwarding Customer_A
ip address 200.0.6.1 255.255.255.0
!

!--- Customer B commands.

interface Loopback102
ip vrf forwarding Customer_B
ip address 200.0.6.1 255.255.255.0
!

!--- Customer A commands.

interface Loopback111
ip vrf forwarding Customer_A
ip address 200.1.6.1 255.255.255.0
!
interface Serial0/0
no ip address
encapsulation frame-relay
no ip mroute-cache
random-detect
!
interface Serial0/0.1 point-to-point
description link to Pomerol
bandwidth 512
ip address 10.1.1.22 255.255.255.252
ip router isis
tag-switching ip
frame-relay interface-dlci 603
!
router isis
net 49.0001.0000.0000.0006.00
is-type level-1
!
router bgp 100
neighbor 10.10.10.4 remote-as 100
neighbor 10.10.10.4 update-source Loopback0
!

!--- Customer B commands.

address-family ipv4 vrf Customer_B
redistribute connected
no auto-summary
no synchronization
exit-address-family
!

!--- Customer A commands.

address-family ipv4 vrf Customer_A
redistribute connected
no auto-summary
no synchronization
exit-address-family

```

```

!

!--- Customer A and B commands.

address-family vpnv4
neighbor 10.10.10.4 activate
neighbor 10.10.10.4 send-community both
exit-address-family
!
ip classless
!
end

```

Pomerol

```

Current configuration:
!
version 12.0
!
hostname Pomerol
!
ip cef

!
interface Loopback0
ip address 10.10.10.3 255.255.255.255
ip router isis

!
interface Serial0/1
no ip address
no ip directed-broadcast
encapsulation frame-relay
random-detect
!
interface Serial0/1.1 point-to-point
description link to Pauillac
ip address 10.1.1.6 255.255.255.252
no ip directed-broadcast
ip router isis
tag-switching mtu 1520
tag-switching ip
frame-relay interface-dlci 301
!
interface Serial0/1.2 point-to-point
description link to Pulligny
ip address 10.1.1.9 255.255.255.252
no ip directed-broadcast
ip router isis
tag-switching ip
frame-relay interface-dlci 303
!
interface Serial0/1.3 point-to-point
description link to Pesaro
ip address 10.1.1.21 255.255.255.252
no ip directed-broadcast
ip router isis
tag-switching ip

frame-relay interface-dlci 306
!
router isis
net 49.0001.0000.0000.0003.00
is-type level-1
!
ip classless
!
end

```

Pulligny

```

Current configuration:
!
version 12.1
!
hostname Pulligny
!
!
ip cef

!
!
interface Loopback0
 ip address 10.10.10.2 255.255.255.255
!
interface Serial0/1
 no ip address
 encapsulation frame-relay
 random-detect
!
interface Serial0/1.1 point-to-point
 description link to Pauillac
 ip address 10.1.1.2 255.255.255.252
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 201
!
interface Serial0/1.2 point-to-point
 description link to Pomerol
 ip address 10.1.1.10 255.255.255.252
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 203
!
router isis
 passive-interface Loopback0
 net 49.0001.0000.0000.0002.00
 is-type level-1
!
ip classless
!
end

```

Pauillac

```

!
version 12.1
!
hostname pauillac
!
!
ip cef

!
!
interface Loopback0
 ip address 10.10.10.1 255.255.255.255
 ip router isis
!
interface Serial0/0
 no ip address
 encapsulation frame-relay
 no ip mroute-cache
 tag-switching ip
 no fair-queue
!
interface Serial0/0.1 point-to-point
 description link to Pomerol
 bandwidth 512
 ip address 10.1.1.1 255.255.255.252
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 102
!
interface Serial0/0.2 point-to-point
 description link to Pulligny ip address 10.1.1.5 255.255.255.252

 ip router isis
 tag-switching ip
 frame-relay interface-dlci 103
!
interface Serial0/0.3 point-to-point
 description link to Pescara
 bandwidth 512
 ip address 10.1.1.13 255.255.255.252
 ip router isis
 tag-switching ip
 frame-relay interface-dlci 104
!
router isis
 net 49.0001.0000.0000.0001.00
 is-type level-1
!
ip classless
!
end

```

This section provides information you can use to confirm your configuration is working properly.

The Output Interpreter Tool (<https://www.cisco.com/cgi-bin/Support/OutputInterpreter/home.pl>) (registered (<https://tools.cisco.com/RPF/register/register.do>) customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

- **show ip vrf** ([/swi_s2.html#wp1063588](https://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_s2.html#wp1063588))—Verifies that the correct VRF exists.
- **show ip vrf interfaces**—Verifies the activated interfaces.
- **show ip route vrf** ([/swi_s2.html#wp1057479](https://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_s2.html#wp1057479)) **Customer_A**—Verifies the routing information on the PE routers.
- **traceroute vrf** ([/ipv6_17.html#wp2446869](https://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_17.html#wp2446869)) **Customer_A 200.0.6.1**—Verifies the routing information on the PE routers.
- **show ip bgp vpnv4 tag**—Verifies the BGP.
- **show ip cef vrf** ([/swi_s2.html#wp1081015](https://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_s2.html#wp1081015)) **Customer_A 200.0.6.1 detail**—Verifies the routing information on the PE routers.

More commands are detailed in the MPLS VPN Solution Troubleshooting Guide (https://www.cisco.com/en/US/docs/net_mgmt/vpn_solutions_center/1.1/user/guide/VPN_UG7.html).

The following is sample command output of the **show ip vrf** command.

```
Pescara#show ip vrf
Name                Default RD          Interfaces
Customer_A          100:110             Loopback101
Customer_B          100:120             Loopback102
```

The following is sample command output of the **show ip vrf interfaces** command.

```
Pesaro#show ip vrf interfaces
Interface            IP-Address          VRF                  Protocol
Loopback101         200.0.6.1           Customer_A            up
Loopback111         200.1.6.1           Customer_A            up
Loopback102         200.0.6.1           Customer_B            up
```

The following **show ip route vrf** commands show the same prefix 200.0.6.0/24 in both the outputs. This is because the remote PE has the same network for two customers, Customer_A and Customer_B, which is allowed in a typical MPLS VPN solution.

```
Pescara#show ip route vrf Customer_A
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR

Gateway of last resort is not set

C    200.0.4.0/24 is directly connected, Loopback101
B    200.0.6.0/24 [200/0] via 10.10.10.6, 05:10:11
B    200.1.6.0/24 [200/0] via 10.10.10.6, 04:48:11

Pescara#show ip route vrf Customer_B
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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    200.0.4.0/24 is directly connected, Loopback102

B    200.0.6.0/24 [200/0] via 10.10.10.6, 00:03:24
```

By running a traceroute between two sites of Customer_A, it is possible to see the label stack used by the MPLS network (if it is configured to do so by **mpls ip ttl ...**).

```
Pescara#traceroute vrf Customer_A 200.0.6.1

Type escape sequence to abort.
Tracing the route to 200.0.6.1

 0  10.1.1.13 [MPLS: Labels 20/26 Exp 0] 400 msec 276 msec 264 msec
 1  10.1.1.6 [MPLS: Labels 18/26 Exp 0] 224 msec 460 msec 344 msec
 2  200.0.6.1 108 msec * 100 msec
```

Note: Exp 0 is an experimental field used for Quality of Service (QoS).

There is currently no specific troubleshooting information available for this configuration.

Title: Configuring MPLS VPNs > Troubleshooting Any Transport over MPLS Based VPNs

Content courtesy of: <http://www.ciscopress.com/articles/article.asp?p=391649&seqNum=2>

Configuring MPLS VPNs

Misconfiguration is a common cause of problems with MPLS VPNs. In this section, therefore, MPLS VPN configuration is discussed.

When configuring an MPLS VPN, there are three types of devices that must be configured, the CE router, the PE router, and the P router. The configuration of each of these devices is discussed in this section.

It may be useful to reference Figure 6-31 on page 476 while reading this section. Note, however, that not all configuration discussed in this section is illustrated.

Configuring the CE Router

Configuration of the CE router is standard; nothing special is required. The only restriction is that the routing protocol used between the CE and PE routers must currently be RIP version 2, EIGRP, OSPF, or EBGP. Static routes can also be used.

Configuring the PE Router

Configuration of the PE router is much more complicated than that of the CE router.

The 12 basic steps involved are summarized as follows:

Step 1

Configure the loopback interface to be used as the BGP update source and LDP router ID.

Step 2

Enable CEF.

Step 3

Configure the label distribution protocol.

Step 4

Configure the TDP/LDP router-id (optional).

Step 5

Configure MPLS on core interfaces.

Step 6

Configure the MPLS VPN backbone IGP.

Step 7

Configure global BGP parameters.

Step 8

Configure MP-BGP neighbor relationships.

Step 9

Configure the VRF instances.

Step 10

Configure VRF interfaces.

Step 11

Configure PE-CE routing protocols / static routes.

Step 12

Redistribute customer routes into MP-BGP.

The sections that follow examine each step in detail.

Step 1: Configure the Loopback Interface to Be Used as the BGP Update Source and LDP Router ID

A loopback interface should be configured to act as the update source for BGP sessions, as well as the LDP router ID.

Ensure that the IP address on the loopback interface is configured with a 32-bit mask. This will prevent a lot of problems later.

For example, if the IGP used in the MPLS backbone is OSPF, and the loopback interface is not configured with a 32-bit mask, the PE router will advertise a label binding for the loopback address with the mask as specified on the loopback interface. The route advertised in OSPF to neighboring routers, on the other hand, will include a 32-bit mask. This is because OSPF advertises loopback addresses with a 32-bit mask by default (irrespective of the

configured mask). The neighboring routers (LSRs) will create a label binding that corresponds to the OSPF route advertised by the PE router (using the advertised 32-bit mask), but because the label binding advertised by the PE router uses the configured non-32-bit mask, an LSP failure will result.

There are two ways around this: either configure the loopback interface on the PE router with a 32-bit mask, or configure the **ip ospf network point-to-point** command on the loopback interface, which will cause OSPF to advertise the mask as it is actually configured.

It is also very important to configure just one update source for MP-BGP if you intend to configure MVPNs. More than one update source can break MVPN.

Example 6-1 shows the configuration of the loopback interface to be used as the BGP update source and LDP router ID.

Example 6-1 Configuration of the Loopback Interface

```
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
```

It's good practice to allocate one address block to use for all PE router loopback interface addresses.

Note that PE router loopback addresses should not be summarized in the core because this will break LSPs within the MPLS backbone.

Step 2: Enable CEF

Be sure to enable CEF. If CEF is not enabled on the PE router, MPLS will not function.

Example 6-2 shows how to enable CEF on the router.

Example 6-2 Enabling CEF

```
ip cef [distributed]
```

Note the keyword **distributed**. This is used to enable distributed CEF (dCEF). dCEF is available on high-end platforms such as the 12000 GSR and 7500 series.

Step 3: Configure the LDP

If using LDP in the MPLS backbone, you should configure LDP next. Note that TDP is the default label distribution protocol on Cisco routers. Example 6-3 shows the global configuration of LDP as the label distribution protocol.

Example 6-3 Configuration of LDP as the Label Distribution Protocol

```
mpls label protocol ldp
```

Step 4: Configure the TDP/LDP Router ID (Optional)

The next step is to configure the TDP/LDP router ID. This step is optional, but it can make the troubleshooting process easier if you are able to easily identify TDP/LDP routers in the network.

Example 6-4 shows the configuration of the LDP router ID.

Example 6-4 Configuration of the LDP Router ID

```
mpls ldp router-id Loopback0 [force]
```

In Example 6-4, the IP address on interface loopback 0 is configured as the LDP router ID. Note the optional **force** keyword, which ensures that the IP address on interface loopback 0, and not the IP address of any other interface, becomes the LDP router ID.

If the LDP router ID is not explicitly configured as shown in Example 6-4, the LDP ID will become the highest loopback interface address or, in the absence of a loopback interface, the highest IP address configured on a physical interface. It is definitely a good idea to ensure that the LDP ID corresponds to a loopback interface because loopback interfaces are always in an up state.

Step 5: Configure MPLS on Core Interfaces

The next step is to enable MPLS on interfaces connected to other PE and Prouters. Note that when MPLS is enabled on the first interface, it is also globally enabled on the router.

Example 6-5 shows the configuration of MPLS on core frame-mode interfaces.

Example 6-5 Configuring MPLS on Core Frame-Mode Interfaces

```
interface Serial4/0
 mpls ip
```

As previously mentioned, ATM interfaces can be configured for either frame-mode or cell-mode.

Frame-mode can be configured over ATM PVCs between edge LSRs. In this case, intervening ATM switches do not participate in MPLS at all and do not need to be MPLS-enabled.

Example 6-6 shows the configuration of an ATM interface for frame-mode MPLS.

Example 6-6 Configuration of Frame-mode MPLS on an ATM Interface

```
interface ATM3/0.1 point-to-point
 ip address 10.20.100.1 255.255.255.0
 pvc 1/50
 encapsulation aal5snap
 !
 mpls ip
```

In Example 6-6, MPLS is enabled on an ATM PVC with VPI/VCI 1/50. Note that the subinterface type is **point-to-point** and that the **mpls ip** command is configured on the subinterface.

ATM interfaces can also be configured for cell-mode MPLS. These interfaces are known as Label Controlled ATM (LC-ATM) interfaces.

Example 6-7 shows the configuration of cell-mode MPLS on an ATM interface of an IOS router.

Example 6-7 Configuration of Cell-Mode MPLS on an ATM Interface

```
interface ATM3/0.1 mpls
ip address 10.20.90.1 255.255.255.0
mpls ip
```

In Example 6-7, the subinterface type is **mpls**. Also note the command **mpls ip** on the subinterface itself.

When cell-mode MPLS is enabled on an ATM interface, a PVC with VPI/VCI 0/32 (by default) is automatically created for control plane traffic.

Step 6: Configure the MPLS VPN Backbone IGP

Although it is possible to use any IGP for IP reachability within the MPLS VPN backbone, IS-IS and OSPF are the two most commonly chosen because they are the only two IGPs that currently support MPLS traffic engineering.

The OSPF and IS-IS protocol configurations covered in the two sections that follow are only examples.

IS-IS

The configuration of IS-IS on the PE router is, to a large extent, standard.

Example 6-8 shows the configuration of IS-IS for IP reachability within the MPLS VPN backbone.

Example 6-8 Configuration of IS-IS as the MPLS VPN Backbone IGP

```
router isis
passive-interface Loopback0
net 49.0001.0000.0000.0001.00
is-type level-2-only
metric-style wide
```

The **router isis** command enables IS-IS on the PE router.

Interface loopback 0 is then enabled for IS-IS using the **passive-interface loopback0** command. Note that because the interface is passive, no IS-IS packets are needlessly sent on the interface.

Be sure to advertise the BGP update source into the IS-IS. If the update source is not advertised, MPLS VPNs will break.

The third command in the configuration is **net 49.0001.0000.0000.0001.00**. This is used to configure the network entity title (NET). **49.0001** is the area ID, **0000.0000.0001** is the system ID, and **.00** is the selector value.

The next command, **is-type level-2-only**, configures the PE router as a Level 2 (backbone) router only.

Finally, the command **metric-style wide** configures the router to send and to receive only new style 24- or 32-bit metrics. Support for new style metrics are essential if you are intending to use MPLS traffic engineering.

Ensure that all IS-IS routers in the backbone are configured to support standard or wide metrics (or both). IS-IS must also be enabled on each of its core interfaces.

Example 6-9 shows the configuration of IS-IS on core interfaces.

Example 6-9 Configuration of IS-IS on Core Interfaces

```
interface FastEthernet1/0
ip router isis
```

In Example 6-9, IS-IS for IP is enabled on interface FastEthernet1/0 using the command **ip router isis**.

OSPF

OSPF configuration for the backbone is, again, fairly standard.

Example 6-10 shows the configuration of OSPF for IP reachability within the MPLS VPN backbone.

Example 6-10 Configuration of OSPF as the MPLS VPN Backbone IGP

```
router ospf 100
passive-interface Loopback0
network 10.0.0.0 0.255.255.255 area 0
```

The command **router ospf 100** enables OSPF process 100 on the PE router.

All backbone interfaces in network 10.0.0.0/8 are placed in OSPF area 0 using the **network 10.0.0.0 0.255.255.255 area 0** command.

Finally, the **passive-interface Loopback0** prevents the sending of OSPF packets on interface loopback 0.

Be sure to advertise the BGP update source into OSPF. If the update source is not advertised, MPLS VPNs will break.

Note that if your network consists of ATM-LSRs, make sure that summarization of IGP routes is not configured on PE routers. This is because ATM-LSRs have no "IP intelligence" on the data plane.

Step 7: Configure Global BGP Parameters

MP-BGP is used to advertise customer routes across the MPLS VPN backbone between PE routers. The configuration of MP-BGP is a two-step process, with neighbors being configured globally and then activated for MP-BGP route exchange under the VPNv4 (VPN-IPv4) address family.

Example 6-11 shows global BGP configuration on the PE router.

Example 6-11 Global BGP Configuration on the PE Router

```
router bgp 64512
no synchronization
neighbor 10.1.1.4 remote-as 64512
neighbor 10.1.1.4 update-source Loopback0
neighbor 10.1.1.6 remote-as 64512
neighbor 10.1.1.6 update-source Loopback0
no auto-summary
```

The first command, **router bgp autonomous_system**, enables BGP on the PE router.

Global IGP synchronization is then disabled using the **no synchronization** command.

The command **neighbor ip_address remote-as** *autonomous_system* configures the IP address and autonomous system of the remote PE router or route reflector.

Next comes the **neighbor ip_address update-source Loopback0**. This configures interface loopback 0 as the update source for the BGP session.

It is highly recommended that a single interface (preferably with a 32-bit mask) be configured as the MP-BGP update source. Not doing so might result in broken MPLS VPNs and MVPNs.

The command **no auto-summary** is used to ensure that routes redistributed into BGP (via the **redistribute** command) are not summarized at major network boundaries.

One other command that might be useful on the PE router is the **no bgp default ipv4-unicast** command, which disables the exchange of global BGP (Internet) routes. Only MP-BGP, and not global BGP, routes are required for MPLS VPN functionality.

Step 8: Activate MP-BGP Neighbors

MP-BGP is used for the exchange of VPN routes between the PE routers. MP-BGP route exchange must be activated under the VPNv4 address family.

Example 6-12 shows the activation of MP-BGP route exchange.

Example 6-12 Activation of MP-BGP Route Exchange

```
router bgp 64512
!
address-family vpnv4
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 send-community extended
neighbor 10.1.1.6 activate
neighbor 10.1.1.6 send-community extended
no auto-summary
exit-address-family
```

The command **address-family vpnv4** is used to enter the VPNv4 address family configuration mode.

The **neighbor ip_address activate** is used to activate MP-BGP route exchange.

The command **neighbor ip_address send-community extended** is configured by default and enables the exchange of BGP extended communities, such as route target and site of origin.

TIP

If you want BGP peers to also exchange standard BGP communities, you must use the keyword **both** in place of the **extended** keyword.

Finally, the command **no auto-summary** command specifies that redistributed routes should not be summarized at major network boundaries. This command is configured by default.

Note that if route reflectors are used for VPN route exchange between PE routers, ensure that they are also configured for MP-BGP route exchange between route reflector clients.

Step 9: Configure the VRF Instances

The next step is the configuration of the VRFs, as demonstrated in Example 6-13.

Example 6-13 Configuration of a VRF

```
ip vrf mjlnet_VPN
rd 64512:100
route-target export 64512:100
route-target import 64512:100
```

The first line of the configuration enables a VRF named mjlnet_VPN.

Step 10: Configure VRF Interfaces

After configuring the VRF, the next step is to associate customer interfaces with it.

Example 6-14 shows the configuration of VRF interfaces on PE routers.

Example 6-14 Configuration of VRF Interfaces

```
interface Serial4/1
ip vrf forwarding mjlnet_VPN
```

The **ip vrf forwarding mjlnet_VPN** command associates an interface with a customer VRF. In this case, interface serial 4/1 is associated with VRF mjlnet_VPN.

Step 11: Configure PE-CE Routing Protocols / Static Routes

Configuration of the PE-CE routing protocol varies according to whether RIP version 2, EIGRP, OSPF, or EIGRP is being used. Static routes can also be used for PE-CE connectivity.

The sections that follow describe configuration of the various PE-CE routing protocols.

RIP Version 2

When configuring RIP version 2 for PE-CE routing, most of the configuration is under the IPv4 address family.

Example 6-15 shows the configuration of RIP version 2 for PE-CE routing.

Example 6-15 Configuration of RIP Version 2 for PE-CE Routing

```
router rip
version 2
!
address-family ipv4 vrf mjlnet_VPN
version 2
redistribute bgp 64512 metric transparent
network 172.16.0.0
no auto-summary
exit-address-family
```

The command **router rip** enables RIP on the PE router. RIP version 2 is then configured using the command **version 2**.

Next comes the **address-family ipv4 vrf vrf_name** command. RIP configuration for the VRF is configured under the IPv4 address family.

By specifying **version 2** globally (directly under **router rip**), it is inherited by all the address families configured under RIP.

Under the address family, be sure to specify redistribution from (MP-BGP or BGP into RIP. Alternatively, you can originate a default route into RIP if it is a large network. Remember that customer routes are advertised between PE routers using MP-BGP. These routes are then imported into the customer VRFs. The command **redistribute bgp autonomous_system metric transparent** can then be used to redistribute these routes into RIP for advertisement to the attached customer site or sites.

Note the use of **metric transparent**. RIP metrics are preserved when they are advertised in MP-BGP (they are copied into the MED attribute), which ensures that these metrics are redistributed back into RIP unmodified.

Make sure that a metric, whether a specific metric or the keyword **transparent**, is configured when redistributing MP-BGP routes into RIP. If one is not specified, the routes may not be redistributed.

The rest of the configuration is pretty standard stuff, with the **network** command used to specify the networks enabled for RIP, and the **no auto-summary** command used to ensure that networks are not summarized at major network boundaries. Note that **no auto-summary** is on by default under the address family.

EIGRP

Configuration of EIGRP is similar to RIP, with most parameters configured under the IPv4 address-family.

Example 6-16 shows a sample configuration of EIGRP for PE-CE routing.

Example 6-16 Configuration of EIGRP for PE-CE Routing

```
router eigrp 10
 no auto-summary
 !
 address-family ipv4 vrf mjlnet_VPN
 redistribute bgp 64512 metric 1 1 255 1 1500
 network 172.16.0.0
 no auto-summary
 autonomous-system 100
 exit-address-family
```

The **router eigrp 10** command enables EIGRP autonomous system 10 on the PE router.

The second command is **no auto-summary**. This ensures that networks are not summarized at major network boundaries.

The configuration of EIGRP for PE-CE connectivity itself is specified under an IPv4 address-family (**address-family ipv4 vrf vrf_name**). Each customer VRF requires a separate address family.

The configuration under the IPv4 address family starts with redistribution of MP-BGP routes from other customer sites into EIGRP using **redistribute bgp autonomous_system metric metric** (bandwidth, delay, reliability, load, and MTU). Make sure that you specify a metric when redistributing MP-BGP routes into EIGRP. If one is not specified, redistribution may fail. Next is the **network** command, which is used to specify the networks enabled for EIGRP. The **no auto-summary** command is configured by default under the address family.

The final command under the address family is **autonomous-system autonomous_system**. This is the EIGRP autonomous system number for the customer VPN. If this is not the same as that configured as that on the CE router, then no adjacency will be formed.

OSPF

When configuring OSPF, a separate OSPF process must be configured for each customer VRF running OSPF as the PE-CE routing protocol.

Example 6-17 shows the configuration of OSPF for customer site routing.

Example 6-17 Configuration of OSPF for PE-CE Routing

```
router ospf 100 vrf mjlnet_VPN
 redistribute bgp 64512 subnets
 network 172.16.4.0 0.0.0.255 area 0
```

The first command in the configuration is **router ospf process_ID vrf vrf_name**. In this case, OSPF process 100 is configured for VRF mjlnet_VPN.

The third command is **redistribute bgp autonomous_system subnets**. This is used to configure redistribution of MP-BGP (routes from remote sites) into OSPF. Note the **subnets** keyword. This ensures that subnets, and not just major networks, are redistributed.

EBGP

Configuration of EBG for PE-CE connectivity is pretty straightforward. Again, most of the configuration is under the IPv4 address family.

Example 6-18 shows the configuration of EBG for PE-CE routing.

Example 6-18 Configuration of EBG for PE-CE Routing

```
router bgp 64512
 !
 address-family ipv4 vrf mjlnet_VPN
 neighbor 172.16.4.2 remote-as 65001
 neighbor 172.16.4.2 activate
 no auto-summary
 no synchronization
 exit-address-family
```

The **address-family ipv4 vrf vrf_name** command is used to enter the IPv4 address family configuration mode.

The first command under the IPv4 address family is **neighbor ip_address remote-as autonomous_system**. This

configure the IP address and autonomous system of the CE router.

Next is **neighbor ip_address activate**. This activates the BGP session with the CE router.

Finally, the **no auto-summary** and **no synchronization** commands are used to disable auto summarization at major network boundaries for routes redistributed via the **redistribute** command into BGP, and to disable IGP synchronization. These two commands are enabled by default.

Note that unlike for other PE-CE routing protocols, redistribution is unnecessary from MP-BGP into EBGP.

Static Routes

Static routes can also be used for PE-CE connectivity. Example 6-19 shows configuration of static routes for PE-CE connectivity.

Example 6-19 Configuration of Static Routes for PE-CE Connectivity

```
ip route vrf mjlnet_VPN 172.16.1.0 255.255.255.0 172.16.4.2 [permanent]
```

Configuration of static routes is the same as that for regular static routes with the network, mask, and next-hop specified. The **vrf** keyword must be used, however, to ensure that the static route is placed in the VRF specified (in this case, mjlnet_VPN).

Note also the **permanent** keyword. This can optionally be used to ensure that the route will remain in the VRF even if reachability to the next hop is lost. This can be important for stability when redistributing static routes into MP-BGP.

Step 12: Redistribute Customer Routes into MP-BGP

The final step is to configure the redistribution of customer routes into MP-BGP, as demonstrated in Example 6-20.

Example 6-20 Redistribution of Customer Routes into MP-BGP

```
router bgp 64512
!
address-family ipv4 vrf mjlnet_VPN
 redistribute rip
 no auto-summary
 no synchronization
 exit-address-family
```

The **address-family ipv4 vrf vrf_name** command is used to enter the IPv4 address family configuration mode.

The **redistribute rip** command is used to redistribute customer RIPv2 routes into MP-BGP.

If the PE-CE routing protocol is EIGRP, the command **redistribute eigrp autonomous_system** is used. Ensure that the autonomous system number configured corresponds to that specified under the EIGRP IPv4 address family.

For OSPF, the command **redistribute ospf process_ID match internal external 1 external 2** can be used. Note that in this case, internal and external type 1 and 2 routes are redistributed.

Finally, if static routes are being used, the command **redistribute static** can be used. It is also worth noting that if EBGP is being used, redistribution is not required.

Finally, the **no auto-summary** and **no synchronization** commands are defaults that specify that redistributed routes should not be summarized at major network boundaries, and that synchronization should be disabled.

That concludes the configuration of the PE router.

PE Router Sample Configuration

Example 6-21 shows a complete sample configuration of a PE router.

Example 6-21 Complete Sample Configuration of a PE Router

```

Chengdu_PE#show running-config
Building configuration...
Current configuration : 3434 bytes
!
version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Chengdu_PE
!
logging buffered 16384 debugging
enable secret 5 $1$4pDG$SmVThUgDZG33pNYZ20.UKU/
!
ip subnet-zero
no ip source-route
!
! Enable Cisco Express Forwarding (CEF)
ip cef
!
!
no ip finger
no ip bootp server
!
! Configure the VPN Routing and Forwarding (VRF) instances
ip vrf mjlnet_VPN
rd 64512:100
route-target export 64512:100
route-target import 64512:100
!
ip vrf cisco_VPN
rd 64512:200
route-target export 64512:200
route-target import 64512:200
!
! Configure the label distribution protocol
mpls label protocol ldp
no mpls tra&#x0000;mc-eng auto-bw timers frequency 0
!
! Configure the TDP/LDP router-id (tag-switching tdp router-id = mpls ldp router-id)
tag-switching tdp router-id Loopback0 force
!
! Configure the loopback interface to be used as the BGP update source and LDP router ID
interface Loopback0
ip address 10.1.1.1 255.255.255.255
no ip directed-broadcast
!
! Configure MPLS on core interfaces
interface FastEthernet1/0
ip address 10.20.10.1 255.255.255.0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
ip router isis
tag-switching ip
no cdp enable
!
! Configure VRF interfaces
interface Serial4/1
ip vrf forwarding mjlnet_VPN
ip address 172.16.4.1 255.255.255.0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
encapsulation ppp
no cdp enable
!
interface Serial4/2
ip vrf forwarding cisco_VPN
ip address 192.168.4.1 255.255.255.0
no ip redirects
no ip directed-broadcast
no ip proxy-arp
no cdp enable
!
! Configure PE-CE routing protocol for cisco_VPN
router ospf 200 vrf cisco_VPN
log-adjacency-changes
redistribute bgp 64512 subnets
network 192.168.4.0 0.0.0.255 area 0
!
! Configure the MPLS VPN backbone IGP
router isis
passive-interface Loopback0
net 49.0001.0000.0000.0001.00
is-type level-2-only
metric-style wide
!
! Configure PE-CE routing protocol for mjlnet_VPN
router rip
version 2
!

```

```

address-family ipv4 vrf mjlnet_VPN
version 2
redistribute bgp 64512 metric transparent
network 172.16.0.0
no auto-summary
exit-address-family
!
! Configure basic BGP parameters
router bgp 64512
no synchronization
bgp log-neighbor-changes
neighbor 10.1.1.4 remote-as 64512
neighbor 10.1.1.4 update-source Loopback0
neighbor 10.1.1.6 remote-as 64512
neighbor 10.1.1.6 update-source Loopback0
no auto-summary
!
! Configure MP-BGP neighbor relationships
address-family vpnv4
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 send-community extended
neighbor 10.1.1.6 activate
neighbor 10.1.1.6 send-community extended
no auto-summary
exit-address-family
!
! Redistribute customer routes into MP-BGP
address-family ipv4 vrf cisco_VPN
redistribute ospf 200 match internal external 1 external 2
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf mjlnet_VPN
redistribute rip
no auto-summary
no synchronization
exit-address-family
!
ip classless
!
logging trap debugging
!
!
line con 0
exec-timeout 0 0
password 7 1511021F0725
login
line aux 0
line vty 0 4
password 7 110A1016141D
login
!
end

```

You might notice that a number of commands discussed in this section are not immediately apparent in the configuration shown in Example 6-21. An example is the **mpls ip** command. In fact, the **mpls** keyword is translated into the **tag-switching** keyword. This allows backward compatibility with versions of the Cisco IOS software that do not support the **mpls** keyword.

The only exception to this is the **mpls label protocol** command, which remains in its original form.

Configuring the P Router

Configuration of P routers is, by comparison with that of PE routers, very simple.

The six basic steps in the configuration are as follows:

Step 1

Configure the loopback interface to be used as the LDP router ID.

Step 2

Enable CEF.

Step 3

Configure the label distribution protocol.

Step 4

Configure the TDP/LDP router ID (optional).

Step 5

Configure MPLS on core interfaces.

Step 6

Configure IS-IS or OSPF as the MPLS VPN backbone IGP.

As you can see, these six steps are identical to the first six steps for the configuration of the PE router. Please refer to the previous section for an explanation of each of these steps.

Example 6-22 shows a complete sample configuration of a P router.

Example 6-22 Complete Sample Configuration of a P Router

```

Chengdu_P#show running-config
Building configuration...
Current configuration : 1991 bytes
!
version 12.0
service nagle
no service pad
service tcp-keepalives-in
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
!
hostname Chengdu_P
!
logging buffered 16384 debugging
no logging console
enable secret 5 $1$4pDG$mVThUgDZG33pNYZ20.UKU/
!
ip subnet-zero
no ip source-route
!
! Enable Cisco Express Forwarding (CEF)
ip cef
!
!
no ip finger
no ip bootp server
!
! Configure the label distribution protocol
mpls label protocol ldp
no mpls traffic-eng auto-bw timers frequency 0
!
! Configure the TDP/LDP router-id
tag-switching tdp router-id Loopback0 force
!
! Configure the loopback interface to be used as the LDP router id
interface Loopback0
 ip address 10.1.1.2 255.255.255.255
 no ip directed-broadcast
!
! Configure MPLS on core interfaces
interface FastEthernet1/0
 ip address 10.20.10.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip router isis
 tag-switching ip
 no cdp enable
!
interface Serial1/0
 ip address 10.20.20.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip router isis
 encapsulation ppp
 tag-switching ip
 no fair-queue
 no cdp enable
!
interface Serial1/1
 ip address 10.20.40.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip router isis
 encapsulation ppp
 tag-switching ip
 no fair-queue
 no cdp enable
!
!
! Configure IS-IS as the MPLS VPN backbone IGP
router isis
 passive-interface Loopback0
 net 49.0001.0000.0000.0002.00
 is-type level-2-only
 metric-style wide
!
ip classless
!
logging trap debugging
!
line con 0
 exec-timeout 0 0
 password 7 1511021F0725
 login
line aux 0
line vty 0 4
 password 7 110A1016141D
 login
!
end

```

Notice again that the **mpls** keyword has been converted into the **tag-switching** keyword for backward compatibility. This completes the configuration of MPLS VPNs.

Title: BGP / MPLS Layer 3 VPNs Practical Configuration | Noction

Content courtesy of: <https://www.noction.com/blog/bgp-mpls-layer3-vpn-practical-configuration>

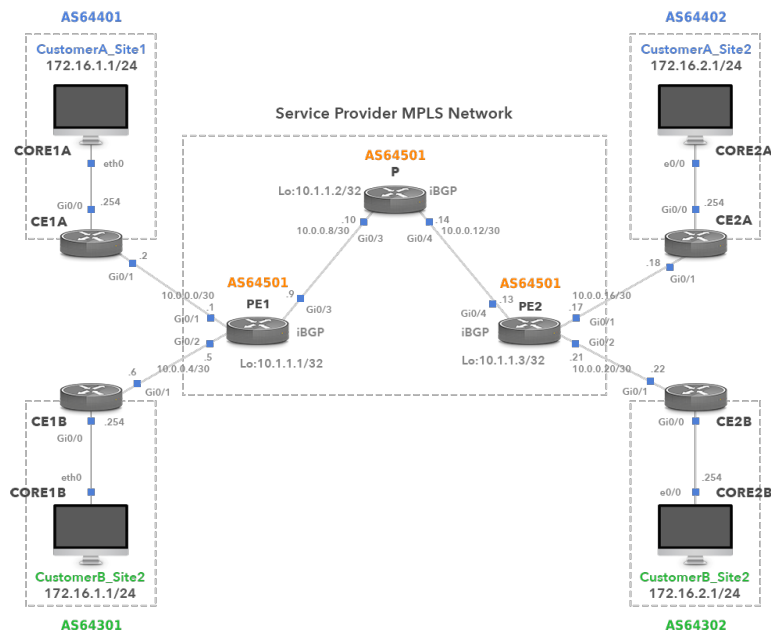


In our previous (<https://www.noction.com/blog/bgp-mpls-layer3-vpn-practical-configuration>)

blog article we've discussed the benefits and the fundamental principles of BGP/MPLS L3 VPNs. We have covered the definition of the basic terms such as the Route Distinguisher (RD), the Route Target (RT) and the VPN-IPv4 prefix. This post goes further. We are going to support the theory behind the BGP/MPLS L3 VPNs with a practical configuration.

Our lab network consists of PE1, PE2 and P routers, which are part of a service provider's MPLS network. There are two remote sites: 1 (with CustomerA_Site1 and CustomerB_Site1) and 2 (with CustomerA_Site2 and CustomerB_Site2) both connected to a service provider's MPLS network. Our goal is to interconnect the remote customer sites so that they can communicate privately over a shared medium. This is where BGP/MPLS VPNs come in handy, separating traffic from both customers, using a combination of the VRF, MPLS and MP-BGP.

The customers use private addresses inside their routing domains, which overlap each other. For instance, both customers use the same prefix 172.16.1.0/24 for site 1 and 172.16.2.0/24 for site 2.



Picture 1: Network Topology

IGP Configuration on P and PE routers

First, we will configure the IGP protocol among all P and PE routers to support LDP and BGP adjacencies within the provider network. Even IGP or static routes might be a choice. We can configure EIGRP, as all routers in our example are from Cisco.

```
PE1(config)# router eigrp 1
PE1(config-router)# network 10.0.0.8 0.0.0.3
PE1(config-router)# network 10.1.1.1 0.0.0.0

P(config)# router eigrp 1
P(config-router)# network 10.0.0.8 0.0.0.3
P(config-router)# network 10.0.0.12 0.0.0.3
P(config-router)# network 10.1.1.2 0.0.0.0

PE2(config)# router eigrp 1
PE2(config-router)# network 10.0.0.12 0.0.0.3
PE2(config-router)# network 10.1.1.3 0.0.0.0
```

eBGP Configuration On Customer Routers

Now let's configure the eBGP adjacency between CE and PE routers. BGP AS numbers at each customer site must be unique and differ from the provider's ASN. For instance, the customer A BGP AS number is 64401 at site 1 and ASN 64402 at site 2. We also advertise each customers' subnet from CE to PE router with the following network commands:

```

CE1A(config)# router bgp 64401
CE1A(config-router)# neighbor 10.0.0.1 remote-as 64501
CE1A(config-router)# network 172.16.1.0 mask 255.255.255.0

CE2A(config)# router bgp 64402
CE2A(config-router)# neighbor 10.0.0.17 remote-as 64501
CE2A(config-router)# network 172.16.2.0 mask 255.255.255.0

CE1B(config)# router bgp 64301
CE1B(config-router)# neighbor 10.0.0.5 remote-as 64501
CE1B(config-router)# network 172.16.1.0 mask 255.255.255.0

CE2B(config)# router bgp 64302
CE2B(config-router)# neighbor 10.0.0.21 remote-as 64501
CE2B(config-router)# network 172.16.2.0 mask 255.255.255.0

```

Configuring MP-BGP on PE Routers

Multiprotocol BGP is explained in RFC 4760 (<https://tools.ietf.org/html/rfc4760>). It defines the extensions to BGP-4 to enable it to carry the routing information for multiple Network Layer protocols (e.g., IPv6, L3VPN). Therefore, we will configure the MP-BGP to distribute customers' prefixes. The extensions are backward compatible. A router that supports the extensions can interoperate with a router that doesn't support the extensions.

iBGP neighborship is formed between the PE routers, using ASN 64501. No BGP is configured on router P.

```

PE1(config)# router bgp 64501
PE1(config-router)# neighbor 10.1.1.3 remote-as 64501
PE1(config-router)# neighbor 10.1.1.3 update-source lo0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 10.1.1.3 activate
PE1(config-router-af)# exit

```

Note: The command *neighbor 10.1.1.3 send-community extended* is automatically configured under the address-family vpnv4 section.

```

PE2(config)# router bgp 64501
PE2(config-router)# neighbor 10.1.1.1 remote-as 64501
PE2(config-router)# neighbor 10.1.1.1 update-source lo0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 10.1.1.1 activate
PE2(config-router-af)# exit

```

Note: The command *neighbor 10.1.1.1 send-community extended* is automatically configured under the address-family vpnv4 section.

Enable MPLS on PE and P Routers

We need to enable MPLS in a provider's network. Customers' data are then switched in the MPLS network based on the outer (LSP) label. We will enable MPLS on a provider's P router and on PE routers.

```

PE1(config)# interface GigabitEthernet 0/3
PE1(config-if)# mpls ip

P(config)# interface GigabitEthernet 0/3
P(config-if)# mpls ip
P(config)# interface GigabitEthernet 0/4
P(config-if)# mpls ip

PE2(config)# interface GigabitEthernet 0/4
PE2(config-if)# mpls ip

```

Create and Assign VRFs

Customers' forwarding tables are separated by using the VPN routing and forwarding table (VRF) concept on the PE router. One VRF is configured on the PE router for each customer. The Router's PE interface that connects CE router to provider's MPLS network is then assigned to the customer VRF.

Route distinguisher is added on the PE router to customer's prefix to distinguish the same prefix and mask in a different VRF. For instance, PE1 router announces prefixes RD1:172.16.10/24 and RD2:172.16.1.0/24 along with VPN label to PE2 router inside the BGP update message. The RD is used to distinguish the prefixes and it has no impact how the routes are installed into the VRFs.



BGP in Large Networks Implementation Guide

[DOWNLOAD GUIDE](#)

(https://www.noction.com/resource_center/bgp-in-large-networks?utm_source=Reference-from-VPN-blog&utm_campaign=Reference-from-VPN-blog&utm_medium=Reference-from-VPN-blog)

The route target is an extended community attribute used for the import/export of VPN routes. For instance, a VPN prefix 172.16.1.0/24 sent from PE1 to PE2 inside of the MP-BGP update message and carrying the route-target 64501:1 is imported into VRF Customer A on PE2.

```

PE1(config)# ip vrf CustomerA
PE1(config-vrf)# rd 64501:1
PE1(config-vrf)# route-target both 64501:1

```

Note: the commands *route-target export 64501:1* and *route-target import 64501:1* are automatically configured under vrf configuration.

```

PE1(config-vrf)# ip vrf CustomerB
PE1(config-vrf)# rd 64501:2
PE1(config-vrf)# route-target both 64501:2

```

Note: the commands *route-target export 64501:2* and *route-target import 64501:2* are automatically configured under vrf configuration.

Now we need to assign L3 interfaces to customer VRF.

```
PE1(config)# interface gigabitEthernet 0/1
PE1(config-if)# ip vrf forwarding CustomerA
PE1(config-if)# ip address 10.0.0.1 255.255.255.252

PE1(config)# interface gigabitEthernet 0/2
PE1(config-if)# ip vrf forwarding CustomerB
PE1(config-if)# ip address 10.0.0.5 255.255.255.252
```

We will create the same VRFs on PE2 and assign interfaces to VRFs.

```
PE2(config)# ip vrf CustomerA
PE2(config-vrf)# rd 64501:1
PE2(config-vrf)# route-target both 64501:1
PE2(config-vrf)# ip vrf CustomerB
PE2(config-vrf)# rd 64501:2
PE2(config-vrf)# route-target both 64501:2

PE1(config)# interface gigabitEthernet 0/1
PE1(config-if)# ip vrf forwarding CustomerA
PE1(config-if)# ip address 10.0.0.17 255.255.255.252

PE1(config)# interface gigabitEthernet 0/2
PE1(config-if)# ip vrf forwarding CustomerB
PE1(config-if)# ip address 10.0.0.21 255.255.255.252
```

Configure eBGP towards Customers on the PE Routers

So far, we have configured eBGP on the customers' routers. However, we also need to define the BGP neighbors for the PE routers under address-family ipv4 vrf section, in order to establish the BGP adjacencies with the CE routers.

```
PE1(config)# router bgp 64501
PE1(config-router)# address-family ipv4 vrf CustomerA
PE1(config-router-af)# neighbor 10.0.0.2 remote-as 64401
PE1(config-router-af)# exit

PE1(config-router)# address-family ipv4 vrf CustomerB
PE1(config-router-af)# neighbor 10.0.0.6 remote-as 64301
PE2(config)# router bgp 64501
PE2(config-router)# address-family ipv4 vrf CustomerA
PE2(config-router-af)# neighbor 10.0.0.18 remote-as 64402
PE2(config-router-af)# exit

PE2(config-router)# address-family ipv4 vrf CustomerB
PE2(config-router-af)# neighbor 10.0.0.22 remote-as 64302
```

Inspecting the Forwarding Plane

Picture 2 depicts the captured traffic on the link between the PE1 and P routers, while pinging from PC1A to PC2B. The outer MPLS label Switching Path (LSP) is 18 and is used for label switching. It is learned via the LDP (Label Distribution Protocol) and has a local significance.

No	Tin	Source	Destination	Protoc	Len	Info
4	172.16.1.1	172.16.2.1	ICMP	106	Echo (ping) request	id=0xda05, seq=15/3840, ttl=62 (reply in 5897)
4	172.16.2.1	172.16.1.1	ICMP	102	Echo (ping) reply	id=0xda05, seq=15/3840, ttl=62 (request in 5896)

> Frame 5896: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
 > Ethernet II, Src: 52:3f:8f:5f:37:03 (52:3f:8f:5f:37:03), Dst: 52:3f:8f:d5:8f:03 (52:3f:8f:d5:8f:03)
 > MultiProtocol Label Switching Header, Label: 18, Exp: 0, S: 0, TTL: 62
 > MultiProtocol Label Switching Header, Label: 21, Exp: 0, S: 1, TTL: 62
 > Internet Protocol Version 4, Src: 172.16.1.1, Dst: 172.16.2.1
 > Internet Control Message Protocol

Picture 2: Captured Traffic Between PE1 and P Routers

MPLS forwarding table of PE1 is depicted in Picture 3.

```
PE1#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing Interface	Next Hop
16	Pop Label	10.0.0.12/30	0	Gi0/3	10.0.0.10
17	Pop Label	10.1.1.2/32	0	Gi0/3	10.0.0.10
18	18	10.1.1.3/32	0	Gi0/3	10.0.0.10
21	No Label	172.16.1.0/24[V]	1862	Gi0/1	10.0.0.2
22	No Label	172.16.1.0/24[V]	0	Gi0/2	10.0.0.6

```
PE1#
```

Picture 3: MPLS Forwarding Table of PE1 Router

The label 21 is the inner (VPN) label, added by the PE1 router. It is used to identify the correct next-hop (10.0.0.18) on the PE2 router for Customer A data traffic. The inner label is kept untouched by the P router. Only the PE routers perform either push or pop of the VPN labels. The VPN label for Customer B traffic is 22.

The P router is a transit router that performs pop of LSP labels 18 and 19 (Picture 4). This router takes the forwarding decision solely based on labels. The label 19 is the LSP label pushed on packet by PE2 router when sending traffic to 10.1.1.1.

```
P#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing Interface	Next Hop
18	Pop Label	10.1.1.3/32	97346	Gi0/4	10.0.0.13
19	Pop Label	10.1.1.1/32	104734	Gi0/3	10.0.0.9

```
P#
```

Picture 4: MPLS Forwarding Table of P Router

Picture 5 depicts the captured traffic on the link between P and PE2 routers, while issuing the ping command from

PC1A to PC2B. There is only one MPLS header with VPN label 21 because the P router has popped the label 18. Router PE2 removes the inner VPN header 21 and forwards ICMP request as a plain IP packet to CE2A (10.0.0.18).

No.	Time	Source	Destination	Protocol	Length	Info
8	58	172.16.1.1	172.16.2.1	ICMP	102	Echo (ping) request id=0xda05, seq=11/2816, ttl=62 (reply in 8323)
8	58	172.16.2.1	172.16.1.1	ICMP	106	Echo (ping) reply id=0xda05, seq=11/2816, ttl=62 (request in 8322)

```

> Frame 8322: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
> Ethernet II, Src: 52:3f:8f:d5:8f:04 (52:3f:8f:d5:8f:04), Dst: 52:3f:8f:c9:52:04 (52:3f:8f:c9:52:04)
> MultiProtocol Label Switching Header, Label: 21, Exp: 0, S: 1, TTL: 61
> Internet Protocol Version 4, Src: 172.16.1.1, Dst: 172.16.2.1
> Internet Control Message Protocol

```

Picture 5: Captured Traffic Between P and PE2 Routers

In the opposite direction, a packet carrying ICMP echo reply message from PC2A to PC1A contains the LSP label in the MPLS header. The VPN label is the same as in echo request (21) because both sides are customer A. Picture 6 depicts MPLS forwarding table of PE2 router.

```
PE2#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
16	Pop Label	10.0.0.8/30	0	Gi0/4	10.0.0.14
17	Pop Label	10.1.1.2/32	0	Gi0/4	10.0.0.14
18	19	10.1.1.1/32	0	Gi0/4	10.0.0.14
21	No Label	172.16.2.0/24[V]	1568	Gi0/1	10.0.0.18
22	No Label	172.16.2.0/24[V]	0	Gi0/2	10.0.0.22

```
PE2#
```

Picture 6: MPLS Forwarding Table of PE2 Router

Picture 7 depicts a forwarding table of the PE2 router for VRF Customer A. It contains two routes learned via BGP. It is the route 172.16.2.0/24 announced by customer router CE2A and the route 172.16.1.0 advertised by the router PE1.

```

PE2#show ip route vrf CustomerA | b Gateway
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.16/30 is directly connected, GigabitEthernet0/1
L       10.0.0.17/32 is directly connected, GigabitEthernet0/1
       172.16.0.0/24 is subnetted, 2 subnets
B       172.16.1.0 [200/0] via 10.1.1.1, 01:10:44
B       172.16.2.0 [200/0] via 10.0.0.18, 01:10:44
PE2#

```

Picture 7: VRF of Customer A on PE2 Router

Inspecting Control Plane

BGP Update message sent from PE1 to PE2 is depicted in Picture 8. Notice, that there is only one MPLS header with LSP label 18, VPN label is missing. It ensures that MP-BGP message is sent via the MPLS network. VPN label is distributed inside the MP-BGP update message along with the unique VPN-IPv4 prefix.

No	Tin	Source	Destination	Protocol	Len	Info
4	10.1.1.1	10.1.1.3	BGP	279	UPDATE Message, UPDATE Message, UPDATE Message	
4	10.1.1.1	10.1.1.3	BGP	81	UPDATE Message	

```
> Frame 5325: 279 bytes on wire (2232 bits), 279 bytes captured (2232 bits) on interface 0
> Ethernet II, Src: 52:3f:8f:5f:37:03 (52:3f:8f:5f:37:03), Dst: 52:3f:8f:d5:8f:03 (52:3f:8f:d5:8f:03)
> MultiProtocol Label Switching Header, Label: 18, Exp: 6, S: 1, TTL: 255
> Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.3
> Transmission Control Protocol, Src Port: 179, Dst Port: 20139, Seq: 104, Ack: 85, Len: 221
> Border Gateway Protocol - UPDATE Message
> Border Gateway Protocol - UPDATE Message
> Border Gateway Protocol - UPDATE Message
```

Picture 8: BGP Update Message with LSP label 18

VPN-IPv4 route is a customer's route that is modified to be unique in order to use the same private IP address for customers. VPN-IPv4 routes consists of the Route Distinguisher (RD) and the prefix. Picture 9 shows the content of the NLRI inside the MP_REACH_NLRI path attribute. It is the prefix 172.16.1.0 with the RD 64501:2 and the label stack (VPN label) 22 (Customer B).

No	Tin	Source	Destination	Proto	Len	Info
4...	10.1.1.1	10.1.1.3	BGP	279	UPDATE Message, UPDATE Message, UPDATE Message	
4...	10.1.1.1	10.1.1.3	BGP	81	UPDATE Message	

```

> MultiProtocol Label Switching Header, Label: 18, Exp: 6, S: 1, TTL: 255
> Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.3
> Transmission Control Protocol, Src Port: 179, Dst Port: 20139, Seq: 104, Ack: 85, Len: 221
√ Border Gateway Protocol - UPDATE Message
  - Marker: ffffffffffffffffffffffffffffffff
  - Length: 96
  - Type: UPDATE Message (2)
  - Withdrawn Routes Length: 0
  - Total Path Attribute Length: 73
  √ Path attributes
    √ Path Attribute - MP_REACH_NLRI
      > Flags: 0x80, Optional: Optional, Non-transitive, Complete
      - Type Code: MP_REACH_NLRI (14)
      - Length: 32
      - Address family identifier (AFI): IPv4 (1)
      - Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
      > Next hop network address (12 bytes)
      - Number of Subnetwork points of attachment (SNPA): 0
      √ Network layer reachability information (15 bytes)
        √ BGP Prefix
          - Prefix Length: 112
          - Label Stack: 22 (bottom)
          - Route Distinguisher: 64501:2
          - MP Reach NLRI IPv4 prefix: 172.16.1.0
    
```

Picture 9: Unique VPN-IPv4 Route

The BGP update message also contains the Path attribute – EXTENDED_COMMUNITIES where the route-target 64501:2 is located. It is shown in Picture 10.

No	Tin	Source	Destination	Proto	Len	Info
4...	10.1.1.1	10.1.1.3	BGP	279	UPDATE Message, UPDATE Message, UPDATE Message	
4...	10.1.1.1	10.1.1.3	BGP	81	UPDATE Message	

```

- Marker: ffffffffffffffffffffffffffffffff
- Length: 96
- Type: UPDATE Message (2)
- Withdrawn Routes Length: 0
- Total Path Attribute Length: 73
√ Path attributes
  > Path Attribute - MP_REACH_NLRI
  > Path Attribute - ORIGIN: IGP
  > Path Attribute - AS_PATH: 64301
  > Path Attribute - MULTI_EXIT_DISC: 0
  > Path Attribute - LOCAL_PREF: 100
  √ Path Attribute - EXTENDED_COMMUNITIES
    > Flags: 0xc0, Optional, Transitive: Optional, Transitive, Complete
    - Type Code: EXTENDED_COMMUNITIES (16)
    - Length: 8
    √ Carried extended communities: (1 community)
      √ Community Transitive Two-Octet AS Route Target: 64501:2
        - Community type high: Transitive Two-Octet AS (0x00)
        - Subtype as2: Route Target (0x02)
        - Two octets AS specific: 64501
        - Four octets AN specific: 2

```

Picture 10: Route Target Inside Extended Community

Conclusion:

We have provided the exact configuration steps that can help our readers create a BGP/MPLS L3 VPNs and grasp the overall concept. If you need to acquire more theoretical knowledge about the BGP/MPLS VPNs concept, read our first blog post (<https://www.noction.com/blog/bgp-mpls-layer3-vpn>).

Title: MPLS Configuration Tutorial Example [Step by Step Guide with VIDEO]

Content courtesy of: <https://www.rogerperkin.co.uk/ccie/mpls/cisco-mpls-tutorial/>

If you are looking for an MPLS Tutorial or step by step mpls configuration examples, this basic **MPLS VPN configuration example** will guide you from configuring the first router to a 3 router MPLS core with 2 external sites.

If you are looking for an explanation of MPLS then I would advise you read the "What is MPLS (<https://www.networkworld.com/article/2297171/sd-wan/network-security-mpls-explained.html>)" post first



before attempting this lab.

Cisco MPLS Configuration Video

The entire tutorial is covered in this video above so if you like to just watch the video is there, if you want to follow along I suggest you open this page twice or print it out so you can make notes.

Building the simple MPLS topology below this will consist of a 3 router MPLS core and two remote sites in the same VRF running OSPF as the PE-CE routing protocol. This will be quite a long post as I will be taking you through every single verification along the way to ensure you understand how each section works.


I will be using GNS3 (<https://www.gns3.com/>) and configuring the routers as we go so you can follow along.

**Learn about my my journey to becoming
CCIE #50038**

**Save yourself time, money and effort for
only £7.99**

*All my CCIE
Lab Study Tips
for only*

£7.99



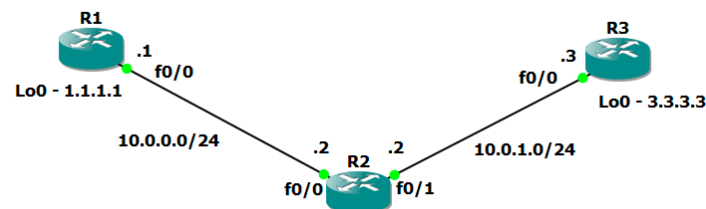
(<https://www.rogerperkin.co.uk/ccie-study-plan/>)

I have also started to get into Network Automation – if this is more your thing check out my pages on Ansible for Network Engineers (<https://www.rogerperkin.co.uk/network-automation/ansible/>)

Cisco MPLS Tutorial Topology

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

 0 10.0.0.2 [MPLS: Label 17 Exp 0] 84 msec 72 msec 44 msec
 1 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**

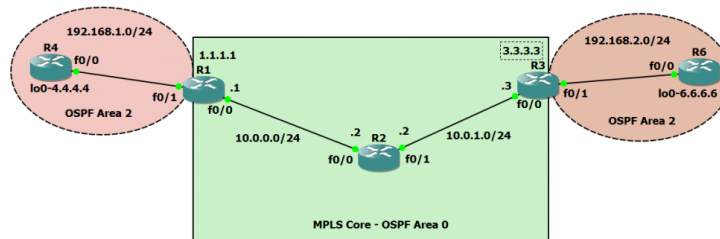
```
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

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 (<https://www.rogerperkin.co.uk/ccie/mps/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 int 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

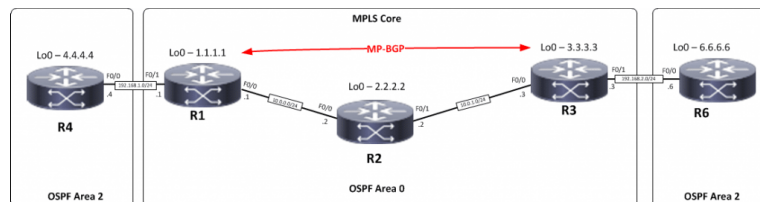
Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

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 ?
*> 6.6.6.6/32 3.3.3.3 11 100 0 ?
*> 192.168.1.0 0.0.0.0 0 32768 ?
*> 192.168.2.0 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 4.4.4.4 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
5.0.0.0/32 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

 0 192.168.1.1 20 msec 8 msec 8 msec
 1 10.0.0.2 [MPLS: Labels 17/20 Exp 0] 36 msec 40 msec 36 msec
 2 192.168.2.1 [MPLS: Label 20 Exp 0] 16 msec 40 msec 16 msec
 3 192.168.2.6 44 msec 40 msec 56 msec
R4#
```

More MPLS Configuration Resources

Want to learn more about MPLS? Check out these other great posts on [MPLS configuration and troubleshooting](#).

In the next MPLS Tutorial I will add a second customer site into the mix and also go through some MPLS Troubleshooting where I will go through turning off different features and trying to break the MPLS and show you the logical steps to troubleshoot it.

If you have enjoyed this tutorial, please take a minute to enter your name and email below to be kept up to date on new articles and upcoming training courses.

Roger

Title: MPLS explained | Network World

Content courtesy of: <https://www.networkworld.com/article/2297171/network-security-mpls-explained.html>

The thing about MPLS is that it's a technique, not a service — so it can deliver anything from IP VPNs to metro Ethernet. It's expensive, so with the advent of SD-WAN enterprises are trying to figure how to optimize its use vs. less expensive connections like the internet.

Did you ever order something online from a distant retailer and then track the package as it makes strange and seemingly illogical stops all over the country.

That's similar to the way IP routing on the Internet works. When an internet router receives an IP packet, that packet carries no information beyond a destination IP address. There is no instruction on how that packet should get to its destination or how it should be treated along the way.

Each router has to make an independent forwarding decision for each packet based solely on the packet's network-layer header. Thus, every time a packet arrives at a router, the router has to "think through" where to send the packet next. The router does this by referring to complex routing tables.

The process is repeated at each hop along the route until the packet eventually reaches its destination. All of those hops and all of those individual routing decisions result in poor performance for time-sensitive applications like video-conferencing or voice over IP (VoIP).

What is MPLS?

Multi-protocol label switching (MPLS), that venerable WAN workhorse launched at the turn of the century, addresses this problem by establishing pre-determined, highly efficient routes.

With MPLS, the first time a packet enters the network, it's assigned to a specific forwarding equivalence class (FEC), indicated by appending a short bit sequence (the label) to the packet.

Each router in the network has a table indicating how to handle packets of a specific FEC type, so once the packet has entered the network, routers don't need to perform header analysis. Instead, subsequent routers use the label as an index into a table that provides them with a new FEC for that packet.

This gives the MPLS network the ability to handle packets with particular characteristics (such as coming from particular ports or carrying traffic of particular application types) in a consistent fashion. Packets carrying real-time

traffic, such as voice or video, can easily be mapped to low-latency routes across the network — something that's challenging with conventional routing.

The key architectural point with all this is that the labels provide a way to attach additional information to each packet — information above and beyond what the routers previously had.

How does MPLS work?

The beauty of MPLS is that it's not tied to any underlying technology. It was designed back in the days of ATM and frame relay (<https://www.networkworld.com/article/2287582/lan-wan/frame-relay-vs-atm.html>) as an overlay technique designed to simplify and improve performance -- that's the "multi-protocol" part.

ATM and frame relay are distant memories, but MPLS lives on in carrier backbones and in enterprise networks. The most common use cases are branch offices, campus networks, metro Ethernet services and enterprises that need quality of service (QoS) for real-time applications.

Is MPLS Layer 2 or Layer 3?

There's been a lot of confusion about whether MPLS is a Layer 2 or Layer 3 service. But MPLS doesn't fit neatly into the OSI seven-layer hierarchy (<https://www.networkworld.com/article/3239677/lan-wan/the-osi-model-explained-how-to-understand-and-remember-the-7-layer-network-model.html>), and is sometimes classified as Layer 2.5. In fact, one of the key benefits of MPLS is that it separates forwarding mechanisms from the underlying data-link service. In other words, MPLS can be used to create forwarding tables for any underlying protocol.

Specifically, MPLS routers establish a label-switched path (LSP), a pre-determined path to route traffic in an MPLS network, based on the criteria in the FEC. It is only after an LSP has been established that MPLS forwarding can occur. LSPs are unidirectional which means that return traffic is sent over a different LSP.

When an end user sends traffic into the MPLS network, an MPLS label is added by an ingress MPLS router that sits on the network edge. The MPLS Label consists of four sub-parts:

The Label: The label holds all of the information for the MPLS routers to determine where the packet should be forwarded.

Experimental: Experimental bits are used for Quality of Service (QoS) to set the priority that the labeled packet should have.

Bottom-of-Stack: The Bottom-of-Stack tells the MPLS Router if it is the last leg of the journey and there are no more labels to be concerned with. This usually means the router is an egress router.

Time-To-Live: This identifies how many hops the packet can make before it is discarded.

MPLS Pros and Cons

The benefits of MPLS are scalability, performance, better bandwidth utilization, reduced network congestion and a better end-user experience.

MPLS itself does not provide encryption, but it is a virtual private network and, as such, is partitioned off from the public Internet. Therefore, MPLS is considered a secure transport mode. And it is not vulnerable to denial of service attacks, which might impact pure-IP-based networks.

On the negative side, MPLS is a service that must be purchased from a carrier and is far more expensive than sending traffic over the public Internet.

As companies expand into new markets, they may find it difficult to find an MPLS service provider who can deliver global coverage. Typically, service providers piece together global coverage through partnerships with other service providers, which can be costly.

And MPLS was designed in an era when branch offices sent traffic back to a main headquarters or data center, not for today's world where branch office workers want direct access to the cloud.

Is MPLS dead?

Gartner raised that provocative question back in 2013 and answered itself by predicting that MPLS would continue to be a fundamental part of the WAN landscape, but that most enterprises would slowly transition to a hybrid environment consisting of both MPLS networks and the public Internet.

MPLS will continue to have a role connecting specific point-to-point locations, like large regional offices, retail facilities with point of sale systems, regional manufacturing facilities, and multiple data centers. And it is required for real-time applications.

But enterprise WAN architects need to make a risk/reward calculation between the top-notch but expensive performance of MPLS vs. the cheaper but less reliable performance of the Internet. Which brings us to an exciting new technology called SD-WAN.

MPLS vs. SD-WAN

If you listen to the hype, cheap, flexible SD-WAN is going to wipe out MPLS, the slow-footed dinosaur. But, in fact, both technologies have a role to play in modern WANS.

SD-WAN is the application of Software Defined Networking (SDN) concepts to the WAN. This means the deployment of SD-WAN edge devices that apply rules and policies to send traffic along the best path.

SD-WAN is a transport-agnostic overlay that can route any type of traffic — including MPLS. The advantage of SD-WAN is that an enterprise WAN-traffic architect can sit at a central point and easily apply policies across all WAN devices.

By contrast, with MPLS predetermined routes need to be painstakingly provisioned and once the fixed circuits are up, making changes is not a point-and-click exercise.

But once an MPLS network is deployed, it delivers guaranteed performance for real-time traffic. SD-WAN can route traffic along the most efficient path, but once those IP packets hit the open Internet, there are no performance guarantees.

The most sensible strategy going forward will be to offload as much MPLS traffic as possible to the public Internet, but continue to use MPLS for time-sensitive applications that require guaranteed delivery. Nobody wants to get caught in the cross-hairs when the CEO's monthly videoconference with branch office employees drops off mid-sentence.

More about MPLS:

Join the Network World communities on

Facebook (<https://www.facebook.com/NetworkWorld/>)
and

LinkedIn (<https://www.linkedin.com/company/network-world>)
to comment on topics that are top of mind.

Title: Configuring the Customer Side of an MPLS VPN WAN, Part 1 - NetCraftsmen

Content courtesy of: <https://www.netcraftsmen.com/configuring-the-customer-side-of-an-mpls-vpn-wan-part-1/>

I've recently been doing a lot of consulting work involving variations on a common theme. The general theme is how do I configure my routers to work with the MPLS VPN service I have bought or am about to buy from Provider X. If the service is to be your sole WAN connection, there's little problem or complexity. Do get the carrier to hand routes off to you in your IGP, unless you happen to want EBGP in your life. If you are going to have two WAN connections, well, things can get more interesting. Which is what this article is about.

I've been noticing how the requests for MPLS WAN related consulting work has evolved over the years:

- Initially, folks discovered there is some "interesting" routing behavior when you tie an existing IPsec VPN or legacy WAN as alternative / backup path to a shiny new MPLS VPN.
- More recently, sites are doing two-carrier MPLS (i.e. diverse connections via two separate MPLS VPN clouds), often with MPLS support for QoS
- Layer 2 MPLS VPN has come onto the scene, often with QoS as well
- And some sites are now doing the legacy WAN + MPLS VPN approach, with the new MPLS VPN as a quick fix to support IP Telephony (IPT) or IP Video-conferencing (IPVC) and the older QoS-less and possibly congested WAN for data

I haven't noticed much on the web about how to configure for these sorts of situations. I just googled as a check, and found a couple of sites, plus links to prior writings of mine.

The lack of discussion is a bit surprising to me, since the routing can get a little more complicated than your average internal routing does. I've been aware of that since around 2003, when I started consulting on this. (And in presenting at a Cisco Powered Network, got the impression that MPLS Providers seemed to be all thinking that they'd be your only WAN connection, and weren't prepared for some of the issues discussed below.)

MPLS WAN Scenarios

I see this as three basic WAN settings or scenarios, each requiring mildly different approaches to configuration, and each entailing different important considerations. The settings are:

- Legacy WAN with IGP (EIGRP or OSPF) routing plus Layer 3 (routed) MPLS VPN WAN
- Dual-carrier L3 MPLS VPN WAN
- Legacy + Single, or dual-carrier L2 MPLS VPN

The last situation, differentially routing certain traffic over the MPLS WAN, with data over the legacy WAN, I see as QoS-related: how do I route traffic based on QoS needs or based on DSCP bit settings?

If you have IPT or IPVC or video, you probably do want an MPLS service supporting QoS via DSCP bits (not per-VLAN QoS, which I think is pretty useless). But that's Yet Another Topic, more properly a QoS topic, and not much different than regular QoS. You just have to match the carrier's DSCP bit preferences (if non-negotiable), etc., at the WAN edge.

This blog will cover the legacy + MPLS scenario, without or with the differential QoS over MPLS requirement. Others to be written will take a look at the other scenarios.

MPLS WAN: The Big Picture

If your area has fiber available (major league cities, etc.), you may find that Ethernet access to L2 or L3 MPLS VPN or to L2 VPLS or Ethernet Private Line (EPL) service is available and fairly cost effective, while providing flexibility as far as ability to rapidly add bandwidth as needed. In a recent set of preliminary quotes from vendors, we saw that there's a bit of a fee for the fiber access, but the incremental costs as you scale up bandwidth are low. If your network is in the T1 range and you're looking at going above that, the alternatives are pretty limited (IMA and ATM has hardware costs and generally only provides enough bandwidth for 1-2 years of use), so cost-effective Ethernet services with 1 or 5 Mbps increments is good news!

I'm starting to see more any more of a shift to the Ethernet-based access, which is simple and very useful. It does have the problem of "some sites left behind" — more rural sites may only be able to get T1 or costly T3 access still, while others are operating at 10, 100, or even 1 Gig speeds. This tends to result in apps that require more bandwidth and don't work so well at the slower sites.

The big picture here is that MPLS VPN is great, it can be very cost-effective. Both L3 and L2 MPLS VPN have quirks that make them a bit different than more traditional WAN designs. Nothing terrible, but factors that do have to be taken into account.

For L3 MPLS VPN, the challenge is that your Customer Edge (CE) router exchanges routes with the Provider Edge (PE) router, and that's just the way it works. Most providers seem to now grudgingly offer EIGRP, often telling sites you'll be the first customer on it, or back-hauled to a POP with a Cisco MPLS edge router at higher latency. That full disclosure is good and honest, but does seem to have the (intended?) effect of driving customers to NOT using EIGRP for the CE-PE routing.

L2 MPLS VPN is attractive to those who want to do their own routing. I'll note in passing that this is slightly illusory, in that routing in the MPLS core still underlies the L2 connectivity. On the other hand, the carrier should be doing some mix of SONET, MPLS Traffic Engineering (TE) Fast Re-route (FRR), fast core IP re-routing based on limited prefixes, tuned timers, etc. And it's all just up / down from your perspective. The bigger gotcha with L2 MPLS VPN or VPLS is that it makes the "WAN cloud" act like a giant Ethernet switch. Routing is not really intended for 25 or more routers on a LAN, let alone 100's. You can do that, but it's not a good design.

Legacy + MPLS

Let's get technical now, and talk about my first scenario, Legacy WAN plus Layer 3 MPLS VPN for WAN.

Generally the legacy WAN routers are talking EIGRP or OSPF to each other, an IGP (Interior Gateway Protocol). If your MPLS carrier provides EIGRP or OSPF handoff in the right form, that too will appear to be IGP routing and life is good (and simple to manage). Cisco has added features to carrier MPLS allowing the carrier MBGP to carry

EIGRP or OSPF routing metrics and re-advertise them back to your CE router as internal routes. There are minor quirks to consider and work with your carrier on if you have other connections ("backdoor connections") between your MPLS sites.

The main thing that goes wrong with the above is for the carrier to mess up on the EIGRP AS matching, at one or more sites. That has the effect of making the routes learned on the MPLS side external EIGRP routes. EIGRP much prefers internal routes (administrative distance). So if that happens, your routing will be steering traffic over the legacy WAN rather than the new MPLS WAN.

The other way routing can be designed is to talk EBGp to the carrier. (I'm ignoring static routes, which are useless for dynamic re-routing when you have two connections — and all my scenarios have two connections.) If your CE-PE routing on the MPLS WAN is EBGp, there are two sub-cases.

Sub-Case 1: If you only have one router (e.g. small remote office), then it's talking EIGRP or OSPF on one side, EBGp on the other. Due to administrative distance, EBGp wins, so the router will prefer the MPLS WAN for any prefix learned on that side. That's usually what you want, so lucky you, things work as you'd want!

Sub-Case 2: You have two routers, one (usually the older one) talking to the legacy WAN, the other (sometimes shiny new) talking to the new L3 MPLS WAN. EIGRP for legacy, EBGp for MPLS.

How do the two routers exchange routes? Generally people bi-directionally redistribute the EBGp into EIGRP or OSPF. (Some details follow below.) If you do that, the MPLS router has an IGP and EBGp as route sources, and EBGp wins. So if it receives traffic, it will generally prefer the MPLS WAN. That's good. On the other hand, the legacy router has internal IGP routes plus the external IGP routes learned from the MPLS router's redistribution. For an IGP, internal is better than external, so the legacy router "wants" to send traffic over the legacy WAN. Not so good. So will any other routers or L3 switches at the site. Definitely not so good.

This is why I like doing EIGRP or OSPF as the CE-PE protocol, matching your IGP. It keeps things simpler.

From a business perspective, I strongly recommend that the CE-PE routing protocol handoff be a major vendor selection criterion.

What can you do about this? If you have no other L3 devices at the site, you can make the MPLS router the HSRP / VRRP primary, with WAN interface tracking, or even whether an important route, like default, is successfully being learned from the MPLS peer. That way, devices' traffic will go to the MPLS router; it prefers the MPLS link, and life is good. If some site falls out of the MPLS cloud (so to speak), the prefix is only learned on the legacy side, so the MPLS router will redirect traffic to that site to the legacy router, which in turn will forward it out the legacy link.

Ah, but if you have other L3 devices at the site, then things are a bit more ... interesting. Some ideas:

- You can reconfigure the legacy WAN to use EBGp or IBGP. Painful, but it'll work. (Which to use, merits a separate discussion.)
- You can try altering administrative distances for external routes, carefully. That's a great way to create a routing loop — and didn't quite do what's necessary, the last time I tried it with EIGRP in the lab. (It appears like it should be possible with newer code, if configured on all site routers — I just haven't tested it in the lab yet.)
- You can use a "neighbor ... distance" command, to bias the adjacent site router(s) to give high admin distance to routes from the legacy WAN router. Deeper into the site, it doesn't matter, as long as the routers adjacent to the legacy router "prefer to" send traffic towards the MPLS router. I haven't done this or tested this, but it seems a lot cleaner than changing the administrative distance for external routes at every site router.
- You can try doing different IGP processes for WAN and LAN, and redistributing between them, but that gets rather messy. Not recommended. Particularly as you'll need bi-directional redistribution filtering, which adds a bit more complexity. (Route tags are your friend, if you're doing this.)
- You can run "the other" IGP at the site, redistributing both the legacy WAN IGP and the EBGp into it. Thus if the legacy WAN uses EIGRP, use OSPF at the site. You'll probably be redistributing bi-directionally, unless you want to configure a static site summary and redistribute that for advertisement across the WAN. This gets just as ugly as the previous idea.
- Yet another approach (with distinct negatives) is to tunnel between CE routers on the MPLS side, allowing you to run an IGP over the tunnels (GRE or IPsec, generally). If you're running IPsec VPN between CE devices for extra security, then you're set — leverage that for routing.
- You can run EBGp or IBGP on your legacy WAN, but that means re-doing all your legacy WAN routing, which is painful.

Adding QoS Preferential Routing to the Mix

Actually, this case is simpler. If you only have one site router, you can just do metric offsets or Policy-Based Routing (PBR) on it. We've done this sort of thing for PBX to PBX IPT traffic in a non-MPLS setting, where you make the IGP metric look better on the side with QoS for traffic headed to a PBX subnet (ATM in one case). More recently, I'm looking at settings with IPT using Microsoft Office Communicator and soft-phones, or soft IPVC clients. In that case, I lean more towards dealing with such device by somehow marking at the edge or trusting DSCP markings. And then you can do DSCP-based PBR on the site legacy router, to "deflect" such traffic to the MPLS router. If you track objects (MPLS neighbor and perhaps a route), you can not deflect traffic via PBR if the MPLS side is down. (Admittedly, a topic worthy of more discussion in and of itself.)

If you have two routers at the site, things get more complex. If you only want selected traffic using the MPLS side, then don't redistribute the EBGp into the site IGP on the MPLS router. Do still run the IGP on the MPLS router. If you have default routing advertised on the legacy side, then traffic (one way or another) will go to the legacy router, and you can use PBR as above to "deflect" certain traffic to the MPLS router. If the router doesn't have a route to the destination but the legacy router does, the MPLS router will send it back. You can avoid a routing loop if the preferred path back is different than the one you're doing PBR on. A dedicated router to router link rather than one via the site-interface is one way to do that. You may have to use point-to-point interfaces to the L3 site switch(es), rather than a VLAN, to avoid having the two routers peer with each other over a VLAN they also communicate with the rest of the site or site switch(es) over. Tracking the MPLS router can disable the PBR if the MPLS router goes down.

Note: Loss of default over the legacy WAN will mess this up. You could have the legacy WAN router advertise default for the site, being careful to NOT advertise default into the legacy WAN. An alternative is to have the MPLS router redistribute prefixes into the IGP. Since they're external, the legacy router will be preferred unless a prefix stops being advertised on the legacy side. In that case, all traffic for that site will start being routed to the MPLS side. You might want an access list to prevent data traffic from using MPLS. Or a QoS policy that protects your IPT / IPVC, and gives remaining bandwidth for limited data use.

Sites Becoming Transit

We haven't really talked about this yet, but sites tend to redistribute the site IGP into EBGp to get the site prefixes to

be known across the MPLS WAN. The alternative is to configure a static site summary route and carefully redistribute that and only that static route into the EBGp.

The site IGP redistribution approach amounts to bi-directional redistribution. This can be a recipe for routing loops, and is generally best done with some distribute lists and filtering. The problem becomes, configuring an access list or prefix list to only advertise out the site prefixes is messy, making the static summary redistribution above look simpler. In general, I don't like having to do site-specific filters, it creates extra operational work and can be a source of problems.

When you have two WAN connections, conceivably an outage might lead to a remote site (probably with limited access bandwidth) becoming transit, as in traffic going via one WAN to the site to get to a site on the other WAN. Routing metrics generally should take care of this. If you're going IGP into MPLS redistribution with too attractive a metric, the routing metrics may not work in your favor. This is the common problem when doing bi-directional redistribution.

One form of insurance against this is to tag routes being learned at a site or redistributed into the IGP. You can then use the EIGRP or OSPF tags to prevent such remote routes from being advertised back into the WAN, so that only the site prefixes are advertised. This does take consistent configuration. The good news is that the route maps and distribute lists needed can be made site-neutral, i.e. part of a cookie-cutter configuration template.

Coming Topics

A lot of the above discussion carries over the the other scenarios above, with some note-worthy differences in how it affects you. We'll save specific discussion for the follow-on blog(s) about this.

References

Ivan Pepelnjak has a great site with lots of information, and I have huge respect for the depth of his knowledge of routing. He's got a number of good and relevant articles, see for instance <http://blog.ioshints.info/2008/12/basic-mpls-vpn-design-configuration.html> (<http://blog.ioshints.info/2008/12/basic-mpls-vpn-design-configuration.html>). I come at this a little differently, with less focus on the MPLS provider side, and more on simplicity for the customer side (to the extent possible). I like the validation of seeing some of his discussions of customer side factors (MPLS security, VPLS service, etc.) — although I slightly disagree with the emphasis of the MPLS security article. MPLS Security takes a little more work by the provider than FR or ATM perhaps — but should be attainable. (I worry about sites unnecessarily doing IPsec with all its burdens, somewhat defeating the value of high speed MPLS service.) I love the title "VPLS is not Aspirin".

Cisco has a good posting about buying MPLS services, the Enterprise Consumer Guide, at http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/L3VPNCon.html (http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/L3VPNCon.html). It's a good source of questions to ask vendors about in an RFP, although I recently thought of a number of things I wanted to ask that might not have been in there. (And there was an amazing first in dealing with WAN provider discussions: we actually got some darn good answers, and I hereby award the prize for best technical answers in a customer document to Level 3.) There is a longer (more detailed?) version available as a book from Cisco Press. See <http://www.amazon.com/Selecting-MPLS-Services-Chris-Lewis/dp/1587051915> (http://www.amazon.com/Selecting-MPLS-Services-Chris-Lewis/dp/1587051915/ref=ntt_at_ep_dpi_2).



Peter Welcher

Architect, Operations Technical Advisor

A principal consultant with broad knowledge and experience in high-end routing and network design, as well as data centers, Pete has provided design advice and done assessments of a wide variety of networks. CCIE #1773, CCDP, CCSI (#94014)

Title: Creating an MPLS VPN - PacketLife.net

Content courtesy of: <https://packetlife.net/blog/2011/may/16/creating-mpls-vpn/>

Today we're going to look at the configuration required to create a basic MPLS VPN servicing two customers, each with a presence at two physical sites. If you're unfamiliar with the concepts of MPLS switching and VRFs on Cisco IOS, you may want to check out a few of my past articles before continuing:

Our lab topology looks like this:

[topology.png](#)

As a review, recall that

- P (provider) routers are ISP core routers which don't connect to customer routers and typically run only MPLS
- PE (provider edge) routers connect to customer sites and form the edge of a VPN
- CE (customer edge) routers exist at the edge of a customer site; they have no VPN awareness
- an IGP running among all P and PE routers is used to support LDP and BGP adjacencies within the provider network
- MP-BGP is run only among PE routers
- an IGP (typically) is run between each CE router and its upstream PE router

In our lab, OSPF is already in operation as the provider network IGP. OSPF processes have also been preconfigured on the CE routers; however, these OSPF topologies will remain separate from the provider OSPF.

There are five core tasks we need to accomplish to get an MPLS VPN up and running:

1. Enable MPLS on the provider backbone.
2. Create VRFs and assign routed interfaces to them.
3. Configure MP-BGP between the PE routers.
4. Configure OSPF between each PE router and its attached CE routers.
5. Enable route redistribution between the customer sites and the backbone.

Although plenty of CLI outputs are shown below, you may want to grab the finished router configurations (/media

/blog/attachments/586/MPLS_VPN_configs.zip) if you'd like to duplicate the lab on your own.

Enable MPLS

First we need to enable MPLS on all P-P and P-PE links with the `mpls ip` interface command. MPLS is *not* enabled on any CE-facing interfaces; CE routers do not run MPLS, just plain IP routing. LDP is enabled automatically as the default label distribution protocol (versus Cisco's legacy TDP). LDP typically runs between loopback addresses not directly reachable by LDP peers, which is why it's important to configure an IGP in the core before enabling MPLS.

We can verify the configuration of MPLS interfaces with `show mpls interfaces`.

```
P1(config)# interface f0/1
P1(config-if)# mpls ip
P1(config-if)# interface f1/0
P1(config-if)# mpls ip
P1(config-if)# do show mpls interfaces
Interface          IP          Tunnel    Operational
FastEthernet0/1    Yes (ldp)   No        Yes
FastEthernet1/0    Yes (ldp)   No        Yes
```

```
P2(config)# interface f0/1
P2(config-if)# mpls ip
P2(config-if)# interface f1/0
P2(config-if)# mpls ip
```

```
PE1(config)# interface f1/0
PE1(config-if)# mpls ip
```

```
PE2(config)# interface f1/0
PE2(config-if)# mpls ip
```

LDP adjacencies can be verified with the command `show mpls ldp neighbor`:

```
P1# show mpls ldp neighbor
Peer LDP Ident: 10.0.0.2:0; Local LDP Ident 10.0.0.1:0
TCP connection: 10.0.0.2.45114 - 10.0.0.1.646
State: Oper; Msgs sent/rcvd: 12/13; Downstream
Up time: 00:02:43
LDP discovery sources:
FastEthernet0/1, Src IP addr: 10.0.9.2
Addresses bound to peer LDP Ident:
10.0.9.2      10.0.9.9      10.0.0.2
Peer LDP Ident: 10.0.0.3:0; Local LDP Ident 10.0.0.1:0
TCP connection: 10.0.0.3.20327 - 10.0.0.1.646
State: Oper; Msgs sent/rcvd: 12/12; Downstream
Up time: 00:02:25
LDP discovery sources:
FastEthernet1/0, Src IP addr: 10.0.9.6
Addresses bound to peer LDP Ident:
10.0.9.6      10.0.0.3
```

Create and Assign VRFs

Our next step is to create customer VRFs on our PE routers and assign the customer-facing interfaces to them. We need to assign each VRF a route distinguisher (RD) to uniquely identify prefixes as belonging to that VRF and one or more route targets (RTs) to specify how routes should be imported to and exported from the VRF.

We'll use a route distinguisher for each VRF in the form of `<ASN>:<customer number>`. For simplicity, we'll reuse the same value as both an import and export route target within each VRF (though we are free to choose a different or additional route targets if we prefer). VRF configuration must be performed on both PE routers.

```
PE1(config)# ip vrf Customer_A
PE1(config-vrf)# rd 65000:1
PE1(config-vrf)# route-target both 65000:1
PE1(config-vrf)# ip vrf Customer_B
PE1(config-vrf)# rd 65000:2
PE1(config-vrf)# route-target both 65000:2
```

```
PE2(config)# ip vrf Customer_A
PE2(config-vrf)# rd 65000:1
PE2(config-vrf)# route-target both 65000:1
PE2(config-vrf)# ip vrf Customer_B
PE2(config-vrf)# rd 65000:2
PE2(config-vrf)# route-target both 65000:2
```

The command `route-target both` is used as a shortcut for the two commands `route-target import` and `route-target export`, which appear separately in the running configuration.

Now we need to assign the appropriate interfaces to each VRF and reapply their IP addresses. (Assigning an interface to a VRF automatically wipes it of any configured IP addresses. Your version of IOS may or may not inform you of this when it happens.) The command `show ip vrf interfaces` can be used to verify interface VRF assignment and addressing.

```

PE1(config)# interface f0/0
PE1(config-if)# ip vrf forwarding Customer_A
% Interface FastEthernet0/0 IP address 10.0.1.1 removed due to enabling VRF Customer_A
PE1(config-if)# ip address 10.0.1.1 255.255.255.252
PE1(config-if)# interface f0/1
PE1(config-if)# ip vrf forwarding Customer_B
% Interface FastEthernet0/1 IP address 10.0.1.5 removed due to enabling VRF Customer_B
PE1(config-if)# ip address 10.0.1.5 255.255.255.252
PE1(config-if)# ^Z
PE1# show ip vrf interfaces

```

Interface	IP-Address	VRF	Protocol
Fa0/0	10.0.1.1	Customer_A	up
Fa0/1	10.0.1.5	Customer_B	up

```

PE2(config)# interface f0/0
PE2(config-if)# ip vrf forwarding Customer_A
% Interface FastEthernet0/0 IP address 10.0.2.1 removed due to enabling VRF Customer_A
PE2(config-if)# ip address 10.0.2.1 255.255.255.252
PE2(config-if)# interface f0/1
PE2(config-if)# ip vrf forwarding Customer_B
% Interface FastEthernet0/1 IP address 10.0.2.5 removed due to enabling VRF Customer_B
PE2(config-if)# ip address 10.0.2.5 255.255.255.252
PE2(config-if)# ^Z
PE2# show ip vrf interfaces

```

Interface	IP-Address	VRF	Protocol
Fa0/0	10.0.2.1	Customer_A	up
Fa0/1	10.0.2.5	Customer_B	up

Configure MP-BGP on the PE Routers

This is where things start to get interesting. In order to advertise VRF routes from one PE router to the other, we must configure multiprotocol BGP (MP-BGP). MP-BGP is a little different from legacy BGP in that it supports multiple *address families* (e.g. IPv4 and IPv6) over a common BGP adjacency. It also supports the advertisement of VPN routes, which are longer than normal routes due to the addition of a 64-bit route distinguisher (which we assigned under VRF configuration).

MP-BGP runs only on the PE routers: P routers rely entirely on the provider IGP and MPLS to forward traffic through the provider network, and CE routers have no knowledge of routes outside their own VRF.

Minimal MP-BGP configuration is pretty straightforward. Both PE routers exist in BGP AS 65000.

```

PE1(config)# router bgp 65000
PE1(config-router)# neighbor 10.0.0.4 remote-as 65000
PE1(config-router)# neighbor 10.0.0.4 update-source loopback 0
PE1(config-router)# address-family vpnv4
PE1(config-router-af)# neighbor 10.0.0.4 activate

```

```

PE2(config)# router bgp 65000
PE2(config-router)# neighbor 10.0.0.3 remote-as 65000
PE2(config-router)# neighbor 10.0.0.3 update-source loopback 0
PE2(config-router)# address-family vpnv4
PE2(config-router-af)# neighbor 10.0.0.3 activate

```

If we look at the running configuration of the BGP process on either PE router, we notice that a bit more configuration than we provided has appeared:

```

PE1# show running-config | section router bgp
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.4 remote-as 65000
 neighbor 10.0.0.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.0.0.4 activate
  neighbor 10.0.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Customer_B
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf Customer_A
  no synchronization
 exit-address-family

```

In addition to our VPNv4 address family, address families for the two customer VRFs have been created automatically. Also, support for extended community strings has been added to the VPNv4 neighbor configuration.

Verify that the MP-BGP adjacency between PE1 and PE2 was formed successfully with the command `show bgp vpnv4 unicast all summary`:

```

PE1# show bgp vpnv4 unicast all summary
BGP router identifier 10.0.0.3, local AS number 65000
BGP table version is 1, main routing table version 1

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.0.0.4	4	65000	12	12	1	0	0	00:06:05	0

Currently, there are no routes in the BGP table, because we have not specified anything to be advertised or redistributed, but we'll get to that after this next step.

Configure PE-CE OSPF

We just configured MP-BGP between the two PE routers. Now, let's configure an IGP between each PE router and its attached CE routers to exchange routes with the customer sites. We're going to use OSPF for this lab, but we could just as easily use another IGP like EIGRP or RIP.

Single-area OSPF has already been configured on the CE routers; all CE interfaces are in area 0. Remember that although we're using OSPF between each of the CE routers and its upstream PE router, these OSPF processes are isolated from the provider OSPF topology. The overall routing topology will look like this:

routing_topology.png

The provider OSPF process has already been configured on the PE routers as process 1. We'll configure an *additional* OSPF process for each CE router on each PE router. Each PE router will then have three OSPF processes total: one for the provider network, and one for each CE router. Whereas the provider OSPF process exists in the global routing table, the two CE processes will each be assigned to their respective customer VRFs.

```
PE1 (config) # router ospf 2 vrf Customer_A
PE1 (config-router) # router-id 10.0.1.1
PE1 (config-router) # interface f0/0
PE1 (config-if) # ip ospf 2 area 0
PE1 (config-if) # router ospf 3 vrf Customer_B
PE1 (config-router) # router-id 10.0.1.5
PE1 (config-router) # interface f0/1
PE1 (config-if) # ip ospf 3 area 0
```

```
PE2 (config) # router ospf 2 vrf Customer_A
PE2 (config-router) # router-id 10.0.2.1
PE2 (config-router) # interface f0/0
PE2 (config-if) # ip ospf 2 area 0
PE2 (config-if) # router ospf 3 vrf Customer_B
PE2 (config-router) # router-id 10.0.2.5
PE2 (config-router) # interface f0/1
PE2 (config-if) # ip ospf 3 area 0
```

We should see each PE router form an OSPF adjacency with both of its attached CE routers, and the customer routes should appear in the VRF tables on the PE routers.

```
PE1# show ip route vrf Customer_A

Routing Table: Customer_A
...

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       172.16.1.0/24 [110/11] via 10.0.1.2, 00:04:21, FastEthernet0/0
O       172.16.0.1/32 [110/11] via 10.0.1.2, 00:04:21, FastEthernet0/0
       10.0.0.0/30 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, FastEthernet0/0
PE1# show ip route vrf Customer_B

Routing Table: Customer_B
...

172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       172.17.1.0/24 [110/11] via 10.0.1.6, 00:03:03, FastEthernet0/1
O       172.17.0.1/32 [110/11] via 10.0.1.6, 00:03:04, FastEthernet0/1
       10.0.0.0/30 is subnetted, 1 subnets
C       10.0.1.4 is directly connected, FastEthernet0/1
```

Configure Route Redistribution

We're almost done! We have our MPLS and MP-BGP backbone up and running, and our CE routers are sending routes to our PE routers within their VRFs. The last step is to glue everything together by turning on route redistribution from the customer-side OSPF processes into MP-BGP and vice versa on the PE routers.

First we'll configure redistribution of CE routes in each VRF into MP-BGP. This is done under the BGP IPv4 address family for each VRF.

```
PE1 (config) # router bgp 65000
PE1 (config-router) # address-family ipv4 vrf Customer_A
PE1 (config-router-af) # redistribute ospf 2
PE1 (config-router-af) # address-family ipv4 vrf Customer_B
PE1 (config-router-af) # redistribute ospf 3
```

```
PE2 (config) # router bgp 65000
PE2 (config-router) # address-family ipv4 vrf Customer_A
PE2 (config-router-af) # redistribute ospf 2
PE2 (config-router-af) # address-family ipv4 vrf Customer_B
PE2 (config-router-af) # redistribute ospf 3
```

This enables redistribution of OSPF routes into BGP for transport across the provider network between the two sites. We can verify that the routes learned from the customer sites (the 172.16.0.0/16 and 172.17.0.0/16 networks) now appear in the BGP tables for their respective VRFs.

```

PE1# show ip bgp vpnv4 vrf Customer_A
...

Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65000:1 (default for vrf Customer_A)
*> 10.0.1.0/30      0.0.0.0              0           32768 ?
*>i10.0.2.0/30      10.0.0.4              0    100     0 ?
*> 172.16.0.1/32    10.0.1.2              11          32768 ?
*>i172.16.0.2/32    10.0.0.4              11    100     0 ?
*> 172.16.1.0/24    10.0.1.2              11          32768 ?
*>i172.16.2.0/24    10.0.0.4              11    100     0 ?

PE1# show ip bgp vpnv4 vrf Customer_B
...

Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65000:2 (default for vrf Customer_B)
*> 10.0.1.4/30      0.0.0.0              0           32768 ?
*>i10.0.2.4/30      10.0.0.4              0    100     0 ?
*> 172.17.0.1/32    10.0.1.6              11          32768 ?
*>i172.17.0.2/32    10.0.0.4              11    100     0 ?
*> 172.17.1.0/24    10.0.1.6              11          32768 ?
*>i172.17.2.0/24    10.0.0.4              11    100     0 ?

```

The last step is to complete the redistribution in the opposite direction: from BGP into the customer OSPF processes. If you're accustomed to route redistribution, there's nothing new here. (We don't have to specify any VRF information in the redistribution statement because each customer OSPF process is already assigned to a VRF.)

```

PE1(config)# router ospf 2
PE1(config-router)# redistribute bgp 65000 subnets
PE1(config-router)# router ospf 3
PE1(config-router)# redistribute bgp 65000 subnets

```

```

PE2(config)# router ospf 2
PE2(config-router)# redistribute bgp 65000 subnets
PE2(config-router)# router ospf 3
PE2(config-router)# redistribute bgp 65000 subnets

```

Testing and Confirmation

If has gone well, we should now have end-to-end connectivity between the CE routers within each VRF. Both routers for each customer should now have complete routing tables. Here are customer A's routes:

```

CE1A# show ip route
...

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.1.0/24 is directly connected, Loopback1
C    172.16.0.1/32 is directly connected, Loopback0
O IA  172.16.2.0/24 [110/21] via 10.0.1.1, 00:03:50, FastEthernet0/0
O IA  172.16.0.2/32 [110/21] via 10.0.1.1, 00:03:50, FastEthernet0/0
     10.0.0.0/30 is subnetted, 2 subnets
O IA  10.0.2.0 [110/11] via 10.0.1.1, 00:03:50, FastEthernet0/0
C    10.0.1.0 is directly connected, FastEthernet0/0

```

```

CE2A# show ip route
...

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
O IA  172.16.1.0/24 [110/21] via 10.0.2.1, 00:02:49, FastEthernet0/0
O IA  172.16.0.1/32 [110/21] via 10.0.2.1, 00:02:49, FastEthernet0/0
C    172.16.2.0/24 is directly connected, Loopback1
C    172.16.0.2/32 is directly connected, Loopback0
     10.0.0.0/30 is subnetted, 2 subnets
C    10.0.2.0 is directly connected, FastEthernet0/0
O IA  10.0.1.0 [110/11] via 10.0.2.1, 00:02:49, FastEthernet0/0

```

You may notice that OSPF routes sent between two sites belonging to the same customer appear as inter-area routes. Remember that although OSPF area 0 is being used at both sites, each site exists as a separate link-state topology connected by the MPLS VPN.

We should be able to ping from one CE router to the other. (Remember that we don't need to specify a VRF when doing so because CE routers have no knowledge that they're in a VRF.)

```

CE1A# ping 172.16.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/21/32 ms

```

We can perform a traceroute to verify the path taken as well as the MPLS labels used to traverse the provider network.

```
CE1A# traceroute 172.16.0.2

Type escape sequence to abort.
Tracing the route to 172.16.0.2

 0 10.0.1.1 4 msec 4 msec 8 msec
 1 10.0.9.5 [MPLS: Labels 19/22 Exp 0] 16 msec 12 msec 24 msec
 2 10.0.9.2 [MPLS: Labels 19/22 Exp 0] 24 msec 20 msec 16 msec
 3 10.0.2.1 [MPLS: Label 22 Exp 0] 20 msec 16 msec 24 msec
 4 10.0.2.2 16 msec * 36 msec
```

Here's a packet capture (/media/captures/traceroute_MPLS.cap) of the above traceroute if you're interested in how the MPLS label information is returned. And again, here are the finished router configurations (/media/blog/attachments/586/MPLS_VPN_configs.zip) if you'd like to replicate the lab yourself.

(Thanks to Ivan Pepelnjak (<http://twitter.com/#!/ioshints>) of Cisco IOS Hints (<http://blog.ioshints.info/>) helping revise this article!)

Title: MPLS Layer 3 VPN Configuration | NetworkLessons.com

Content courtesy of: <https://networklessons.com/mpls/mpls-layer-3-vpn-configuration>

In this lesson we'll take a look how to configure a MPLS Layer 3 VPN PE-CE scenario. Here's the topology I will use:

MPLS L3 VPN PE CE

Above we have five routers where AS 234 is the service provider. There's one customer with two sites, AS 1 and AS 5. Our customer wants to exchange 1.1.1.1 /32 and 5.5.5.5 /32 between its sites using BGP. To achieve this, we'll have to do a couple of things:

- Configure IGP and LDP within the service provider network.
- Configure VRFs on the PE routers.
- Configure IBGP between the PE routers.
- Configure BGP between the PE and CE routers.

There are a lot of difference pieces in the MPLS puzzle to make this work. Instead of configuring everything at once and praying that it will work, we'll build this network step-by-step. At each step, I'll show you how to verify that it's working before we continue with the next step.

Having said that, let's get started!

Configuration

IGP and LDP

First we will configure the service provider network. On the PE1, P and PE2 routers we will create a loopback interface that will be advertised in OSPF. LDP will then use the addresses as the transport address for the TCP connection. Let's add those interfaces and enable OSPF:

```
PE1 (config) #interface loopback 0
PE1 (config-if) #ip address 2.2.2.2 255.255.255.255
```

```
P (config) #interface loopback 0
P (config-if) #ip address 3.3.3.3 255.255.255.255
```

```
PE2 (config) #interface loopback 0
PE2 (config-if) #ip address 4.4.4.4 255.255.255.255
```

Now we will configure OSPF to advertise all interfaces in the service provider network:

```
PE1 (config) #router ospf 1
PE1 (config-router) #network 192.168.23.0 0.0.0.255 area 0
PE1 (config-router) #network 2.2.2.2 0.0.0.0 area 0
```

```
P (config) #router ospf 1
P (config-router) #network 192.168.23.0 0.0.0.255 area 0
P (config-router) #network 192.168.34.0 0.0.0.255 area 0
P (config-router) #network 3.3.3.3 0.0.0.0 area 0
```

```
PE2 (config) #router ospf 1
PE2 (config-router) #network 192.168.34.0 0.0.0.255 area 0
PE2 (config-router) #network 4.4.4.4 0.0.0.0 area 0
```

And let's enable LDP on all internal interfaces:

```
PE1 (config) #interface FastEthernet 0/1
PE1 (config-if) #mpls ip
```

```
P (config) #interface FastEthernet 0/0
P (config-if) #mpls ip

P (config) #interface FastEthernet 0/1
P (config-if) #mpls ip
```

```
PE2 (config) #interface FastEthernet 0/0
PE2 (config-if) #mpls ip
```


That takes care of that. Let's see if MPLS is enabled:

```
PE1#show mpls interfaces
Interface      IP          Tunnel  BGP Static Operational
FastEthernet0/1  Yes (ldp)   No      No  No      Yes
```

```
P#show mpls interfaces
Interface      IP          Tunnel  BGP Static Operational
FastEthernet0/0  Yes (ldp)   No      No  No      Yes
FastEthernet0/1  Yes (ldp)   No      No  No      Yes
```

```
PE2#show mpls interfaces
Interface      IP          Tunnel  BGP Static Operational
FastEthernet0/0  Yes (ldp)   No      No  No      Yes
```

That's looking good to me. Do we have any LDP neighbors?

```
P#show mpls ldp neighbor
Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 3.3.3.3:0
TCP connection: 2.2.2.2.646 - 3.3.3.3.55065
State: Oper; Msgs sent/rcvd: 10/11; Downstream
Up time: 00:02:39
LDP discovery sources:
  FastEthernet0/0, Src IP addr: 192.168.23.2
Addresses bound to peer LDP Ident:
  192.168.12.2  192.168.23.2  2.2.2.2
Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 3.3.3.3:0
TCP connection: 4.4.4.4.52817 - 3.3.3.3.646
State: Oper; Msgs sent/rcvd: 10/11; Downstream
Up time: 00:02:02
LDP discovery sources:
  FastEthernet0/1, Src IP addr: 192.168.34.4
Addresses bound to peer LDP Ident:
  192.168.34.4  192.168.45.4  4.4.4.4
```

Our P router in the middle has two neighbors so we know that LDP is working. Just to be sure, let's check if we have connectivity between PE1 and PE2:

```
PE1#ping 4.4.4.4 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

A quick ping tells us that it's working. Are we switching based on labels though? Let's do a trace to find out:

```
PE1#traceroute 4.4.4.4 source loopback 0
Type escape sequence to abort.
Tracing the route to 4.4.4.4
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.23.3 [MPLS: Label 17 Exp 0] 0 msec 0 msec 4 msec
 2 192.168.34.4 0 msec 0 msec *
```

Above you can see that we are using a label for the packet from PE1 to PE2. The P router is popping the label (penultimate hop popping) so PE1 receives a normal IP packet. So far, this is looking good.

VRF on the PE routers

Since we want our customer routes separated from the service provider's routes, we'll have to create some VRFs. Here's how it's done:

```
PE1 (config) #ip vrf CUSTOMER
```

First I will create a VRF called CUSTOMER. The next step will be configuring a RD (Route Distinguisher):

```
PE1 (config-vrf) #rd ?
ASN:nn or IP-address:nn VPN Route Distinguisher
```

The RD is to make sure that all prefixes are unique. The customer prefix + RD together are a VPNv4 route. I'll pick something simple:

```
PE1 (config-vrf) #rd 1:1
```

Our RD will be 1:1. The next item to configure is the RT (Route Target). This defines where we will import and export our VPNv4 routes. I want to make sure that all routes from CE1 and CE2 will be exchanged:

```
PE1 (config-vrf) #route-target both 1:1
```

I will use RT value 1:1 and use parameter both. This means that all routes of this VRF will be imported and exported.

I used the same value (1:1) for the RD and RT, keep in mind that these are two different things...don't mix them up!

Here's what the VRF now looks like:

```
PE1#show run | begin vrf
ip vrf CUSTOMER
  rd 1:1
  route-target export 1:1
  route-target import 1:1
```

After creating the VRF globally, we have to assign the interface that is facing the customer to the VRF:

```

PE1 (config) #interface FastEthernet 0/0
PE1 (config-if) #ip vrf forwarding CUSTOMER
% Interface FastEthernet0/0 IPv4 disabled and address(es) removed due to enabling VRF CUSTOMER

```

Once you add an interface to a VRF, Cisco IOS will remove its IP address. Let's add it again:

```

PE1 (config-if) #ip address 192.168.12.2 255.255.255.0

```

The VRF configuration of PE1 is now complete. We'll configure the exact same thing on PE2:

```

PE2 (config) #ip vrf CUSTOMER
PE2 (config-vrf) #rd 1:1
PE2 (config-vrf) #route-target export 1:1
PE2 (config-vrf) #route-target import 1:1

PE2 (config) #interface FastEthernet 0/1
PE2 (config-if) #ip vrf forwarding CUSTOMER
PE2 (config-if) #ip address 192.168.45.4 255.255.255.0

```

The VRFs are now configured. If you want to reach the CE1 or CE2 routers then you'll have to use the VRFs from now on:

```

PE1#ping vrf CUSTOMER 192.168.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

```

```

PE2#ping vrf CUSTOMER 192.168.45.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.45.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

```

Great our VRFs are operational!

IBGP Configuration on PE1 and PE2

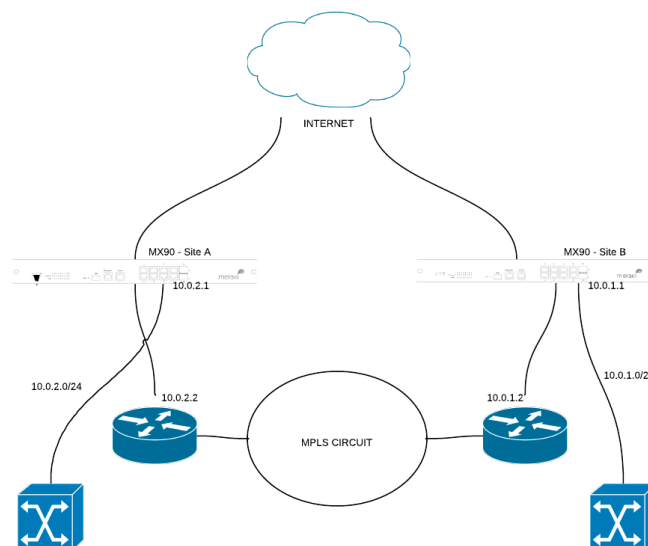
PE1 and PE2 will have to exchange VPNv4 routes through IBGP. When you configure iBGP, your routers will only exchange IPv4 unicast routes by default. Since we need the PE routers to exchange VPNv4 routes, we'll have to activate an additional address-family:

Title: Integrating an MPLS Connection on the MX LAN - Cisco Meraki

Content courtesy of: https://documentation.meraki.com/MX/Networks_and_Routing/Integrating_an_MPLS_Connection_on_the_MX_LAN

Customers with an MPLS connection between sites can use this article as a guide for allowing communication over the LAN when the MPLS connection is not intended for accessing the Internet. Alternatively, if the MPLS connection is the primary WAN link for the location and needs to be implemented with VPN failover, refer to the guide on configuring site-to-site VPN over MPLS (https://documentation.meraki.com/MX/Site-to-site_VPN/Configuring_Site-to-site_VPN_over_MPLS).

In the example below, two sites exist. Each site has an independent connection to the Internet and an MPLS circuit between the two sites. Site A has a local subnet of 10.0.2.0/24, and Site B has a local subnet of 10.0.1.0/24.



In order for both sites to communicate with each other, a static route must be configured on each MX for the subnet

of the remote site, pointing to the local MPLS router (connected to the MPLS CIRCUIT) as the next hop. The MPLS router, generally owned by the ISP, will then pass the traffic to the remote site. If a client at Site A wants to talk to a client at Site B, the traffic will be forwarded over the MPLS link.

The screenshot below shows the **Routing** section of the **Security & SD-WAN > Configure > Addressing & VLANs** page in Dashboard for Site B. The first entry describes the local network at Site B. The second entry describes the static route to reach Site A over the MPLS link.

Routing

Use VLANs ☐ (Disabled: Use Single LAN)

LAN Config

Subnet	Name	Link IP
10.0.1.0/24	single lan settings	10.0.1.1

Static routes

[Delete](#) [Add Static Route](#)

Enabled	Name	Subnet	Gateway IP	Conditions
<input checked="" type="checkbox"/>	Site A over MPLS	10.0.2.0/24	10.0.1.2	always

Title: MPLS Training | Implementing Cisco MPLS | Global Knowledge

Content courtesy of: <https://www.globalknowledge.com/us-en/course/91340/mpls-implementing-cisco-mpls-v30/>

This course includes Cisco Training Exclusives

EXCLUSIVE TO GLOBAL KNOWLEDGE - Accelerate your Cisco learning experience with complimentary access to the IT Skills Video On-Demand Library, *Introduction to Cybersecurity* digital learning course, course recordings, IT Resource Library, and digital courseware.

Learn more ([us-en/brands/cisco/cisco-training-exclusives/](/us-en/brands/cisco/cisco-training-exclusives/))

This course is designed to introduce you to MPLS concepts, installation, migration, operation, inspection, and troubleshooting. You'll start with an overview of MPLS and its operation, after which you'll concentrate on MPLS Virtual Private Network (VPN) deployment. The MPLS fundamentals covered in this class will provide the theory and hands-on knowledge to implement, integrate, and deploy an MPLS infrastructure. The MPLS VPN lecture and labs will cover the models, diversity, implementation, troubleshooting, and flexibility of MPLS VPNs.