

Multi-Protocol Label Switching (MPLS) Support

This chapter describes the system's support for BGP/MPLS VPN and explains how it is configured. The product administration guides provide examples and procedures for configuration of basic services on specific systems. It is recommended that you select the configuration example that best meets your service model and configure the required elements for that model, as described in the respective product administration guide, before using the procedures in this chapter.

When enabled through a feature license key, the system supports MPLS to provide a VPN connectivity from the system to the corporate's network.

Important This release provides BGP/MPLS VPN for directly connected PE routers only.

MP-BGP is used to negotiate the routes and segregate the traffic for the VPNs. The network node learns the VPN routes from the connected Provider Edge (PE), while the PE populates its routing table with the routes provided by the network functions.

- Overview, on page 1
- Supported Standards, on page 3
- Supported Networks and Platforms, on page 4
- Licenses, on page 4
- Benefits, on page 4
- Configuring BGP/MPLS VPN with Static Labels, on page 4
- Configuring BGP/MPLS VPN with Dynamic Labels, on page 7

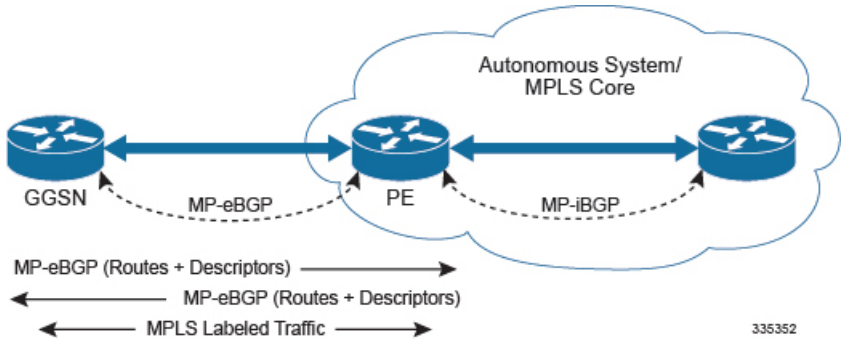
Overview

As seen in the following scenario, the chassis can be deployed as a router while supporting BGP/MPLS-VPN in a network.

- Chassis as MPLS-Customer Edge (MPLS-CE) connecting to Provider Edge (PE)
- Chassis as MPLS-Customer Edge (MPLS-CE) connecting to Autonomous System Border Router (ASBR)

Chassis as MPLS-CE Connecting to PE

Figure 1: Chassis as MPLS-CE Connected to PE



The system in this scenario uses static/dynamic MPLS labels for ingress and egress traffic. For configuration information on static label, refer to the Configuring BGP/MPLS VPN with Static Labels, on page 4 section and refer to Configuring BGP/MPLS VPN with Static Labels, on page 4 for dynamic label configuration.

The system is in a separate autonomous system (AS) from the Provider Edge (PE). It communicates with the PE and all VPN routes are exchanged over MP-BGP. Routes belonging to different VPNs are logically separated, using separate virtual route forwarding tables (VRFs).

Routes for each VPN are advertised as VPN-IPv4 routes, where route distinguishers are prepended to regular IPv4 routes to allow them to be unique within the routing table. Route targets added to the BGP extended community attributes identify different VPN address spaces. The particular upstream BGP peer routing domain (VPN), from which a route is to be imported by the downstream peer into an appropriate VRF, is identified with an extended community in the advertised NLRI.

A unique label is also received or advertised for every VPN route.

The Customer Edge (CE) also advertises routes to the PE using NLRIs that include route distinguishers to differentiate VPNs, an extended community to identify VRFs, and a MPLS-label, which will later be used to forward data traffic.

There is a single MPLS-capable link between the CE and the PE. MP-BGP communicates across this link as a TCP session over IP. Data packets are sent bidirectionally as MPLS encapsulated packets.

This solution does not use any MPLS protocols. The MPLS label corresponding to the immediate upstream neighbor can be statically configured on the downstream router, and similarly in the reverse direction.

When forwarding subscriber packets in the upstream direction to the PE, the CE encapsulates packets with MPLS headers that identify the upstream VRF (the label sent with the NLRI) and the immediate next hop. When the PE receives a packet it swaps the label and forward.

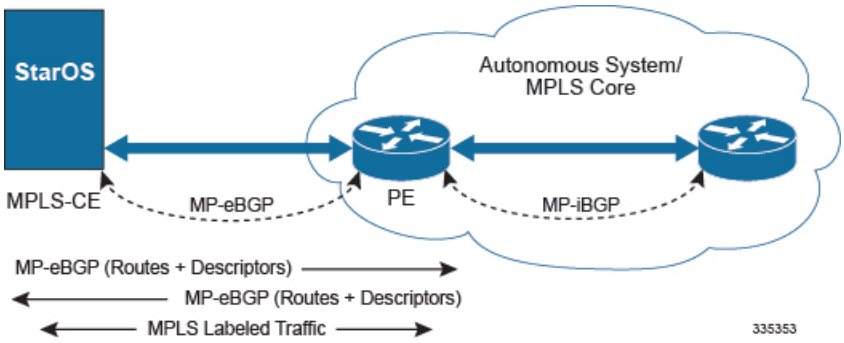
The CE does not run any MPLS protocol (LDP or RSVP-TE).

When receiving data packets in the downstream direction from the PE, the label is checked to identify the destination VRF. Then the packet is de-encapsulated into an IP packet and sent to the session subsystem for processing.

Important MPLS ping/trace route debugging facilities are not supported.

Chassis as MPLS-CE Connected to ASBR

Figure 2: Chassis as MPLS-CE Connected to ASBR



The system in this scenario uses static/dynamic MPLS labels for ingress and egress traffic. For configuration information on static label, refer to Configuring BGP/MPLS VPN with Static Labels, on page 4 and refer to Configuring BGP/MPLS VPN with Dynamic Labels, on page 7 for dynamic label configuration.

This scenario differs from the MPLS-CE with PE scenario in terms of peer functionality even though MPLS-CE functionality does not change. Like the MPLS-CE with PE scenario, MPLS-CE system maintains VRF routes in various VRFs and exchanges route information with peer over MP-eBGP session.

The peer in this scenario is not a PE router but an Autonomous System Border Router (ASBR). The ASBR does not need to maintain any VRF configuration. The PE routers use IBGP to redistribute labeled VPN-IPv4 routes either to an ASBR or to a route reflector (of which the ASBR is a client). The ASBR then uses the eBGP to redistribute those labeled VPN-IPv4 routes to an MPLS-CE in another AS. Because of the eBGP connection, the ASBR changes the next-hop and labels the routes learned from the iBGP peers before advertising to the MPLS-CE. The MPLS-CE is directly connected to the eBGP peering and uses only the MP-eBGP to advertise and learn routes. The MPLS-CE pushes/pops a single label to/from the ASBR, which is learned over the MP-eBGP connection. This scenario avoids the configuration of VRFs on the PE, which have already been configured on the MPLS-CE.

Engineering Rules

- Up to 5,000 "host routes" spread across multiple VRFs per BGP process. Limited to 6,000 pool routes per chassis.
- Up to 2,048 VRFs per chassis.

Supported Standards

Support for the following standards and requests for comments (RFCs) have been added with this interface support:

- RFC 4364, BGP/MPLS IP VPNs
- RFC 3032, MPLS Label Stack Encoding

Important One or more sections of above mentioned IETF are partially supported for this feature. For more information on Statement of Compliance, contact your Cisco account representative.

Supported Networks and Platforms

This feature supports all ASR5500 platforms with StarOS Release 9.0 or later running with network function services.

Licenses

Multi-protocol label switching (MPLS) is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

Benefits

MPLS provides networks with a more efficient way to manage applications and move information between locations. MPLS prioritizes network traffic, so administrators can specify which applications should move across the network ahead of others.

Configuring BGP/MPLS VPN with Static Labels

This section describes the procedures required to configure the system as an MPLS-CE to interact with a PE with static MPLS label support.

The base configuration, as described in the *Routing* chapter in this guide, must be completed prior to attempt the configuration procedure described below.

Important The feature described in this chapter is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements.

Important Commands used in the configuration samples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To configure the system for BGP/MPLS VPN:

Procedure

Step 1 Create a VRF on the router and assign a VRF name, define the Route-Distinguisher first before you execute the Route Target. For more information on the configuration see the Create VRF with Route-distinguisher and Route-target, on page 5 section.

Step 2 Set the neighbors and address family to exchange routing information and establish BGP peering with a peer router by applying the example configuration in Set Neighbors and Enable VPNv4 Route Exchange, on page 5.

Step 3 Configure the address family and redistribute the connected routes domains into BGP by applying the example configuration in Configure Address Family and Redistributed Connected Routes, on page 6. This takes any routes from another protocol and redistributes them to BGP neighbors using the BGP protocol.

Step 4 Configure IP Pools with MPLS labels for input and output by applying the example configuration in Configure IP Pools with MPLS Labels, on page 6.

Step 5 *Optional*. Bind DHCP service to work with MPLS labels for input and output in corporate networks by applying the example configuration in Bind DHCP Service for Corporate Servers, on page 6.

Step 6 *Optional*. Bind AAA/RADIUS server group in corporate network to work with MPLS labels for input and output by applying the example configuration in Bind AAA Group for Corporate Servers, on page 7.

Step 7 Save your configuration as described in the *System Administration Guide*.

Create VRF with Route-distinguisher and Route-target

Use this example to first create a VRF on the router and assign a VRF name. The second **ip vrf** command creates the route-distinguisher and route-target.

```
configure
   context <context_name> -noconfirm
      ip vrf <vrf_name>
         router bgp <as_number>
            ip vrf <vrf_name>
               route-distinguisher {<as_value> | <ip_address>} <rt_value>
               route-target export {<as_value> | <ip_address>} <rt_value>
               end
```

Set Neighbors and Enable VPNv4 Route Exchange

Use this example to set the neighbors and address family to exchange VPNv4 routing information with a peer router.

```
configure
   context <context_name>
      router bgp <as_number>
         neighbor <ip_address> remote-as <AS_num>
         address-family vpnv4
         neighbor <ip_address> activate
         neighbor <ip_address> send-community both
         exit
      interface <bind_intfc_name>
```

```
      ip address <ip_addr_mask_combo>
      end
```

Configure Address Family and Redistributed Connected Routes

Use this example to configure the **address-family** and to **redistribute** the connected routes or IP pools into BGP. This takes any routes from another protocol and redistributes them using the BGP protocol.

```
configure
   context <context_name>
      router bgp <as_number>
         address-family ipv4 <type> vrf <vrf_name>
            redistribute connected
            end
```



Note StarOS considers the below address range as Martian and hence rejects any BGP updates falling under this range.

Also, StarOS does not allow BGP network command to configure from the below ranges.

- 127.x.x.x
- 128.0.x.x
- 191.255.x.x
- 192.0.0.x
- 223.255.255.x

Configure IP Pools with MPLS Labels

Use this example to configure IP Pools with MPLS labels for input and output.

```
configure
   context <context_name> -noconfirm
      ip pool <name> <ip_addr_mask_combo> private vrf <vrf_name> mpls-label input
 <in_label_value> output <out_label_value1> nexthop-forwarding-address
<ip_addr_bgp_neighbor>
      end
```

Bind DHCP Service for Corporate Servers

Use this example to bind DHCP service with MPLS labels for input and output in Corporate network.

```
configure
   context <dest_ctxt_name>
      interface <intfc_name> loopback
         ip vrf forwarding <vrf_name>
         ip address <bind_ip_address subnet_mask>
         exit
```

```
      dhcp-service <dhcp_svc_name>
         dhcp ip vrf <vrf_name>
         bind address <bind_ip_address> [ nexthop-forwarding-address
<nexthop_ip_address> [ mpls-label input <in_mpls_label_value> output
<out_mpls_label_value1> [ <out_mpls_label_value2> ]]]
         dhcp server <ip_address>
         end
```

Notes:

- To ensure proper operation, DHCP functionality should be configured within a destination context.
- Optional keyword **nexthop-forwarding-address** <*ip_address*> **mpls-label input** <*in_mpls_label_value*> **output <** <*out_mpls_label_value1*> applies DHCP over MPLS traffic.

Bind AAA Group for Corporate Servers

Use this example to bind AAA server groups with MPLS labels for input and output in Corporate network.

```
configure
   context <dest_ctxt_name>
      aaa group <aaa_grp_name>
         radius ip vrf <vrf_name>
         radius attribute nas-ip-address address <nas_address>
nexthop-forwarding-address <ip_address> mpls-label input <in_mpls_label_value>
output < <out_mpls_label_value1>
         radius server <ip_address> encrypted key <encrypt_string> port <iport_num>

         end
```

Notes:

- *aaa_grp_name* is a pre-configured AAA server group configured in Context Configuration mode. Refer *AAA Interface Administration Reference* for more information on AAA group configuration.
- Optional keyword **nexthop-forwarding-address** <*ip_address*> **mpls-label input** <*in_mpls_label_value*> **output <** <*out_mpls_label_value1*> associates AAA group for MPLS traffic.

Configuring BGP/MPLS VPN with Dynamic Labels

This section describes the procedures required to configure the system as an MPLS-CE to interact with a PE with dynamic MPLS label support.

The base configuration, as described in the *Routing* chapter in this guide, must be completed prior to attempt the configuration procedure described below.



Important The features described in this chapter is an enhanced feature and need enhanced feature license. This support is only available if you have purchased and installed particular feature support license on your chassis.

Important Commands used in the configuration samples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To configure the system for BGP/MPLS VPN:

Procedure

Step 1 Create a VRF on the router and assign a VRF name, define the Route-Distinguisher first before you execute the Route Target. For more information on the configuration see the Create VRF with Route-distinguisher and Route-target, on page 5 section.

Step 2 Set the neighbors and address family to exchange routing information and establish BGP peering with a peer router by applying the example configuration in Set Neighbors and Enable VPNv4 Route Exchange, on page 9.

Step 3 Configure the address family and redistribute the connected routes domains into BGP by applying the example configuration in Configure Address Family and Redistributed Connected Routes, on page 9. This takes any routes from another protocol and redistributes them to BGP neighbors using the BGP protocol.

Step 4 Configure IP Pools with dynamic MPLS labels by applying the example configuration in Configure IP Pools with MPLS Labels, on page 9.

Step 5 *Optional*. Bind DHCP service to work with dynamic MPLS labels in corporate networks by applying the example configuration in Bind DHCP Service for Corporate Servers, on page 9.

Step 6 *Optional*. Bind AAA/RADIUS server group in corporate network to work with dynamic MPLS labels by applying the example configuration in Bind AAA Group for Corporate Servers, on page 10.

Step 7 *Optional*. Modify the configured IP VRF, which is configured to support basic MPLS functionality, for mapping between DSCP bit value and experimental (EXP) bit value in MPLS header for ingress and egress traffic by applying the example configuration in DSCP and EXP Bit Mapping, on page 10.

Step 8 Save your configuration as described in the *System Administration Guide*.

Create VRF with Route-distinguisher and Route-target

Use this example to first create a VRF on the router and assign a VRF name. The second **ip vrf** command creates the route-distinguisher and route-target.

```
configure
   context <context_name> -noconfirm
      ip vrf <vrf_name>
      router bgp <as_number>
         ip vrf <vrf_name>
            route-distinguisher {<as_value> | <ip_address>} <rt_value>
            route-target export {<as_value> | <ip_address>} <rt_value>
            route-target import {<as_value> | <ip_address>} <rt_value>
            end
```

Notes:

- If export and import route targets are the same, alternate command **route-target both** {*<as_value>* | *<ip_address>* } *<rt_value>* can be used in place of **route-target import** and **route-target export** commands.

Set Neighbors and Enable VPNv4 Route Exchange

Use this example to set the neighbors and address family to exchange VPNv4 routing information with a peer router.

```
configure
   context <context_name>
      mpls bgp forwarding
      router bgp <as_number>
         neighbor <ip_address> remote-as <AS_num>
         address-family vpnv4
         neighbor <ip_address> activate
         neighbor <ip_address> send-community both
         exit
      interface <bind_intfc_name>
         ip address <ip_addr_mask_combo>
         end
```

Configure Address Family and Redistributed Connected Routes

Use this example to configure the **address-family** and to **redistribute** the connected routes or IP pools into BGP. This takes any routes from another protocol and redistributes them using the BGP protocol.

```
configure
   context <context_name>
      router bgp <as_number>
         address-family ipv4 <type> vrf <vrf_name>
            redistribute connected
            end
```

Configure IP Pools with MPLS Labels

Use this example to configure IP Pools with dynamic MPLS labels.

```
configure
   context <context_name> -noconfirm
      ip pool <name> <ip_addr_mask_combo> private vrf <vrf_name>
      end
```

Bind DHCP Service for Corporate Servers

Use this example to bind DHCP service with dynamic MPLS labels in Corporate network.

```
configure
   context <dest_ctxt_name>
      interface <intfc_name> loopback
```

```
      ip vrf forwarding <vrf_name>
      ip address <bind_ip_address subnet_mask>
      exit
   dhcp-service <dhcp_svc_name>
      dhcp ip vrf <vrf_name>
      bind address <bind_ip_address>
      dhcp server <ip_address>
      end
```

Notes:

- To ensure proper operation, DHCP functionality should be configured within a destination context.

Bind AAA Group for Corporate Servers

Use this example to bind AAA server groups with dynamic MPLS labels in Corporate network.

```
configure
   context <dest_ctxt_name>
      aaa group <aaa_grp_name>
         radius ip vrf <vrf_name>
         radius attribute nas-ip-address address <nas_address>
         radius server <ip_address> encrypted key <encrypt_string> port <iport_num>

         end
```

Notes:

- *aaa_grp_name* is a pre-configured AAA server group configured in Context Configuration mode. Refer *AAA Interface Administration Reference* for more information on AAA group configuration.

DSCP and EXP Bit Mapping

Use this example to modify the configured IP VRF to support QoS mapping.

```
configure
   context <context_name>
      ip vrf <vrf_name>
         mpls map-dscp-to-exp dscp <dscp_bit_value> exp <exp_bit_value>
         mpls map-exp-to-dscp exp <exp_bit_value> dscp <dscp_bit_value>
         end
```