MPLS Support on UPF

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product (s) or Functional Area	5G-UPF
Applicable Platforms	VPC-SI
Feature Default Setting	Disabled – Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*UCC 5G UPF Configuration and Administration Guide*

Revision History

Revision Details	Release
First introduced.	2022.04.0

Feature Description

In the existing platforms (VPC-DI, ASR 5500), the boxer supports MPLS, which uses the underlying data plane forwarder to switch MPLS traffic. In ASR 5500, the NP4c network processor generates and processes MPLS traffic while in VPC-DI, the IFTask generates and processes MPLS traffic.

This feature enables MPLS support on UPF. VPC-SI uses VPP as the data plane forwarder. VPP supports and provides multiple data plane features that include the MPLS stack as a separate graph node. VPP encapsulates and decapsulates subscriber traffic with MPLS labels. This helps to differentiate between different customer VRFs and support many corporate APNs having different addressing models and requirements.

UPF supports the following functionalities for MPLS:

- Uses the VPP MPLS stack to send the MPLS labeled packet

- Uses the VPP MPLS stack to process the incoming labeled MPLS packet

- MPLS on UPF uses only MP-BGP as the label distribution protocol

- Supports VPPCTL CLI commands to display FTN and ILM tables that are in VPP for debugging and comparing values with boxer configuration

How it Works

This section briefly describes how the MPLS Support for UPF works.

In the current architecture, VPP forwarder provides its own MPLS stack, which supports all the existing functionalities for MPLS packet processing. The VPP MPLS stack is configured with the appropriate FTN (FEC To NHLFE) and incoming label map (ILM) tables. This generates the MPLS packet on the egress with the correct MPLS header. It also processes the incoming MPLS packet and switches the packet based on the incoming label to the appropriate VRF table.

VPC-SI also supports VPNv6 as described in RFC 4659 – *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*.

MPLS-CE Connected to PE

In this scenario the VPC-SI functions as an MPLS-CE (Customer Edge) network element connected to a Provider Edge (PE) Label Edge Router (LER), which in turn connects to the MPLS core (RFC 4364). See the figure below.

Figure 1: VPC-SI MPLS-CE to PE
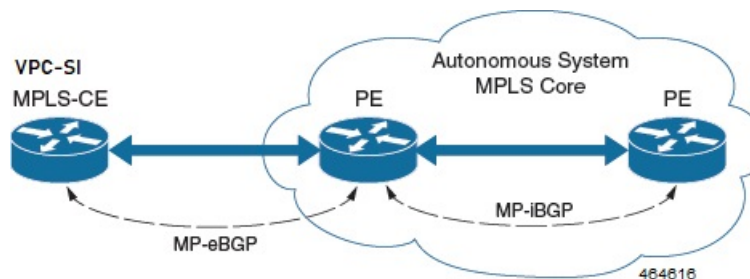
The MPLS-CE functions like a PE router within its own Autonomous System (AS). It maintains Virtual Routing and Forwarding (VRF) routes and exchanges VPN route information with the PE via an MP-eBGP (Multi-Protocol-external BGP) session.

The PE is also configured with VRFs and exchanges VPN routes with other PEs in its AS via MP-iBGP (Multi-Protocol-internal BGP) connections and the MPLS-CE via an MP-eBGP connection.

The EBGP connection allows the PE to change next-hop IP addresses and labels in the routes learned from IBGP peers before advertising them to the MPLS-CE. The MPLS-CE in this case uses only MP-eBGP to advertise and learn routes. Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) are not required because of direct-connect EBGP peering. The MPLS-CE in this scenario pushes/pops a single label (learned over the MP-eBGP connection) to/from the PE.

VPN-related CLI Commands

VPN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VPN-related functions.

Table 2: VPN-Related Configuration Commands

CLI Mode	Command	Description
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **send community { both	extended	standard }**	Sends the community attributes to a peer router (neighbor).
BGP Address-Family (IPv4/IPv6) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **send community { both	extended	standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.

CLI Mode	Command	Description
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **send community { both	extended	standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Configuration Mode	**address-family { ipv4 vrf** *vrf_name*	**vpnv4 }**	Enables the exchange of IPv4 VRF routing information. There is a different mode for each address-family.
BGP Configuration Mode	**address-family { ipv6 vrf** *vrf_name*	**vpnv6 }**	Configures a VPNv6 address family and IPv6 VRF routing in BGP.
BGP Configuration Mode	**ip vrf** *vrf_name*	Adds a VRF to BGP and switches to the VRF Configuration mode to allow configuration of BGP attributes for the VRF.
BGP IP VRF Configuration Mode	**route-distinguisher {** *as_value*	*ip_address* **}** *rd_value*	Assigns a Route Distinguisher (RD) for the VRF. The RD value must be a unique value on the router for each VRF.
BGP IP VRF Configuration Mode	**route-target { both	import	export } {** *as_value*	*ip_address* **}** *rt_value*	Adds a list of import and export route-target extended communities to the VRF.
Context Configuration Mode	**ip pool** *pool_name addr_range* **vrf** *vrf_name* **[mpls-label input** *inlabel1* **output** *outlabel1 outlabel2* **]**	Configures a pool into the specified VRF. This parameter must be specified with the Next-Hop parameter. *inlabel1* is the MPLS label that identifies inbound traffic destined for this pool. *outlabel1* and *outlabel2* specify the MPLS labels to be added to packets sent for subscribers from this pool.
Context Configuration Mode	**ip vrf** *vrf_name*	Creates a VRF and assigns a VRF-ID. A VRF is created in the router.

CLI Mode	Command	Description
Context Configuration Mode	**ipv6 pool** *pool_name* **vrf** *vrf_name*	Associates the pool with that VRF. **Note:** By default the configured ipv6 pool will be associated with the global routing domain.
Context Configuration Mode	**mpls bgp forwarding**	Globally enables MPLS Border Gateway Protocol (BGP) forwarding.
Context Configuration Mode	**mpls exp** *value*	Sets the default behavior as Best Effort using a zero value in the 3-bit MPLS EXP header. This value applies to all the VRFs in the context. The default behavior is to copy the DSCP value of mobile subscriber traffic to the EXP header, if there is no explicit configuration for DSCP to EXP (via the **mpls map-dscp-to-exp dscp** *n* **exp** *m* command). **mpls exp** disables the default behavior and sets the EXP value to the configured *value*.
Context Configuration Mode	**mpls ip**	Globally enables the MPLS forwarding of IPv4 packets along normally routed paths.
Context Configuration Mode	**radius change-authorize-nas-ip** *ip_address* **ip_address** *ip_address* **{ encrypted	key }** *value* **port** *port_num* **mpls input** *inlabel* **output** *outlabel1 outlabel2*	Configures COA traffic to use the specified MPLS labels. *inlabel* identifies inbound COA traffic. *outlabel1* and *outlabel2* specify the MPLS labels to be added to the COA response. *outlabel1* is the inner output label; *outlabel2* is the outer output label.
Ethernet Interface Configuration Mode	**mpls ip**	Enables dynamic MPLS forwarding of IP packets on this interface.
Exec Mode	**clear ip bgp peer**	Clears BGP sessions.

Table 3: VPN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show Commands	**show ip bgp neighbors**	Displays information regarding BGP neighbors.

CLI Mode	Command	Description
Exec Mode show Commands	**show ip bgp vpnv4 { all | route-distinguisher | vrf }**	Displays all VPNv4 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip bgp vpnv6**	Displays contents of VPNv6 routing table.
Exec Mode show Commands	**show ip bgp vpnv6 { all | route-distinguisher | vrf }**	Displays all VPNv6 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip pool**	Displays pool details including the configured VRF.
Exec Mode show Commands	**show mpls cross-connect**	Displays MPLS cross-connect information. MPLS tunnel cross-connects between interfaces and Label-Switched Paths (LSPs) connect two distant interface circuits of the same type via MPLS tunnels that use LSPs as the conduit.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name*	Displays MPLS FEC-to-NHLFE (FTN) table information.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name* **]**	Displays contents of the MPLS FTN table for a specified VRF.
Exec Mode show Commands	**show mpls ilm**	Displays MPLS Incoming Label Map (ILM) table information.
Exec Mode show Commands	**show mpls nexthop-label-forwarding-entry**	Displays MPLS Next-Hop Label Forwarding Entry (NHLFE) table information.

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show mpls ftn vpp

The output of this CLI command contains the "vpp" field for the MPLS Support on UPF feature.

This field enables viewing of the VPP dataplane values that are confiigured in the VPP dataplane forwarder. This show command is used for debugging along with the existing debug commands.

- vpp
    - all-vrf
    - summary
    - vrf