Cisco Catalyst SD-WAN Network Configuration Guide, Releases 26.x and Later

Cisco SD-WAN
Updated April 24, 2026

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Topics included

1 VPN....11

VPN....12

Interfaces in the WAN Transport VPN....12

Interfaces in the Management....13

Configure VPN....13

Configure VPN using configuration groups....14

Configure VPN using templates....28

Configure VPN parameters using CLI commands....30

Verify the VPN configuration....32

2 TLOC....35

TLOC....36

System IP address....37

Color....37

Encapsulation....38

3 TLOC Extension....39

Feature history for TLOC extension....40

TLOC extension....40

TLOC extension over IPv6....40

Limitations for TLOC extension over IPv6....41

How TLOC extension over IPv6 works....41

Configure TLOC extension using CLI commands....42

Verify TLOC extension....44

4 Bandwidth Utilization Reference Values....47

Feature history for bandwidth utilization Reference Values....48

Bandwidth Utilization Reference Values....48

Generating Notifications....48

Monitoring Bandwidth Utilization with Interface Charts....49

Restrictions for bandwidth utilization reference values....49

Configure bandwidth utilization reference values....49

Configure bandwidth utilization reference values using a configuration group....49

Configure bandwidth utilization reference values using CLI commands....50

Configure bandwidth utilization reference values using templates....51

Monitor bandwidth utilization events....51

Monitor bandwidth utilization reference values....51

Verify bandwidth utilization reference values....52

5 Carrier Supporting Carrier....53
Feature history for carrier supporting carrier....54
Carrier supporting carrier....54
Traffic flow configuration between CSC-CE and CSC-PE devices....55
Edge device functioning as a CSC-CE device....55
Use cases for carrier supporting carrier....56
Configure carrier supporting carrier....56
Configure carrier supporting carrier using a feature template....57
Configure carrier supporting carrier using CLI....58
Verify device configuration for carrier supporting carrier....60

6 DHCP Vendor Option Support....63
Feature history for DHCP vendor option support....64
DHCP vendor option support....64
Configure a DHCP vendor option using CLI command....66
Configure a DHCPv6 client option option using a CLI template....67
Configure DHCP server using templates....68

7 Dynamic On-Demand Tunnels....73
Feature history for dynamic on-demand tunnels....74
Dynamic on-demand tunnels....74
How on-demand tunnels work....74
How on-demand tunnels work with a transport gateway....76
Prerequisites for on-demand tunnels....77
Prerequisites: OMP settings....77
Prerequisites: Hub device traffic engineering service....78
Prerequisites: Spoke Device ECMP Limit....79
Restrictions for on-demand tunnels....80
Configure on-demand tunnels....80
Configure on-demand tunnels using control policy....81
Configure on-demand tunnels using a transport gateway....83
Monitor the status of on-demand tunnels....85
View the current status of on-demand tunnels using Cisco SD-WAN Manager....85
View a chart of the on-demand tunnel status over time in Cisco SD-WAN Manager....85
View the route to a destination device....86
View OMP routes....86

8 GRE-in-UDP....89
Feature history for GRE-in-UDP....90
GRE-in-UDP....90
Configure GRE-in-UDP using CLI commands....90

9 IP DHCP Smart-Relay....93
Feature history for IP DHCP smart-relay....94
IP DHCP smart-relay....94
Benefits of IP DHCP smart-relay....95
Configure IP DHCP smart-relay agent using a CLI template....95

10 Layer 2 VPN....97
Feature history for L2VPN....98
Layer 2 VPNs within the SD-WAN overlay network....98
Supported platforms for Layer 2 VPN....101
Restrictions for Layer 2 VPN....101
Methods to configure Layer 2 VPN using CLI template....102
Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template....102
Configure a point-to-point Layer 2 VPN using CLI template....103
Configure an edge router at Site A for point-to-point Layer 2 VPN using CLI template....103
Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple....104
Configure a point-to-multipoint Layer 2 VPN using CLI template....106
Configure an edge router at sites A, B and C....106
Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple....107
Configure an edge router at Site C for point-to-point Layer 2 VPN using CLI template....108
Configure a Layer 2 VPN Switchport using CLI template....110
Methods to verify Layer 2 VPN using CLI....111
View a Layer 2 VPN status....112
View L2VPN information learned through OMP route on a Cisco SD-WAN Controller....112
View Bridge-domain information....113
View Cisco Catalyst SD-WAN Flood List Information and Packet Counters in Data Plane....114
View packet counters in data plane....114
Monitor configured layer 2 VPN using CLI....116

11 IPv6 Functionality....119
Configure IPv6 functionality....120
Configure IPv6 functionality for an Interface or Subinterface using templates....120
Configure IPv6 functionality for an interface or subinterface using CLI commands....120
Configure IPv6 functionality for OMP using templates....120
Configure IPv6 functionality for OMP using CLI commands....121
Configure IPv6 functionality for BGP using templates....122
Configure IPv6 functionality for BGP using CLI commands....123
Configure IPv6 functionality for VRRP using templates....124
Configure IPv6 functionality for VRRP using CLI commands....125
Configure IPv6 functionality for SNMP using templates....125
Configure IPv6 functionality for SNMP using CLI commands....126
Configure IPv6 functionality for a DHCP relay agent using templates....127

Configure IPv6 functionality for a DHCP relay agent using CLI commands....127
Configure IPv6 functionality for ACL and QoS using templates....128
Configure IPv6 functionality for ACL and QoS using CLI commands....128
Configure IPv6 functionality for a logging host using templates....130
Configure IPv6 functionality for a logging host using CLI commands....130
Configure IPv6 functionality for a prefix list using templates....131
Configure IPv6 functionality for a prefix list using CLI commands....131
Configure IPv6 functionality for a data prefix using templates....132
Configure IPv6 functionality for a data prefix using CLI commands....132
Configure IPv6 functionality for a centralized policy using templates....132
Configure IPv6 functionality for a centralized policy using CLI commands....133
Configure IPv6 functionality for a localized policy using templates....133
Configure IPv6 functionality for a localized policy using CLI commands....133

12 IPv6 in a Dual Stack Environment....135

Dual stack environment with IPv6 as the preferred address family....136
Dual stack environment with IPv6 as the preferred address family....136
Methods for configuring a dual stack environment with IPv6 as the preferred address family....137
Configure IPv6 as the preferred address family for devices, using a configuration group....138
Configure IPv6 as the preferred address family for devices, using the Quick Connect workflow....138
Configure IPv6 as the preferred address family for devices, using CLI commands....139
Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using CLI commands....140
Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using templates....140
Monitor the use of IPv6 as the preferred address family in a dual stack environment, in SD-WAN Manager....141
Monitoring IPv6 as the preferred address family in a dual stack environment....141

13 DHCP for IPv6....143

Prerequisites for DHCP for IPv6....144
Restrictions for DHCP for IPv6....144
DHCP for IPv6....144
Use cases for DHCP for IPv6....146
Methods for configuring DHCP for IPv6 using CLI commands....147
Methods for verifying a DHCPv6 client and server configuration using CLI commands....150

14 Per Packet Load Balancing....155

Feature history for per packet load balancing....156
Per packet load balancing....156
Key concepts for understanding PPL....156
Sequence of events for PPL 157
Benefits of PPL in Cisco SD-WAN....157
Use cases for PPL....158

Supported platforms for PPL....158
Restrictions for PPL....159
Configure PPL....159
Configure PPL using configuration groups....160
Configure a data policy with load balancing using policy groups....161
Configure maximum packet value in PPL using CLI commands....161
Configure latency offset value in PPL using CLI commands....161
Monitor PPL....162

15 PPPoE....167
Feature history for PPPoE....168
PPPoE....168
PPPoE over ATM....168
Supported platforms for PPPoE over ATM....169
Configure PPPoE using templates....169
Configure PPPoE over ATM using templates....172
Configuration examples for PPPoE....173

16 TCP MSS and Clear Dont Fragment....175
Feature history for TCP MSS and clear dont fragment....176
TCP MSS....176
Restrictions for TCP MSS and clear dont fragment....176
Configure TCP MSS and clear dont fragment....177
Configure TCP MSS and clear dont fragment using CLI commands....177
Configure TCP MSS and clear dont fragment using templates....178
Verify TCP MSS and dont clear fragment configurations....179

17 Track Static Routes for Service VPNs....181
Feature history of track static routes for service VPNs....182
Track static routes for service VPNs....182
Supported platforms....183
Restrictions for IPv4 static route tracking....183
Configure tracker group using a configuration group....183
Create a static route tracker....184
Configure a next hop static route with tracker....186
Monitor static route tracker configuration....187
Configure static routes using CLI....187
Configuration examples static route tracking....189
Verify static route tracking configuration....191

18 VDSL and GSHDSL....193
VDSL and GSHDSL....194
VDSL configuration guidelines....194

GSHDSL configuration guidelines....196
GSHDSL EFM-ATM NIM....197
Cisco VDSL examples....197
Cisco GSHDSL examples....200

19 VFR and Underlay Fragmentation....203
Feature history for VFR and underlay fragmentation....204
Virtual Fragmentation Reassembly....204
Underlay fragmentation....205
Benefits of VFR and underlay fragmentation....205
Prerequisites for configuring VFR and underlay fragmentation....205
Restrictions for configuring VFR and underlay fragmentation....205
Use cases for VFR and underlay fragmentation....206
Boost mode....206
Configure VFR and underlay fragmentation....206
Configure underlay fragmentation using a configuration group....206
Configure VFR using CLI commands....207
Configure underlay fragmentation using CLI commands....208
Enable boost mode using CLI commands....210
Verify VFR and underlay fragments....210

20 Packet-order preservation during tunnel underlay reassembly....217
Feature history for packet order preservation during tunnel underlay reassembly....218
Fragmentation and reassembly packet order preservation processes....218
Use cases for packet-order preservation during tunnel underlay reassembly....218
Enable boost mode in SD-WAN Manager....219
Verify tunnel underlay reassembly boost mode is active....219

21 Cisco SD-WAN Controller Route Filtering by TLOC Color....221
Feature history for route filtering by TLOC color....222
Route filtering mechanism by TLOC color....222
How Cisco SD-WAN Controller route filtering by TLOC color works....222
Benefits of Cisco SD-WAN Controller route filtering by TLOC color....225
Supported devices for Cisco SD-WAN Controller route filtering by TLOC color....225
Prerequisites for Cisco SD-WAN Controller route filtering by TLOC color....225
Restrictions for Cisco SD-WAN Controller route filtering by TLOC color....226
Configure Cisco SD-WAN Controller route filtering by TLOC color using a CLI template....226
Enable route filtering using a CLI template....226
Configure the update interval for route filtering by TLOC color using a CLI template....227
Override default TLOC color compatibility for Cisco SD-WAN Controller route filtering by TLOC color using a CLI template....227
Monitor Cisco SD-WAN Controller route filtering by TLOC color....228
View TLOC colors for a device....229

Check TLOC color compatibility....229

22 Network Slicing....231
Feature history of network slicing....232
Network slicing....232
Configure network slicing....232
Configure network slicing using CLI....235
Verify network slicing using CLI....235

23 Transport Gateway....237
Feature history for transport gateway....238
Transport gateways for connecting networks in Cisco SD-WAN....238
How a router functions as a transport gateway....239
Site type....240
OMP best path logic and transport gateway path preference....240
How the transport gateway configuration works....241
Use cases for transport gateways....244
Restrictions for transport gateways....246
Configure a router as a transport gateway....247
Configure a router as a transport gateway using a CLI template....247
Configure a router as a transport gateway using Cisco SD-WAN Manager....248
Configure the transport gateway path preference....248
Configure the transport gateway path preference using a CLI Template....248
Configure the transport gateway path preference using Cisco SD-WAN Manager....249
Configure the site type for a router....250
Configure the site type for a router using a CLI template....250
Configure the site type for a router using Cisco SD-WAN Manager....251
Verify the site type of a router using the CLI....251
Verify a transport gateway configuration using the CLI....252

24 Symmetric Routing....253
Feature history for symmetric routing....254
Symmetric routing mechanisms for Cisco SD-WAN....254
Translating OMP metrics for devices outside of the overlay network....255
Translating OMP metrics to BGP attributes....256
Translating OMP metrics to an OSPF metric....258
Symmetric routing scenarios....258
Configuration overview....260
Example configurations for symmetric routing in transport gateway and multi-region fabric deployments....261
Example of configuration for symmetric routing and the mechanism....263
Supported scenarios....275
Configure symmetric routing....285
Configure a router to use automatic affinity group preference 285

Configure router affinity groups for specific VRFs....286
Configure a router to translate OMP metrics to BGP or OSPF using CLI commands....287
Verify symmetric routing....288
Monitor RIB metric translation....290
View OMP metrics 291
View BGP metrics....292
View OSPF metrics....292

25 Radio Aware Routing....295
Feature history for RAR....296
RAR for Cisco IOS XE Catalyst SD-WAN devices....296
System components of RAR....297
RAR architecture....297
Benefits of RAR....299
Supported devices for RAR....299
Restrictions for RAR....299
Prerequisites for RAR....299
Configure RAR....300
Configure RAR in bypass mode....301
Configure RAR in aggregate mode....303

1 VPN

Topics:

- VPN
- Configure VPN
- Verify the VPN configuration

Introduces VPN functionality and references key aspects of VPN deployment, management, and integration within SD-WAN environments.

VPN

Explains foundational VPN concepts, including configuration of interfaces in the WAN transport VPN and management VPN, to provide essential background for VPN deployment and operation.

A VPN template in Cisco Catalyst SD-WAN is a configuration template that

- enables the creation of separate feature templates for each VPN, and
- supports configuration of VPN 0 and VPN 512 on all device types, with additional VPN templates for segmenting service-side user networks on Cisco IOS XE Catalyst SD-WAN devices.

Types of VPNs

The types of VPNs in Cisco Catalyst SD-WAN include:

- **VPN 0–Transport VPN**, which carries control traffic via the configured WAN transport interfaces. Initially, VPN 0 contains all of a device's interfaces except for the management interface, and all interfaces are disabled.
- **VPN 512–Management VPN**, which carries out-of-band network management traffic among the Cisco IOS XE Catalyst SD-WAN devices in the overlay network. The interface used for management traffic resides in VPN 512. By default, VPN 512 is configured and enabled on all Cisco IOS XE Catalyst SD-WAN devices. For controller devices, by default, VPN 512 is not configured.
- **VPNs 1–511, 513–65530–Service VPNs**, for service-side data traffic on Cisco IOS XE Catalyst SD-WAN devices.

You create a separate VPN feature template for each VPN. For example, create one feature template for VPN 0, a second for VPN 1, and a third for VPN 512.

Interfaces in the WAN Transport VPN

A VPN 0 is a WAN transport VPN that

- handles all control plane traffic carried over OMP sessions in the overlay network,
- requires at least one interface configured in VPN 0 for a Cisco IOS XE Catalyst SD-WAN device to participate in the overlay network, and
- mandates that at least one interface connects to a WAN transport network, such as the Internet or an MPLS or a metro Ethernet network.

Tunnel interface configurations

The WAN transport interface, known as a tunnel interface, is configured in VPN 0.

To configure a tunnel interface on a Cisco SD-WAN Controller or a Cisco SD-WAN Manager, you must create an interface in VPN 0, assign an IP address (static or via DHCP), enable the interface with the **no shutdown** command, and mark it as a tunnel interface.

The IP address can be either IPv4 or IPv6. To enable dual stack, configure both address types. Optionally, you can associate a color with the tunnel.

> **Note**
>
> You can configure IPv6 addresses only on transport interfaces in VPN 0. Configuring IPv6 addresses is not supported in VPN 512.

On Cisco IOS XE Catalyst SD-WAN devices, tunnel interfaces must have an IP address, a color, and an encapsulation type. For releases before Cisco IOS XE Catalyst SD-WAN Release 17.3.2, dual stack is enabled by configuring both IPv4 and IPv6 addresses. Starting from Release 17.3.2, only one address type is supported per TLOC or interface. Using a second address type requires a second TLOC or interface on which it can be provisioned.

On Cisco Catalyst SD-WAN Controllers and Controller NMSs, interface names can be either ethnumber or loopbacknumber, and only VPN 0 and VPN 512 are supported for interface configuration.

On Cisco SD-WAN Controller and Cisco SD-WAN Manager, *interface-name* can be either **eth** *number* or **loopback** *number*, and only VPN 0 and VPN 512 are supported for interface configuration. Hence, all interfaces are present only on these VPNs.

Dual stack configuration

To use dual stack with Cisco IOS XE Catalyst SD-WAN devices from Cisco IOS XE Catalyst SD-WAN Release 17.3.2, configure all controllers with both IPv4 and IPv6 addresses. In addition, configure DNS for the Cisco SD-WAN Validator interface to resolve IPv4 and IPv6 address types so that controllers can reach the Cisco SD-WAN Validator through either IP address type.

Starting from Cisco vManage Release 20.6.1, in case of a dual-stack configuration, if an IPv4 address or the fully qualified domain name (FQDN) is not available, but an IPv6 address is available, then the IPv6 address is used to connect to the Cisco SD-WAN Validator.

Interfaces in the Management

VPN 512 is a default out-of-band management VPN that

- is included as part of the factory-default configuration for out-of-band management, and
- is converted to VRF Mgmt-Intf on Cisco IOS XE Catalyst SD-WAN devices, which use VRFs in place of VPNs.

VPN 512 is local to the device and not advertised in the overlay. If you need a management VPN that is reachable through the overlay, create a VPN with a number other than 512.

Configure VPN

Details VPN configuration procedures using configuration groups, templates, and CLI commands, covering transport, management, and service VPN setup, load-balancing algorithm configuration, and host name-to-IP address mapping.

Use one of these methods to configure VPN parameters:

- Configuration group
- Feature template
- CLI commands

Configure VPN using configuration groups

Configure transport VPN using a configuration group

Before you begin

On the **Configuration** > **Configuration Groups** page, choose **SD-WAN** as the solution type.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Create and configure a Transport VPN feature in Transport and Management profile.
   a) Enter the basic configuration information.
   Table 1: Basic Configuration

Field	Description
VPN	Enter the numeric identifier of the VPN.
Enhance ECMP Keying	Enable the use in the ECMP hash key of Layer 4 source and destination ports, in addition to the combination of the source IP address, destination IP address, protocol, and DSCP field, as the ECMP hash key. Default: Disabled

   b) Enter DNS information.
   Table 2: DNS

Field	Description
Add DNS	
Primary DNS Address (IPv4)	Enter the IP address of the primary IPv4 DNS server in this VPN.
Secondary DNS Address (IPv4)	Enter the IP address of a secondary IPv4 DNS server in this VPN.
Add DNS IPv6	
Primary DNS Address (IPv6)	Enter the IP address of the primary IPv6 DNS server in this VPN.
Secondary DNS Address (IPv6)	Enter the IP address of a secondary IPv6 DNS server in this VPN.

   c) Enter host happing information.
   Table 3: Host Mapping

Field	Description
Add New Host Mapping	
Hostname*	Enter the hostname of the DNS server. The name can be up to 128 characters.

Field	Description
List of IP*	Enter up to 14 IP addresses to associate with the hostname. Separate the entries with commas.

3. Configure the following parameters based on the features you choose to configure on your network.
   a) Enter the route details.

Table 4: Route

Field	Description
Add IPv4 Static Route	
Network address*	Enter the IPv4 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv4 static route to configure in the VPN.
Subnet Mask*	Enter the subnet mask.
Gateway*	Choose one of the following options to configure the next hop to reach the static route: • **nextHop**: When you choose this option and click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv4 address. • **Administrative distance***: Enter the administrative distance for the route. • **dhcp** • **null0**: When you choose this option, the following field appears: • **Administrative distance**: Enter the administrative distance for the route.
Add IPv6 Static Route	
Prefix*	Enter the IPv6 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv6 static route to configure in the VPN.

Field	Description
Next Hop/Null 0/NAT	Choose one of the following options to configure the next hop to reach the static route: • **Next Hop**: When you choose this option and click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv6 address. **Administrative distance***: Enter the administrative distance for the route. • **Null 0**: When you choose this option, the following field appears: • **IPv6 Route Null 0***: Enable this option to set the next hop to be the null interface. All packets sent to this interface are dropped without sending any ICMP messages. • **NAT**: When you choose this option, the following field appears: • **IPv6 NAT***: Choose NAT64 or NAT66.
Add BGP Routing	Choose a BGP route.

b) Enter the NAT details

Table 5: NAT

Field	Description
Add NAT64 v4 Pool	
NAT64 v4 Pool Name*	Enter a NAT pool number configured in the centralized data policy. The NAT pool name must be unique across VPNs and VRFs. You can configure up to 31 (1–32) NAT pools per router.
NAT64 Pool Range Start*	Enter a starting IP address for the NAT pool.
NAT64 Pool Range End*	Enter a closing IP address for the NAT pool.
NAT64 Overload	Enable this option to configure per-port translation. If this option is disabled, only dynamic NAT is configured on the end device. Per-port NAT is not configured. Default: Disabled

c) Enter the service information.

Table 6: Service

Field	Description
Add Service	

Field	Description
Service Type	Choose the service available in the VPN. Value: **TE**

What to do next

Also see Deploy a Configuration Group.

Configure management VPN using a configuration group

Before you begin

On the **Configuration** > **Configuration Groups** page, choose **SD-WAN** as the solution type.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Create and configure a Management VPN feature in Transport and Management profile.
   a) Enter the basic configuration information.

Table 7: Basic Configuration

Field	Description
VPN	Management VPN carries out-of-band network management traffic among the Cisco IOS XE Catalyst SD-WAN devices in the overlay network. The interface used for management traffic resides in VPN 512. By default, VPN 512 is configured and enabled on all Cisco IOS XE Catalyst SD-WAN devices.
Name	Enter a name for the interface.

   b) Enter DNS information.

Table 8: DNS

Field	Description
Add DNS	
Primary DNS Address (IPv4)	Enter the IPv4 address of the primary DNS server in this VPN.
Secondary DNS Address (IPv4)	Enter the IPv4 address of a secondary DNS server in this VPN.
Add DNS IPv6	
Primary DNS Address (IPv6)	Enter the IPv6 address of the primary DNS server in this VPN.
Secondary DNS Address (IPv6)	Enter the IPv6 address of a secondary DNS server in this VPN.

c) Enter host happing information.

Table 9: Host Mapping

Field	Description
Add New Host Mapping	
Hostname*	Enter the hostname of the DNS server. The name can be up to 128 characters.
List of IP Address*	Enter IP addresses to associate with the hostname. Separate the entries with commas.

d) Enter the IPv4/IPv6 static route information.

Table 10: IPv4/IPv6 Static Route

Field	Description
Add IPv4 Static Route	
IP Address*	Enter the IPv4 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv4 static route to configure in the VPN.
Subnet Mask*	Enter the subnet mask.
Gateway*	Choose one of the following options to configure the next hop to reach the static route: • **nextHop**: When you choose this option and click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv4 address. • **Administrative distance***: Enter the administrative distance for the route. • **dhcp** • **null0**: When you choose this option, the following field appears: • **Administrative distance**: Enter the administrative distance for the route.
Add IPv6 Static Route	
Prefix*	Enter the IPv6 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv6 static route to configure in the VPN.

Field	Description
Next Hop/Null 0/NAT	Choose one of the following options to configure the next hop to reach the static route: • **Next Hop**: When you choose this option and click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv6 address. **Administrative distance***: Enter the administrative distance for the route. • **Null 0**: When you choose this option, the following field appears: • **NULL0***: Enable this option to set the next hop to be the null interface. All packets sent to this interface are dropped without sending any ICMP messages. • **NAT**: When you choose this option, the following field appears: • **IPv6 NAT**: Choose NAT64 or NAT66.

What to do next

Also see Deploy a Configuration Group.

Configure service VPN using a configuration group

This section helps you configure a service VPN (range 1 – 65527, except 512) or the LAN VPN.

Before you begin

On the **Configuration** > **Configuration Groups** page, choose **SD-WAN** as the solution type.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Create and configure Service VPN in a Service profile.
   a) Enter the basic configuration information.

Table 11: Basic Configuration

Field	Description
VPN*	Enter the numeric identifier of the VPN.
Name*	Enter a name for the VPN.
OMP Admin Distance IPv4	Administrative distance for OMP routes. The Cisco SD-WAN Controllers learn the topology of the overlay network and the services available in the network using OMP routes. The distance can be a value between 1–255.
OMP Admin Distance IPv6	Administrative distance for OMP routes. The Cisco SD-WAN Controllers learn the topology of the overlay network and the services available in the network using OMP routes. The distance can be a value between 1–255.

b) Enter DNS information.

Table 12: DNS

Field	Description
Add DNS IPv4	
Primary DNS Address (IPv4)	Enter the IP address of the primary IPv4 DNS server in this VPN.
Secondary DNS Address (IPv4)	Enter the IP address of a secondary IPv4 DNS server in this VPN.
Add DNS IPv6	
Primary DNS Address (IPv6)	Enter the IP address of the primary IPv6 DNS server in this VPN.
Secondary DNS Address (IPv6)	Enter the IP address of a secondary IPv6 DNS server in this VPN.

c) Enter host happing information.

Table 13: Host Mapping

Field	Description
Add New Host Mapping	
Hostname*	Enter the hostname of the DNS server. The name can be up to 128 characters.
List of IP*	Enter up to eight IP addresses to associate with the hostname. Separate the entries with commas.

3. Configure the following parameters based on the features you choose to configure on your network.

a) Enter advertise OMP information.

Table 14: Advertise OMP

Field	Description
Add OMP Advertise IPv4	

Field	Description
Protocol	Choose a protocol to configure route advertisements to OMP, for this VPN: • **bgp** • **ospf** • **ospfv3** • **connected** • **static** • **network** • **aggregate** • **eigrp** • **lisp** • **isis**
Select Route Policy	Enter the name of the route policy. Route policy is not supported in Cisco vManage Release 20.9.1.
Add OMP Advertise IPv6	
Protocol	**Note** Advertising IPv6 OMP routes as network statements is not supported. This applies when using the Service VPN feature in a configuration group, and applies also when using a Cisco VPN feature template. You can configure to advertise: • IPv6 routes by BGP and OSPF protocols • Connected routes, static routes, and aggregate routes The reason for the lack of support is that the Service VPN feature and the Cisco VPN feature template both use the **advertise network** *prefix* command, which does not fully support IPv6 addresses. Choose a protocol to configure route advertisements to OMP, for this VPN: • **BGP** • **OSPF** • **Connected** • **Static** • **Network** • **Aggregate**

Field	Description
Select Route Policy	Enter the name of the route policy. Route policy is not supported in Cisco vManage Release 20.9.1.
Protocol Sub Type	When you choose the OSPF protocol, specify the sub type as external.

b) Enter route information.

Table 15: Route

Field	Description
Add IPv4 Static Route	
Network Address*	Enter the IPv4 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv4 static route to configure in the VPN.
Subnet Mask*	Enter the subnet mask.

Field	Description
Next Hop/Null 0/VPN/DHCP	Choose one of the following options to configure the next hop to reach the static route: • **Next Hop**: When you choose this option, the **IPv4 Route Gateway Next Hop** field appears. Enable this option to add the next hop. You can add a hop with and without a tracker. When you click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv4 address. • **Administrative Distance***: Enter the administrative distance for the route. When you click **Add Next Hop with Tracker**, the following fields appear: • **Address***: Enter the next-hop IPv4 address. • **Administrative Distance***: Enter the administrative distance for the route. • **Tracker***: Enter the name of the gateway tracker to determine whether the next hop is reachable before adding that route to the route table of the device. • **Null 0**: When you choose this option, the following field appears: • **IPv4 Route Null 0***: Enable this option to set the next hop to be the null interface. All packets sent to this interface are dropped without sending any ICMP messages. • **VPN**: When you choose this option, the following field appears: • **IPv4 Route VPN***: Selects VPN as the gateway to direct packets to the transport VPN. • **DHCP**: When you choose this option, the following field appears: • **IPv4 Route Gateway DHCP***: Assigns a static route for the default next-hop router when the DHCP server is accessed for an IP address.
Add BGP Routing	Choose a BGP route.
Add OSPF Routing	Choose an OSPF route.
Add IPv6 Static Route	
Prefix*	Enter the IPv6 address or prefix, in decimal four-point-dotted notation, and the prefix length of the IPv6 static route to configure in the VPN.

Field	Description
Next Hop/Null 0/NAT	Choose one of the following options to configure the next hop to reach the static route: • **Next Hop**: When you choose this option and click **Add Next Hop**, the following fields appear: • **Address***: Enter the next-hop IPv6 address. • **Administrative distance***: Enter the administrative distance for the route. • **Null 0**: When you choose this option, the following field appears: • **IPv6 Route Null 0***: Enable this option to set the next hop to be the null interface. All packets sent to this interface are dropped without sending any ICMP messages. • **NAT**: When you choose this option, the following field appears: • **IPv6 NAT***: Choose NAT64 or NAT66.

c) Enter service information.

Table 16: Service

Field	Description
Add Service	
Service Type	Choose a service available at the local site and in the VPN. Values: **FW**, **IDS**, **IDP**, **netsvc1**, **netsvc2**, **netsvc3**, **netsvc4**, **TE**, **SIG**
IPv4 Addresses (Maximum: 4)*	Enter up to four IP address, separated by commas. The service is advertised to the Cisco SD-WAN Controller only if one of the addresses can be resolved locally, at the local site, not via routes learned through OMP. You can configure up to four IP addresses.
Tracking*	Cisco Catalyst SD-WAN tests each service device periodically to check whether it is operational. Tracking saves the results of the periodic tests in a service log. Tracking is enabled by default.

d) Enter service route information.

Table 17: Service Route

Field	Description
Add Service Route	
Prefix*	Enter the IP address or prefix. For Umbrella SIG, use any RFC 1918 subnet for Service IP addresses.

Field	Description
Service*	Configure routes pointing to any service. Values: **FW**, **IDS**, **IDP**, **netsvc1**, **netsvc2**, **netsvc3**, **netsvc4**.
VPN*	Destination VPN to resolve the prefix.

e) Enter GRE route information.

Table 18: GRE Route

Field	Description
Add GRE Route	
Prefix*	Enter the IP address or prefix, in decimal four-part-dotted notation, and prefix length of the GRE-specific static route.
Interface*	Enter the name of one or two GRE tunnels to use to reach the service.
VPN*	Enter the number of the VPN to reach the service. This must be VPN 0.

f) Enter IPSEC route information.

Table 19: IPSEC Route

Field	Description
Add ipSec Route	
Prefix*	Enter the IP address or prefix, in decimal four-part-dotted notation, and prefix length of the IPsec-specific static route.
Interface*	Enter the name of one or two IPsec tunnel interfaces. If you configure two interfaces, the first is the primary IPsec tunnel, and the second is the backup. All packets are sent only to the primary tunnel. If that tunnel fails, all packets are then sent to the secondary tunnel. If the primary tunnel comes back up, all traffic is moved back to the primary IPsec tunnel.

g) Enter NAT information.

Table 20: NAT

Field	Description
Nat Pool	
NatPool Name*	Enter a NAT pool number configured in the centralized data policy. The NAT pool name must be unique across VPNs and VRFs. You can configure up to 31 (1–32) NAT pools per router.
Prefix Length*	Enter the NAT pool prefix length.
Range Start*	Enter a starting IP address for the NAT pool.
Range End*	Enter a closing IP address for the NAT pool.

Field	Description
Overload*	Enable this option to configure per-port translation. If this option is disabled, only dynamic NAT is configured on the end device. Per-port NAT is not configured. Default: Enabled
Direction*	Choose the NAT direction.
Nat64 V4 Pool	
Nat64 V4 Pool Name*	Enter a NAT pool number configured in the centralized data policy. The NAT pool name must be unique across VPNs and VRFs. You can configure up to 31 (1–32) NAT pools per router.
Nat 64 V4 Pool Range Start*	Enter a starting IP address for the NAT pool.
Nat 64 V4 Pool Range End*	Enter a closing IP address for the NAT pool.
Overload*	Enable this option to configure per-port translation. If this option is disabled, only dynamic NAT is configured on the end device. Per-port NAT is not configured. Default: Disabled

h) Enter route leak information.

Table 21: Route leak from Global VPN

Field	Description
Route Protocol*	Choose a protocol to configure leak routes from global VPN to the service VPN that you are configuring: • **static** • **connected** • **bgp** • **ospf**
Select Route Policy	Choose a route policy from the drop-down list.
Redistribution (in service VPN)	
Protocol*	Choose a protocol from the available options to redistribute the leaked routes: • **bgp** • **ospf** • (Minimum supported release: Cisco Catalyst SD-WAN Manager Release 20.18.1) **eigrp**
Select Route Policy	Choose a route policy from the drop-down list.

Table 22: Route leak to Global VPN

Field	Description
Route Protocol*	Choose a protocol to leak routes from the service VPN that you are configuring to the global VPN: • **static** • **connected** • **bgp** • **ospf** • (Minimum supported release: Cisco Catalyst SD-WAN Manager Release 20.18.1) **eigrp**
Select Route Policy	Choose a route policy from the drop-down list.
Redistribution (in global VPN)	
Protocol*	Choose a protocol from the available options to redistribute the leaked routes: • **bgp** • **ospf**
Select Route Policy	Enter the name of the route policy.
Select Route Policy	Choose a route policy from the drop-down list.

Table 23: Route leak between services

Field	Description
Source VPN	Enter a value of the source VPN.
Route Protocol*	Choose a protocol from the available options to leak routes from the source service VPN to the service VPN that you are configuring: • **static** • **connected** • **bgp** • **ospf** • (Minimum supported release: Cisco Catalyst SD-WAN Manager Release 20.18.1) **eigrp**
Select Route Policy	Choose a route policy from the drop-down list.
Redistribution (in Service VPN)	

Field	Description
Protocol*	Choose a protocol from the available options to redistribute the leaked routes: • **bgp** • **ospf** • (Minimum supported release: Cisco Catalyst SD-WAN Manager Release 20.18.1) **eigrp**
Select Route Policy	Choose a route policy from the drop-down list.

i) Enter route target information.

Table 24: Route Target

Field	Description
IPv4 Settings	
Import Route Target List: Route Target*	Configure a route target for IPv4 interfaces. It imports routing information from the target VPN extended community.
Export Route Target List: Route Target*	Configure a route target for IPv4 interfaces. It exports routing information to the target VPN extended community.
IPv6 Settings	
Import Route Target List: Route Target*	Configure a route target for IPv6 interfaces. It imports routing information from the target VPN extended community.
Export Route Target List: Route Target*	Configure a route target for IPv6 interfaces. It exports routing information to the target VPN extended community.

What to do next

Also see Deploy a Configuration Group.

Configure VPN using templates

Cisco IOS XE Catalyst SD-WAN devices use VRFs for segmentation and network isolation. However, the following steps still apply if you are configuring segmentation for Cisco IOS XE Catalyst SD-WAN devices through Cisco SD-WAN Manager. When you complete the configuration, the system automatically converts the VPNs to VRFs for Cisco IOS XE Catalyst SD-WAN devices.

You can configure a static route through the VPN template.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Device Templates**, and click **Create Template**.

In Cisco vManage Release 20.7.x and earlier releases **Device Templates** is called **Device**.

3. From the **Create Template** drop-down list, choose **From Feature Template**.
4. From the **Device Model** drop-down list, choose the type of device for which you wish to create the template.
5. To create a template for VPN 0 or VPN 512:
   a) Click **Transport & Management VPN**, or scroll to the **Transport & Management VPN** section.
   b) From the VPN 0 or VPN 512 drop-down list, click **Create Template**. The VPN template form appears.

   The form contains fields for naming the template, and fields for defining VPN parameters.

6. To create a template for VPNs 1 through 511, and 513 through 65527:
   a) Click **Service VPN**, or scroll to the **Service VPN** section.
   b) Click the **Service VPN** drop-down list.
   c) From the **VPN** drop-down list, click **Create Template**. The VPN template form displays.

   The form contains fields for naming the template, and fields for defining VPN parameters.

7. In **Template Name**, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.
8. In **Template Description**, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.
9. Configure the VPN template parameters.
   a) Configure basic VPN parameters.

Parameter Name	Description
VPN	Enter the numeric identifier of the VPN. Range for Cisco IOS XE Catalyst SD-WAN devices: 0 through 65527 Values for Cisco SD-WAN Controller and Cisco SD-WAN Manager devices: 0, 512
Name	Enter a name for the VPN. **Note** For Cisco IOS XE Catalyst SD-WAN devices, you can't enter a device-specific name for the VPN.
Enhance ECMP keying	Click **On** to enable the use in the ECMP hash key of Layer 4 source and destination ports, in addition to the combination of the source, and destination IP addresses, as the ECMP hash key. ECMP keying is **Off** by default.
OMP Admin Distance (IPv4)	To configure a site to prefer the OMP route over the leaked route path, configure the IPv4 address with a lower administrative distance than the leaked route. You can apply the configuration at the global level or at the specific VRF level by choosing **Global** or **Device Specific** respectively. Range: 1-251

Parameter Name	Description
OMP Admin Distance (IPv6)	To configure a site to prefer the OMP route over the leaked route path, configure the IPv6 address with a lower administrative distance than the leaked route. You can apply the configuration at the global level or at the specific VRF level by choosing **Global** or **Device Specific** respectively. Range: 1-251

> **Note**
>
> To complete the configuration of the transport VPN on a router, you must configure at least one interface in VPN 0.

b) Configure DNS addresses and static hostname mapping.

Parameter Name	Options	Description
Primary DNS Address	Click either **IPv4** or **IPv6**, and enter the IP address of the primary DNS server in this VPN.	
New DNS Address	Click **New DNS Address** and enter the IP address of a secondary DNS server in this VPN. This field appears only if you have specified a primary DNS address.	
	Mark as Optional Row	Check the **Mark as Optional Row** check box to mark this configuration as device-specific. To include this configuration for a device, enter the requested variable values when you attach a device template to a device, or create a template variables spreadsheet to apply the variables.
	Hostname	Enter the hostname of the DNS server. The name can be up to 128 characters.
	List of IP Addresses	Enter up to eight IP addresses to associate with the hostname. Separate the entries with commas.

To save the DNS server configuration, click **Add**.

Configure VPN parameters using CLI commands

Configure load-balancing algorithm using CLI commands

From Cisco IOS XE Catalyst SD-WAN Release 17.8.1a, you need CLI template to configure the **src-only** load-sharing algorithm for IPv4 and IPv6 Cisco Catalyst SD-WAN and non Cisco Catalyst SD-WAN traffic. For complete details on the load-sharing algorithm CLI, see IP Commands list.

Follow these steps to configure load-balancing algorithm using CLI commands:

Procedure

1. Select a Cisco Express Forwarding load-balancing algorithm for non Cisco Catalyst SD-WAN IPv4 and IPv6 traffic.

Example

```
Device# config-transaction
Device(config)# ip cef load-sharing algorithm {universal [id] | include-ports
 [ source [id] | destination [id]] |
src-only [id]}
```

```
Device# config-transaction
Device(config)# ipv6 cef load-sharing algorithm {universal [id] | include-ports
 [ source [id] | destination [id]] |
src-only [id]}
```

2. Enable load balancing algorithm on an interface for Cisco Catalyst SD-WAN IPv4 and IPv6 traffic.

Example

```
Device# config-transaction
Device(config)# sdwan
Device(config-sdwan)# ip load-sharing algorithm {ip-and-ports | src-dst-ip |
 src-ip-only}
```

```
Device# config-transaction
Device(config)# sdwan
Device(config-sdwan)# ipv6 load-sharing algorithm {ip-and-ports | src-dst-ip
 | src-ip-only}
```

Map host names to IP addresses using CLI commands

Perform this task to associate host names with IP addresses.

A name server is used to keep track of information associated with domain names. A name server can maintain a database of host name-to-address mappings. Each name can map to one or more IP addresses. In order to use this service to map domain names to IP addresses, you must specify a name server.

Procedure

---

1. Define a static host name-to-address mapping in the host name cache.

Example

```
Device(config)# ip host cisco-rtp 192.168.0.148
```

2. Define a default domain name that Cisco Catalyst SD-WAN can use to complete unqualified host names.

Example

```
Device(config)# ip domain name cisco.com
```

3. Specify one or more hosts that supply name information.

Example

```
Device(config)# ip name-server 172.16.1.111 172.16.1.2
```

4. Enable DNS-based address translation.

   DNS is enabled by default. Use this command if DNS has been disabled.

   Example

```
Device(config)# ip domain lookup
```

The following example configures the host-name-to-address mapping process. IP DNS-based translation is specified, the addresses of the name servers are specified, and the default domain name is given.

```
! IP DNS-based host name-to-address translation is enabled
  ip domain lookup
! Specifies hosts 192.168.1.111 and 192.168.1.2 as name servers
  ip name-server 192.168.1.111 192.168.1.2
! Defines cisco.com as the default domain name the device uses to complete
! Set the name for unqualified host names
  ip domain name cisco.com
```

Verify the VPN configuration

Provides verification steps and reference commands to confirm correct VPN configuration and operational status within the network.

This section provides examples for VPN configurations.

Use the **show sdwan running-config | sec vrf definition Mgmt-intf** command to verify the management interface configurations.

```
Device# show sdwan running-config | sec vrf definition Mgmt-intf

vrf definition Mgmt-intf
 address-family ipv4
  exit-address-family
 !
 address-family ipv6
  exit-address-family
 !
============
interface GigabitEthernet0
 no shutdown
 vrf forwarding Mgmt-intf
 negotiation auto
exit
============
config-t
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0

vrf definition Mgmt-intf
 rd 1:512
```

```
 !
 address-family ipv4
  route-target export 1:512
  route-target import 1:512
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
interface GigabitEthernet1
 vrf forwarding Mgmt-intf
 ip address 192.168.20.11 255.255.255.0
!
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0
!
```

To display information about the configured management interfaces, use the **show interface** command.

```
Device# show interface gigabitEthernet0
GigabitEthernet0 is up, line protocol is up
  Hardware is RP management port, address is d478.9bfe.9f7f (bia d478.9bfe.9f7f)

  Internet address is 10.34.9.177/16
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 1000Mbps, link type is auto, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 8000 bits/sec, 12 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
     4839793 packets input, 415574814 bytes, 0 no buffer
     Received 3060073 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     82246 packets output, 41970224 bytes, 0 underruns
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

2 TLOC

Topics:

- TLOC
- System IP address
- Color
- Encapsulation

Introduces TLOC reference information, explaining identifiers, attributes, and their significance in SD-WAN topology.

TLOC

Describes the TLOC concept, outlining key elements, operational roles, and relationships within the SD-WAN environment.

A TLOC (Transport Locator) is a unique identifier in Cisco Catalyst SD-WAN that

- represents a WAN Edge device's connection to a WAN transport, and
- is defined by the combination of its system IP address, a color indicating the type of transport, and an encapsulation type.

Components of TLOC

A TLOC is made up of three components:

- System IP address: The unique IP address of the SD-WAN device.
- Color: A Cisco Catalyst SD-WAN software construct that identifies the transport tunnel.
- Encapsulation: The method used to encapsulate the overlay tunnel data.

Tunnel interface

A tunnel interface in Cisco Catalyst SD-WAN is a network connection that you configure for secure data transport. On a Cisco SD-WAN Controller or Cisco SD-WAN Manager, you can configure one tunnel interface. On a Cisco IOS XE Catalyst SD-WAN device, you can configure up to eight tunnel interfaces.

To limit the remote TLOCs that the local TLOC can establish BFD sessions with, mark the TLOC with the **restrict** option. When a TLOC is marked as restricted, a TLOC on the local router establishes tunnel connections with a remote TLOC only if the remote TLOC has the same color.

When a WAN edge device is configured with two IPv6 TLOCs, one with static default route and the other one with IPv6 address autoconfig default which is the IPv6 neighbor discovery default route, the IPv6 neighbor discovery default route is not installed in the routing table. In this case, the IPv6 TLOC with IPv6 neighbor discovery default route does not work.

For IPv6 TLOC with IPv6 neighbor discovery default route to work, you can configure the static route for TLOC with IPv6 neighbor discovery to overwrite the IPv6 neighbor discovery default route and ensure that both the static routes are installed into the routing table. You can also use the IPv6 neighbor discovery default route on all interfaces.

A tunnel interface allows only DTLS, TLS, and, for Cisco IOS XE Catalyst SD-WAN devices, IPsec traffic to pass through the tunnel. To allow additional traffic to pass without having to create explicit policies or access lists, enable them by including one **allow-service** command for each service. You can also explicitly disallow services by including the **no allow-service** command. Note that services affect only physical interfaces.

STUN server

In Cisco Catalyst SD-WAN, Session Traversal Utilities for NAT (STUN) is a protocol used by Cisco IOS XE Catalyst SD-WAN devices to discover their public IP address and port assigned by a Network Address Translator (NAT).

Use the **allow-service stun** command to enable or disable a Cisco IOS XE Catalyst SD-WAN device from sending requests to a generic STUN server. This allows the device to determine if it is behind a NAT, identify the NAT type, and discover its public IP address and port number. On a Cisco IOS XE Catalyst SD-WAN device that is behind a NAT, you can also configure a tunnel interface to obtain its public IP address and port number from the Cisco SD-WAN Validator.

When you configure the Cisco IOS XE Catalyst SD-WAN device to use the Cisco SD-WAN Validator as a STUN server, the device determines its public IP address and public port number, which enables it to establish TLOC connections and form the overlay fabric over various public transports like broadband or cellular networks. However, the device cannot identify the type of NAT it is behind in this setup. The tunnel interface configured for the Cisco SD-WAN Validator does not carry overlay network control traffic or exchange encryption keys, but BFD establishes connectivity and allows data

traffic. Because this tunnel does not support control traffic, you must configure at least one additional tunnel interface to ensure the device can exchange control traffic with the Cisco SD-WAN Controller and Cisco SD-WAN Manager.

You can log the headers of all packets that the system drops because they do not match a service configured with an **allow-service** command. You can use these logs for security purposes, for example, to monitor the flows that are being directed to a WAN interface and to determine, in the case of a DDoS attack, which IP addresses to block.

System IP address

Details the system interface's role in TLOC configuration, including attributes and integration with TLOCs in SD-WAN systems.

A system interface IP address is a persistent address that

- identifies the Cisco IOS XE Catalyst SD-WAN device,
- is similar to a router ID on a regular router, and
- is used to identify the router from which packets originated.

System IP address configuration

You configure a system interface for each Cisco IOS XE Catalyst SD-WAN device using the **system system-ip** command. Specify the system IP address as an IPv4 address in decimal four-part dotted notation, without including the prefix length; the /32 prefix is implicit. The system IP address must not be within the following ranges: 0.0.0.0/8, 127.0.0.0/8, 224.0.0.0/4, or 240.0.0.0/4 and later. Assign a unique system IP address to each device in the overlay network. You cannot assign this address to another interface in VPN 0.

The system interface is placed in VPN 0 as a loopback interface named **system**. This loopback is not the same as a loopback address that you configure for a specific interface. To display information about the system interface, use the **show interface** command.

Role in OMP TLOC identification

The system IP address is used as one of the attributes of the OMP TLOC (Overlay Management Protocol Transport Locator). Each TLOC is uniquely identified by a 3-tuple: the system IP address, a color, and an encapsulation. Use the **show omp tlocs** command to display TLOC information.

Device management

For device management, configure the same system IP address on a loopback interface located in a service-side VPN appropriate for management purposes. Use a loopback interface because it remains reachable whenever the router is operational and the overlay network is up. Avoid configuring the system IP address on a physical interface, since both the router and the interface must be up for reachability in that case.

Assign the loopback interface to a service-side VPN, which is any VPN other than VPN 0 (the WAN transport VPN) or VPN 512 (the management VPN). Service-side VPNs are used to route data traffic and remain reachable from the data center.

Color

Explains the function of color in TLOC settings, describing color types and usage in transport identification and SD-WAN policies.

A color is a Cisco Catalyst SD-WAN software construct that

- identifies the transport tunnel,

- includes options such as **3g**, **biz-internet**, **blue**, **bronze**, **custom1**, **custom2**, **custom3**, **default**, **gold**, **green**, **lte**, **metro-ethernet**, **mpls**, **private1** through **private6**, **public-internet**, **red**, and **silver**, and
- designates **metro-ethernet**, **mpls**, and **private1** through **private6** as private colors that use private addresses for private networks, which can be used on public networks only if there is no NAT device between the local and remote Cisco IOS XE Catalyst SD-WAN devices.

Encapsulation

Outlines encapsulation methods used with TLOCs, describing encapsulation types and their significance in TLOC transport configurations.

An encapsulation is a required configuration on Cisco IOS XE Catalyst SD-WAN devices that

- specifies the tunnel encapsulation type as either IPsec or GRE, and
- sets the default MTU to 1442 bytes for IPsec and 1468 bytes for GRE, which is enabled by default on all TLOCs, based on BFD path MTU discovery.

You can configure both IPsec and GRE encapsulation by including two **encapsulation** commands under the same **tunnel-interface** command. On the remote Cisco IOS XE Catalyst SD-WAN device, you must configure the same tunnel encapsulation type or types so that the two routers can exchange data traffic. Data transmitted out of an IPsec tunnel can be received only by an IPsec tunnel, and data sent on a GRE tunnel can be received only by a GRE tunnel. The Cisco Catalyst SD-WAN software automatically selects the correct tunnel on the destination Cisco IOS XE Catalyst SD-WAN device.

3 TLOC Extension

Topics:

- Feature history for TLOC extension
- TLOC extension
- How TLOC extension over IPv6 works
- Configure TLOC extension using CLI commands
- Verify TLOC extension

Details TLOC extension reference information, introducing key concepts, operational principles, usage scenarios, and providing an overview of its role in SD-WAN architectures.

Feature history for TLOC extension

Summarizes the historical evolution of TLOC extension, highlighting major feature enhancements, platform updates, and release milestones relevant to SD-WAN deployments.

Table 25: Feature History

Feature Name	Release Information	Description
TLOC Extension Over IPv6	Cisco IOS XE Catalyst SD-WAN Release 17.11.1a Cisco vManage Release 20.11.1	This feature enables the support of TLOC extension for IPv6. In the previous releases, TLOC extension was supported only for IPv4.

TLOC extension

Introduces TLOC extension fundamentals, including implementation over IPv6 and outlining associated limitations and design principles for WAN connectivity.

A TLOC Extension is a Cisco Catalyst SD-WAN feature that

- enables a device to access the opposite WAN transport connected to a neighboring device using a TLOC extension interface, and
- addresses scenarios where devices cannot connect directly to a single transport and only one device can connect to each transport.

Benefits of TLOC extension

There are scenarios when Cisco IOS XE Catalyst SD-WAN devices cannot connect to a single transport directly and only one device can connect to a single transport. A switch is connected to each transport and the devices connect to each transport through the switches. TLOC extension provides the following benefits:

- Eliminates the need for additional switches at branch locations.
- Reduces overall solution costs and simplifies network management.

TLOC extension over IPv6

From Cisco IOS XE Catalyst SD-WAN Release 17.11.1a TLOC extension over IPv6 works only if the underlay supports IPv6 addressing on both the Cisco IOS XE Catalyst SD-WAN devices connecting each other.

In the earlier releases, TLOC extension was supported only over IPv4 interfaces.

Supported configurations

This feature supports the following requirements:

- Implicit IPv6 ACL on TLOC tunnel interface.
- Private and public color TLOC interfaces.
- Dual stack support. When both IPv4 and IPv6 are configured, the tunnel is built on top of either IPv4 or IPv6, based on the configuration.
- NAT66 support. The limitations of NAT66 also applies to the TLOC extended interface.

- Only the Layer 2 setup supports IPv6 TLOC extension. The following interface types supports IPv6 TLOC extension:
  - Physical interface
  - Physical sub-interface
  - Loopback interface
- Loopback TLOC interface that is bound to either:
  - The WAN transport circuit.
  - An extended WAN interface between two Cisco IOS XE Catalyst SD-WAN devices.

Limitations for TLOC extension over IPv6

SIG

Secure Internet Gateway (SIG) is not supported on TLOC extension over IPv6.

NAT64

NAT64 is not supported for TLOC extension over IPv6.

Layer 3 Connectivity

TLOC extension over IPv6 is not supported for Layer 3 connections.

Control connection persistence

When a TLOC configuration is extended to a peer interface and then to the internet service provider, the extended control connections remain active on the peer interface even after the TLOC extension configuration is removed.

Extender interface configuration

In TLOC extension, the extender interface is part of the Cisco Catalyst SD-WAN. However, configuring a tunnel-interface under the extender interface is optional.

How TLOC extension over IPv6 works

Explains the workflow and processes involved in TLOC extension operations over IPv6, describing mechanisms and interactions crucial for SD-WAN connectivity.

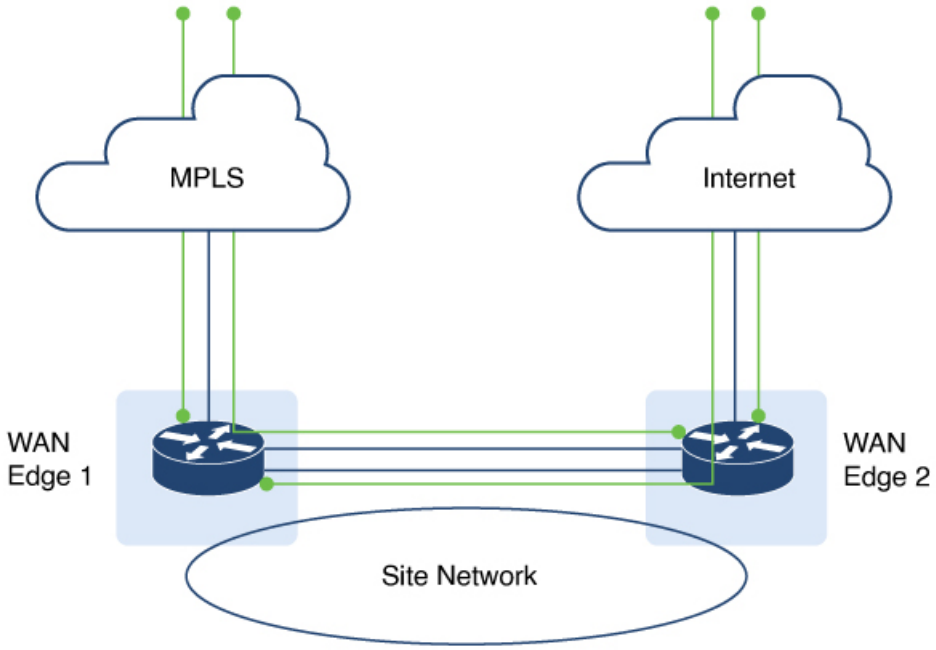
Summary

The key components involved in TLOC extension over IPv6 are:

- Establish TLOC extension interfaces: Each Cisco IOS XE Catalyst SD-WAN device configures a TLOC-extension interface to enable access to the transport network of its neighboring SD-WAN device.
- Access opposite transport via neighbor.

Workflow

Figure 1: TLOC extension



This process outlines how TLOC extension interfaces facilitate transport access and redundancy between two Cisco IOS XE Catalyst SD-WAN devices.

1. SD-WAN device 1 accesses the internet transport through the TLOC-extension interface on SD-WAN device 2, in addition to its direct MPLS connection.
2. SD-WAN device 2 accesses the MPLS transport through the TLOC-extension interface on SD-WAN device 1, in addition to its direct internet connection.

Result

TLOC extension over IPv6 achieves redundancy in a dual-device deployment scenario with only one circuit connection on each device.

Configure TLOC extension using CLI commands

Provides instructions for configuring TLOC extension, detailing step-by-step procedures for implementation within SD-WAN environments.

Follow these steps to configure TLOC extension using CLI commands:

Procedure

1. Enter global configuration mode, and configure an interface.

   Example

   ```
   Device# config-transaction
   ```

2. Enter SD-WAN configuration mode.

Example

```
Device(config)# sdwan
```

3. in the SD-WAN configuration mode, configure an interface type such as, Gigabit Ethernet.

   Example

   ```
   Device(config-sdwan)# interface GigabitEthernet3
   ```

4. Configure tunnel interface.

   Example

   ```
   Device(config-interface-GigabitEthernet3)# tunnel-interface
   ```

5. Configure encapsulation, color, allowed services for TLOC.

   Example

   ```
   Device(config-interface-GigabitEthernet3)# tunnel-interface
   Device(config-interface-GigabitEthernet3)# encapsulation ipsec
   Device(config-interface-GigabitEthernet3)# color color
   Device(config-interface-GigabitEthernet3)# exit
   ```

6. In the global configuration mode, configure an interface.

   Example

   ```
   Device# config-transaction
   Device(config)# ip route 0.0.0.0 0.0.0.0 ip-address
   ```

7. On device 2, the LTE WAN connection is on GigabitEthernet1 and this transport is extended to device 1 GigabitEthernet3 TLOC interface.

   Example

   ```
   Device(config-sdwan)# tloc-extension GigabitEthernet1
   ```

8. Configure NAT routes on GigabitEthernet1 for data traffic to reach back to device 1 through device 2 for GigabitEthernet3 subnet.

The following example describes how TLOC extension is configured on a network interface.

```
On Device1,
Configure TLOC interface on VPN 0
sdwan
interface GigabitEthernet3
  tunnel-interface
   encapsulation ipsec
```

```
   color custom1
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
   allow-service https
   no allow-service snmp
   no allow-service bfd
  exit

  Configure default route via this TLOC interface with nexthop
  to L2 connected interface of the peer (ED2 Gig3).

  ip route 0.0.0.0 0.0.0.0 10.1.19.16

 On Device2,
LTE WAN connection is on Gig1 and this transport is extended to ED1 Gig3 TLOC
 interface(custom1).
 sdwan
 int GigabitEthernet3
 tloc-extension GigabitEthernet1
Configure NAT routes on Gig1 or appropriate routes for data traffic to reach
back to ED1 via ED2 for Gig3 subnet.
```

Verify TLOC extension

Outlines commands and best practices for verifying TLOC extension functionality, ensuring successful deployment and ongoing operational validation.

The following is a sample output of the commands to verify if TLOC extension is configured on a network interface.

```
Device# show sdwan control connections
PEER                                        PEER
              CONTROLLER
PEER    PEER PEER            SITE       DOMAIN PEER
          PRIV  PEER
PUB                                         GROUP
TYPE    PROT SYSTEM IP       ID         ID     PRIVATE IP
          PORT  PUBLIC IP
PORT  ORGANIZATION             LOCAL COLOR     PROXY STATE UPTIME      ID
------------------------------------------------------------------------------------------
vsmart  dtls 172.16.255.19   100        1      2001:a0:5::13

12455 2001:a0:5::13                         12455 vIPtela Inc Regression
custom1             No    up
0:01:23:06 0
vsmart  dtls 172.16.255.20   200        1      2001:a0:c::14
          12456
2001:a0:c::14                           12456 vIPtela Inc Regression  custom1
```

```
             No    up
0:01:23:06 0
```

```
Device# show sdwan bfd sessions
                                       SOURCE TLOC      REMOTE TLOC
DST PUBLIC                         DST PUBLIC         DETECT      TX
  SYSTEM IP        SITE ID     STATE       COLOR            COLOR
SOURCE IP
IP                                          PORT        ENCAP  MULTIPLIER
  INTERVAL(msec  UPTIME
TRANSITIONS
-------------------------------------------------------------------------------------------
172.16.255.14    400         up          custom1             lte
 2001:a0:15::10
2001:a1:e::e                                 12346       ipsec  7           1000
0:00:05:50  3
```

4 Bandwidth Utilization Reference Values

Topics:

- Feature history for bandwidth utilization Reference Values
- Bandwidth Utilization Reference Values
- Generating Notifications
- Monitoring Bandwidth Utilization with Interface Charts
- Restrictions for bandwidth utilization reference values
- Configure bandwidth utilization reference values
- Monitor bandwidth utilization events
- Verify bandwidth utilization reference values

Introduces bandwidth utilization reference values for network interfaces, highlighting their significance in assessing interface performance and informing capacity planning.

Feature history for bandwidth utilization Reference Values

Details the feature history for bandwidth utilization reference values, emphasizing enhancements and updates that optimize network interface monitoring.

Table 26: Feature History

Feature Name	Release Information	Description
Upstream and Downstream Bandwidth Reference Values	Cisco IOS XE Catalyst SD-WAN Release 17.16.1a Cisco Catalyst SD-WAN Manager Release 20.16.1	Use the upstream and downstream bandwidth reference values to govern how Cisco SD-WAN Manager displays interface utilization percentages in charts. The values also act as configurable thresholds that trigger interface-bw events when a network interface's utilization exceeds a defined point.

Bandwidth Utilization Reference Values

Explains conceptual information regarding bandwidth utilization reference values, including their purpose and role in network interface performance management.

A bandwidth utilization reference value is a configurable parameter that

- specifies the upstream (egress) bandwidth and downstream (ingress) bandwidth for each interface, and
- acts as a reference for calculating bandwidth utilization and generating notifications.

You can configure bandwidth utilization reference values for each interface. Cisco SD-WAN Manager uses these reference values to display bandwidth utilization percentages in charts.

Generating Notifications

Provides instructions for generating notifications based on bandwidth utilization reference values, enabling proactive network monitoring and alerting.

To receive notifications when the traffic bandwidth exceeds 85% utilization, configure reference values for both upstream (transmitted) and downstream (received) traffic. These values act as thresholds for generating the interface bandwidth events. The range is from 1 and 2,147,483,647 kbps.

The device samples the interface traffic each 10 seconds. If the received or transmitted bandwidth exceeds 85 percent of the configured value in 85 percent of the sampled intervals in a continuous 5-minute period, the device generates an SNMP trap. After the first trap is generated, sampling continues at the same frequency, but notifications are rate-limited to once per hour. A second trap is sent, and subsequent traps are sent, if the bandwidth exceeds 85 percent of the value in 85 percent of the 10-second sampling intervals over the next 1-hour period. If, after 1 hour, the device does not send another SNMP trap, the notification interval reverts to 5 minutes.

The upstream bandwidth and downstream bandwidth settings are solely for monitoring purposes and do not impose a bandwidth limit on the traffic. For example, in some network configurations, the full bandwidth of an interface may not be available. To ensure that utilization values reflect the available network bandwidth, set bandwidth utilization reference values lower than an interface's full speed.

You can verify the configured upstream and downstream bandwidth values using the **show interface detail** command, which displays the upstream bandwidth (tx-kbps) and downstream bandwidth (rx-kbps) fields usage.

You can monitor transport circuit bandwidth on Cisco IOS XE Catalyst SD-WAN devices and on Cisco SD-WAN Manager.

Monitoring Bandwidth Utilization with Interface Charts

Guides monitoring of bandwidth utilization using interface charts, facilitating visualization and analysis of network performance metrics.

Cisco SD-WAN Manager provides a chart showing bandwidth utilization for each interface of a device. To view the chart, from the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**, click a device and click **Interface**.

Configuring upstream or downstream reference values affects how the chart displays the percentages:

- No bandwidth utilization reference values configured: By default, devices calculate the bandwidth utilization value according to the interface speed of the connection.
- Bandwidth utilization reference values configured: If you configure bandwidth utilization reference values, devices calculate the bandwidth utilization value as a percentage of the reference point.

  For example, if you configure upstream bandwidth and downstream bandwidth values of 500 megabits per second, and if the downstream utilization is 500 megabits per second, the device reports downstream utilization as 100%.

Devices limit the calculated utilization value to 100% even if the traffic utilization is more than 100% of the configured reference value.

Restrictions for bandwidth utilization reference values

Describes restrictions applicable to bandwidth utilization reference values, outlining limitations and considerations for their use in network management.

- Cisco SD-WAN Manager supports bandwidth data rate and utilization statistics only for a primary network interface, not a subinterface or loopback interface. You can still configure upstream and downstream bandwidth under a subinterface or loopback interface as a reference value or for per-tunnel QoS functionality.
- Configure upstream and downstream bandwidth references for service-side VPN interfaces using only a CLI template.

Configure bandwidth utilization reference values

Outlines configuration methods for bandwidth utilization reference values, including using configuration groups, CLI commands, and templates to tailor reference values for network interfaces.

Use one of these methods to configure bandwidth utilization reference values:

- Configuration group
- CLI commands
- Feature template

Configure bandwidth utilization reference values using a configuration group

Follow these steps to configure bandwidth utilization reference values using a configuration group

Before you begin

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.16.1a and Cisco Catalyst SD-WAN Manager Release 20.16.1

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Under Transport & Management Profile, click **...** adjacent to the transport profile, and choose **Edit**.
3. Click the edit icon adjacent to Ethernet Interface.
4. In the **Basic Configuration** section, enter the upstream and downstream bandwidth reference values.
5. Click **Save**.

Configure bandwidth utilization reference values using CLI commands

This section provides example CLI configurations to configure upstream and downstream bandwidth reference values in Cisco SD-WAN Manager. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

By default, CLI templates execute commands in global config mode.

Procedure

1. Enter the SD-WAN configuration mode.

   Example

   ```
   sdwan
   ```

2. Enter interface configuration mode.

   Example

   ```
   interface interface-name
   ```

3. Define the upstream or downstream bandwidth reference value in kbps.

   Example

   ```
   bandwidth-upstream upstream-value
   ```

   ```
   bandwidth-downstream downstream-value
   ```

Here is a complete configuration example for configuring upstream and downstream bandwidth reference values:

```
sdwan
!
interface GigabitEthernet1
```

```
bandwidth-upstream 10000
bandwidth-downstream 10000
```

Configure bandwidth utilization reference values using templates

Follow these steps to configure bandwidth utilization reference values using feature templates.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Do one of these:
   - Click **Add Template**, choose a device, and create a Cisco VPN Interface Ethernet template.
   - Edit an existing Cisco VPN Interface Ethernet template.
4. In the **Basic Configuration** section, enter the upstream and downstream bandwidth reference values.
5. Click **Update** to save the template.

Monitor bandwidth utilization events

Guides you through monitoring upstream and downstream bandwidth utilization events in vManage, including accessing event logs and viewing interface bandwidth details.

Follow these steps to monitor upstream and downstream bandwidth events using the Cisco SD-WAN Manager.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Logs** > **Events**.
2. View the details of upstream and downstream bandwidth values in the device configuration section. The default name for interface bandwidth-related events is **interface-bw**.

Monitor bandwidth utilization reference values

Follow these steps to monitor bandwidth utilization reference values using Cisco SD-WAN Manager.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.
2. Click the **Device** tab.
3. Click the hostname of the device you want to monitor.
4. From the **Applications** section, choose **Interface**.
5. Click the **Chart Options** drop-down list and choose **Utilization**.
6. Hover over the real-time or historical data points to view the utilization percentage.

Verify bandwidth utilization reference values

Provides procedures for verifying bandwidth utilization reference values, ensuring accuracy and reliability in network performance assessment.

Monitor interface alarms

Use the **show platform software sdwan interface-alarm summary** command to display the bandwidth reference values for an interface in the alarm summary. This sample output shows the summary details of a GigabitEthernet interface alarm.

See Interface Alarm Summary, and Base and High Intervals for more information.

```
device# show platform software sdwan interface-alarm summary

============ Interface Alarm Summary ============
Interface Name       Upstream (kbps)  Downstream (kbps)   Base Interval(s)
High Interval(s)
GigabitEthernet1      10000               10000               300
     3600
```

5 Carrier Supporting Carrier

Topics:

- Feature history for carrier supporting carrier
- Carrier supporting carrier
- Traffic flow configuration between CSC-CE and CSC-PE devices
- Edge device functioning as a CSC-CE device
- Use cases for carrier supporting carrier
- Configure carrier supporting carrier
- Verify device configuration for carrier supporting carrier

Introduces Carrier Supporting Carrier fundamentals, establishing the basis for integrating layered carrier networks to deliver scalable, resilient connectivity across service provider domains.

Feature history for carrier supporting carrier

Details the feature evolution of Carrier Supporting Carrier, highlighting major enhancements and development milestones that impact deployment options and interopera

Table 27: Feature history

Feature name	Release information	Description
Cisco Catalyst SD-WAN Support for Carrier Supporting Carrier Connectivity	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	The feature adds support for carrier supporting carrier (CSC) connectivity on Cisco IOS XE Catalyst SD-WAN devices. CSC enables you to interconnect IP or multiprotocol label switching (MPLS) networks operating at different sites over an MPLS backbone network. Using CSC requires an edge router that supports CSC functionality, called a carrier edge (CE) device, at each site. This feature enables a Cisco IOS XE Catalyst SD-WAN device to serve as a CE device, making it unnecessary to have a separate dedicated CE device at each site managed by Cisco Catalyst SD-WAN.

Carrier supporting carrier

Explains core concepts and operational principles of Carrier Supporting Carrier, clarifying foundational technologies and how carrier-level services are layered and managed.

A carrier supporting carrier (CSC) is a hierarchical VPN model that

- allows organizations to interconnect their IP or MPLS networks located at different sites,
- operates over an MPLS backbone network, and
- eliminates the need for organizations to build and maintain their own MPLS backbone.

Components of carrier supporting carrier

- Backbone carrier: The service provider operates the backbone network. Typically, the backbone carrier network employs multiple segments to segregate the traffic of different customer carriers that share it. The same organization as the customer carriers or a different organization manages the backbone carrier.
- Customer carrier: An organization that uses the backbone network to route traffic from one site to another. The customer carrier may be part of the organization that operates the backbone network, or may be independent.
- CSC-CE: The customer edge (CE) device operates within a local site network and connects the site to the backbone carrier using an MPLS connection. It uses the backbone carrier to connect to other sites.
- CSC-PE: The provider edge (PE) device operates within the backbone carrier network and connects to CSC-CE devices at customer sites using an MPLS connection.

Benefits of carrier supporting carrier

A Cisco IOS XE Catalyst SD-WAN device functions as an customer edge device at a site requiring CSC, eliminating the need for a separate CE router.

Traffic flow configuration between CSC-CE and CSC-PE devices

Describes MPLS-based traffic routing mechanisms in Carrier Supporting Carrier deployments, focusing on label switching, hierarchical routing, and inter-carrier traffic flows.

The traffic flow configuration between CSC-CE and CSC-PE devices determines how service VPN, control, and BFD probe traffic is routed based on available MPLS and internet connections. This configuration optimizes network resources, ensures efficient traffic handling, and supports high availability in service provider networks.

Traffic flow

When a CSC-CE device has only an MPLS connection to its neighboring CSC-PE device, it sends all traffic, including service VPN traffic, control traffic, and Cisco Catalyst SD-WAN bidirectional forwarding detection (BFD) probe traffic, over the MPLS connection.

When a CSC-CE device has both an MPLS connection to the neighboring CSC-PE device and a separate internet connection, it routes traffic as follows:

- Based on the configured traffic policy, it can send control traffic and BFD probe traffic over either the internet or the MPLS connection.
- It sends service VPN traffic exclusively over the MPLS connection.

Label switching

When traffic uses an MPLS connection between a CSC device and the backbone carrier, the backbone carrier manages the traffic through label-switched paths and does not store any information about the customer carrier routes.

Edge device functioning as a CSC-CE device

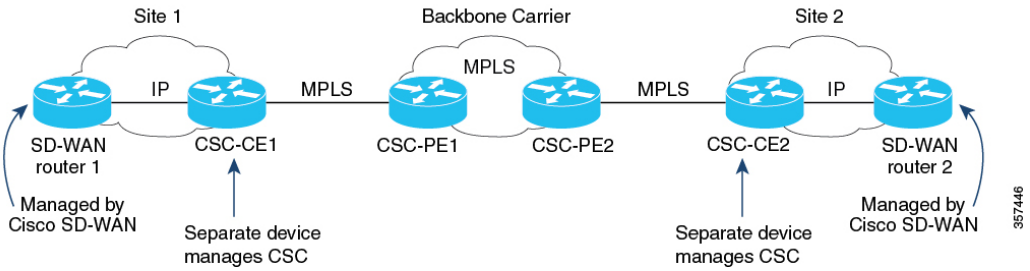
Outlines the transition of Catalyst SD-WAN architectures from dual- to single-device deployments within Carrier Supporting Carrier scenarios, emphasizing operational simplification and efficiency gains.

The Cisco IOS XE Catalyst SD-WAN software simplifies carrier supporting carrier network topologies by enabling a single device to perform both SD-WAN edge and CSC-CE functions.

Before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a

In releases earlier than 17.6.1a, each site in a CSC network topology used two separate devices: an edge device managed by Cisco Catalyst SD-WAN and a dedicated CSC-CE device. This is because the Cisco IOS XE Catalyst SD-WAN device could not function as a CSC-CE. See the illustration for the CSC topology before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a.

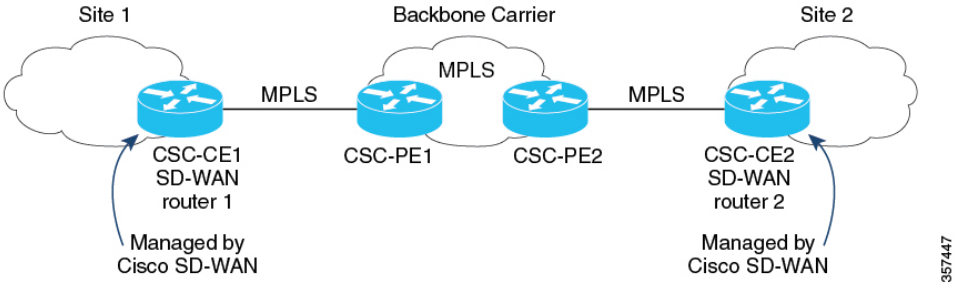
Figure 2: Carrier supporting carrier with Cisco Catalyst SD-WAN, before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a



From Cisco IOS XE Catalyst SD-WAN Release 17.6.1a

From Cisco IOS XE Catalyst SD-WAN Release 17.6.1a, a Cisco IOS XE Catalyst SD-WAN device can function as a CSC-CE, removing the need for a separate dedicated CSC-CE device. See the illustration for the simplified CSC topology with Cisco IOS XE Catalyst SD-WAN devices providing CSC-CE functionality.

Figure 3: Carrier supporting carrier with Cisco Catalyst SD-WAN, Cisco IOS XE Catalyst SD-WAN Release 17.6.1a and later



Use cases for carrier supporting carrier

Presents practical use cases for Carrier Supporting Carrier, demonstrating scenarios where multi-carrier support, scalability, and seamless service provisioning are critical.

Carrier Supporting Carrier (CSC) enables secure, private transport of multiple customer networks over a shared backbone.

Global organizations

Global organizations can use Cisco Catalyst SD-WAN to support CSC with a backbone carrier, enabling multiple, separate divisions of an organization to maintain private traffic while sharing a common backbone carrier.

Service providers

Service providers that implement a CSC topology can benefit from Cisco Catalyst SD-WAN, as it allows carrier edge devices to handle CSC functionality without requiring a separate device.

Configure carrier supporting carrier

Guides configuration of Carrier Supporting Carrier implementations through both feature template and CLI-based workflows, providing step-by-step setup to ensure effective service deployment.

Use one of these methods to configure carrier supporting carrier.

- Feature template

- CLI commands

Configure carrier supporting carrier using a feature template

Follow these steps to configure a CE device for CSC using a new feature template in Cisco SD-WAN Manager.

Procedure

1. Create a new device template.
   a. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b. Choose **Device Templates**, and click **Create Template**.
   c. From the drop-down list, select **From Feature Template**.

   > **Note**
   >
   > In Cisco vManage Release 20.7.x and earlier releases, **Device Templates** is titled as **Device**.

2. Enter device details.
   - In the **Device Model** field, choose the correct device model.
   - In the **Device Role** field, select **SDWAN Edge**.
   - In the **Template Name** field, enter a name for the template.

3. Configure VPN settings.
   a. In the **Transport & Management VPN** section, under **Cisco VPN 0,** choose a template to configure VPN 0.

      For more, see Configure Interfaces in the WAN Transport VPN (VPN 0).
   b. In the **Cisco VPN Interface Ethernet** field, choose a template to configure the interface.

      For more information, see Configure VPN Ethernet Interface.

4. In the **Transport & Management VPN** section, click **Cisco BGP** to add the Cisco BGP field.

   For more information, see see Configure BGP Using SD-WAN Manager Templates.

5. In the **MPLS Interface** section, under **Interface Name 1**, enter the interface used to connect the device to the backbone carrier.
6. In the **Neighbor** section, click **Advanced Options** to configure CSC options.

Table 28: Configure CSC options

Field	Description
Send label	Choose **On** to enable CSC support.
Explicit null	Choose **On** if the device uses a loopback WAN interface.

Field	Description
As override	Choose **On** if CE1 and CE2 use the same autonomous system (AS) number.
Allowas in	Choose **On** if the two CE sites use the same AS number.

7. Click **Save** to save the BGP configuration.
8. Click **Create** to create the feature template.

   The **Configuration > Templates** page displays the available templates.
9. Attach the template to a device.
   a. On the **Configuration** > **Templates** page, click **Device Templates**.
   b. For the new template, click **...** and choose **Attach Devices**.
   c. Move a device to the **Selected Devices** column and click **Attach**.

Configure carrier supporting carrier using CLI

You can use the BGP feature template to configure CSC instead of CLI commands.

Before you begin

Apply a BGP configuration to a Cisco IOS XE Catalyst SD-WAN device before you configure it for CSC-CE functionality.

Follow these steps to configure a CE device for CSC using the CLI.

Procedure

1. Configure CSC-CE1
   a) Map MPLS labels to VRFs.

      The device checks the MPLS label of incoming traffic and uses the IP lookup table of the VRF mapped to that label.For example, if the MPLS label 10 maps to VRF 1, the router uses the IP lookup table of VRF 1 for traffic with label 10.

```
device# config-transaction
device(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
device(config)# mpls label range min-label max-label static min-static-label
 max-static-label
```

   b) Enable MPLS on the interface.

```
device(config)# interfaceinterface
device(config-if)# mpls bgp forwarding
```

   c) Configure BGP.

```
device(config)# router bgp bgp-number
device(config-router)# neighbor neighbor-ip> allowas-in
```

d) Advertise MPLS labels when using a loopback WAN interface.

```
device(config-router)# neighbor <neighbor-ip> send-label explicit-null
```

> **Note**
>
> Using `send-label explicit-null` on non-loopback interfaces does not affect performance.

2. Configure CSC-CE2

   a. Map MPLS labels to VRFs.

```
device# config-transaction
device(config)# mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
device(config)# mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
device(config)# mpls label range min-label max-label static min-static-label
 max-static-label
```

   b. Enable MPLS on the interface.

```
device(config)# interface interface
device(config-if)# mpls bgp forwarding
```

   c. Configure BGP.

```
evice(config)# router bgp bgp-number
Device(config-router)# neighbor neighbor-ip as-override
Device(config-router)# neighbor neighbor-ip send-label explicit-null
```

The following example shows CSC-CE1 and CSC-CE2 configurations with BGP and MPLS:

- CSC-CE1: 10.1.1.10
- CSC-CE2: 10.1.1.20
- CSC-PE1 (neighbor of CSC-CE1): 10.2.2.10
- CSC-PE2 (neighbor of CSC-CE2): 10.2.2.20

CSC-CE1 Configuration

```
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
mpls label range 100000 1048575 static 16 99

interface GigabitEthernet2
 no shutdown
 mpls bgp forwarding
 ip address 10.1.1.15 255.255.255.0

router bgp 10
 bgp log-neighbor-changes
 bgp router-id 172.16.255.15
```

```
 neighbor 10.1.1.20 remote-as 100
 neighbor 10.1.1.20 fall-over bfd
 address-family ipv4 unicast
  maximum-paths 4
  neighbor 10.1.1.20 activate
  neighbor 10.1.1.20 advertisement-interval 30
  neighbor 10.2.2.10 allowas-in
  neighbor 10.2.2.10 send-label explicit-null
  neighbor 10.1.1.20 send-community both
 exit-address-family
 timers bgp 60 180
```

CSC-CE2 Configuration

```
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
mpls label mode all-vrfs protocol bgp-vpnv6 per-vrf
mpls label range 100000 1048575 static 16 99

interface GigabitEthernet5
 ip address 10.0.6.11 255.255.255.0
 negotiation auto
 mpls bgp forwarding

router bgp 10
 bgp log-neighbor-changes
 bgp router-id 172.16.255.11
 neighbor 10.1.1.10 remote-as 200
 address-family ipv4 unicast
  neighbor 10.1.1.10 activate
  neighbor 10.1.1.10 advertisement-interval 30
  neighbor 10.2.2.20 as-override
  neighbor 10.2.2.20 send-label explicit-null
  network 10.0.7.0 mask 255.255.255.0
  redistribute connected
  redistribute static
 exit-address-family
```

Verify device configuration for carrier supporting carrier

Provides verification procedures for Carrier Supporting Carrier device configurations, enabling validation of setup integrity and operational readiness in service provider environments.

To verify if a device is correctly configured to reach the remote CSC-CE device using MPLS-labeled routing.

Procedure

1. Run the following command on the device:

   Example

   ```
   show ip route remote-csc-ce-device-address
   ```

2. Confirm if the output displays a routing entry for the remote site IP address.
3. Check if the output includes one or more routing descriptor blocks that describe the next-hop addresses for the path to the remote CSC-CE device.
4. Ensure that each descriptor block contains an MPLS label.

- If the device is configured correctly, the output shows:

```
Device# show ip route 10.0.1.100
Routing entry for 10.0.1.0/24
...
Routing Descriptor Blocks:
* 10.1.1.100, from 10.1.1.100, 00:00:50 ago
...
MPLS label: 26
```

- If the device is not configured correctly, the output shows:

```
% Subnet not in table
```

6 DHCP Vendor Option Support

Topics:

- Feature history for DHCP vendor option support
- DHCP vendor option support
- Configure a DHCP vendor option using CLI command
- Configure a DHCPv6 client option option using a CLI template
- Configure DHCP server using templates

Details DHCP vendor option support, providing reference information essential for understanding vendor-specific DHCP functionality.

Feature history for DHCP vendor option support

Explains DHCP vendor option support, offering additional reference material to clarify technical aspects and operational behavior.

Table 29: Feature History Table

Feature Name	Release Information	Description
DHCP Option Support	Cisco IOS XE Catalyst SD-WAN Release 16.12.1b	This feature allows DHCP server options, 43 and 191 to configure vendor-specific information in client-server exchanges.
DHCP vendor option support	Cisco IOS XE Catalyst SD-WAN Release 17.10.1a Cisco Catalyst SD-WAN Control Components Release 20.10.1	This feature allows DHCP client option 124 and option 125 to configure vendor-specific information in client-server exchanges. Configure this feature using the CLI Add-on feature template in Cisco SD-WAN Manager.

DHCP vendor option support

Outlines support for configurable DHCP client vendor options, including use of custom identifiers, lease time suggestions, and configuration of vendor-class and vendor-specific information with Options 124/125 (IPv4) and 16/17 (DHCPv6), enabling device classification and flexible feature enablement.

A configurable dynamic host configuration protocol (DHCP) client is a network client feature that:

- allows use of user-specified client or class identifiers,
- supports suggestion of lease time during address requests, and
- enables configuration of vendor-class and vendor-specific information with Option 124 and Option 125.

This functionality provides flexibility for DHCP clients when requesting addresses and enables differential services, device classification, and feature enablement.

DHCP client options

DHCPv6 Client Options

The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 124—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 125—This option is used by DHCP clients and servers to exchange vendor-specific information.

These options are used by Zero-Touch Provisioning (ZTP), Cisco Plug-and-Play (PnP), and Identity Services Engine (ISE) to support multiple use cases.

For example, the content of Option 124 is used for device classification and enabling solution-specific features.

The DHCP Vendor Option Support feature introduces new CLI parameters to make Option 124 and Option 125 flexible. You can modify and customize enabling vendor specific options to carry different values for different customer features. The combination of Option 124 and Option 125 enables various features.

The **ip dhcp client vendor-class** command provides flexibility to pack either Device PID or MAC Address of the DHCP client or any user configurable string in option-124. The default behavior for this command is to continue to send device PID when you choose option 124.

This default behavior can be overridden to carry MAC Address in Day 1 configuration mode by explicitly requesting option-125 from the server using the **ip dhcp client vendor-class** command.

By default, Cisco IOS XE DHCP client sends the following data:

Attribute	IPv4 DHCP Option	Default Value
Vendor-Identifying Vendor Class Option	124	PID

Note

The **ip dhcp client vendor-class [mac-address | ascii | disable | hex]** command overrides PID with MAC Address / user defined string / disable Option 124.

Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message.

The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 16—This option is used by DHCP clients and servers to exchange vendor-class information.
- Option 17—This option is used by DHCP clients and servers to exchange vendor-specific information

In DHCPv6, option-16 and option-17 are used by DHCP clients and servers to exchange vendor-specific information.

The **ipv6 dhcp client vendor-class** command provides flexibility to pack either Device PID or MAC Address of the DHCP Client or any user configurable string in option-16.

The default behavior for this command is to continue to send device PID when you choose option-16 but it can be overridden to carry MAC Address in Day 1 configuration mode using the **ipv6 dhcp client vendor-class** command.

By default, Cisco IOS XE DHCP client sends the following data:

Attribute	IPv4 DHCP Option	Default Value
Vendor Class Option	16	PID

> **Note**
>
> The **ipv6 dhcp client vendor-class [mac-address | hex | ascii | disable]** command can be used to override default value of PID with MAC Address or User defined string or disable the option.

Configure a DHCP vendor option using CLI command

Guides users through the process of configuring a DHCP vendor option using CLI templates, including required commands and verification steps to ensure correct implementation.

This section provides a sample CLI configuration to configure a DHCP vendor option.

For more information about using CLI templates, see CLI Add-On Feature Templates.

Before you begin

By default, CLI templates execute commands in global config mode.

Procedure

1. Configure an interface type and enter the interface configuration mode.

   Example

   ```
   interface type number
   ```

2. Acquire an IP address on an interface from DHCP.

   Example

   ```
   ip address dhcp
   ```

3. Configure the DHCP vendor-class option.

   Example

   ```
   ip dhcp client vendor-class [mac-address | ascii | hex | disable]
   ```

   > **Note**
   >
   > You must first configure the **no ip dhcp-client** command before configuring the IP address.

Override the device PID with MAC address:

The DHCP vendor-class option overrides the device PID with the MAC Address.

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class mac-address
    !
```

Override the device PID with user defined string in hex or in ascii format:

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class hex aabbcc
    !

interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class ascii cisco
    !
```

Disable option-124 in DHCP messages:

```
interface GigabitEthernet 0/0/0
    ip address dhcp
    ip dhcp client vendor-class disable
    !
```

Configure a DHCPv6 client option option using a CLI template

Details the steps to configure DHCP client vendor options using CLI templates, with instructions on command usage and validation procedures for successful configuration.

By default, CLI templates execute commands in global config mode.

For more information about using CLI templates, see *CLI Add-On Feature Templates* in the *Cisco Catalyst SD-WAN Systems and Interfaces Configuration Guide, Cisco IOS XE Catalyst SD-WAN Release 17.x*.

The section provides a sample CLI configuration to configure DHCPv6 client option.

Procedure

1. Configure an interface type and enter the interface configuration mode.

   Example

   ```
   interface type number
   ```

2. Acquire an IPv6 address on an interface from DHCP.

   Example

   ```
   ipv6 address dhcp
   ```

3. Configure the DHCP vendor-class option.

Example

```
ipv6 dhcp client vendor-class {mac-address | ascii | hex | disable}
```

Note

By default, the DHCPv6 client carries the device PID of the device in option-16.

This default behaviour can be overridden by configuring the **ipv6 dhcp client vendor-class** command.

Override the device PID with the MAC address:

The DHCP vendor-class option overrides the Device PID with the MAC Address.

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class mac-address
    !
```

Override the device PID with user defined string in hex or in ascii format:

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class hex aabbcc
    !

interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class ascii cisco
    !
```

Disable option-16 in DHCP messages:

```
interface GigabitEthernet 0/0/0
    ipv6 address dhcp
    ipv6 dhcp client vendor-class disable
    !
```

Configure DHCP server using templates

Outlines procedures to configure DHCP servers with CLI templates, providing instructions for template selection, command execution, and verification of server configuration.

To configure a Cisco IOS XE Catalyst SD-WAN device to act as a DHCP server using Cisco SD-WAN Manager templates:

1. Create a DHCP-Server feature template to configure DHCP server parameters, as described in this topic.
2. Create one or more interface feature templates, as described in the VPN-Interface-Ethernet and the VPN-Interface-PPP-Ethernet sections.
3. Create a VPN feature template to configure VPN parameters.

To configure a Cisco IOS XE Catalyst SD-WAN device interface as a DHCP helper to broadcast DHCP requests from DHCP servers, in the **DHCP Helper** field of the applicable interfaces template, enter the addresses of the DHCP servers.

Use the DHCP-Server template for all Cisco Catalyst SD-WANs.

You enable DHCP server functionality on a Cisco IOS XE Catalyst SD-WAN device interface so it can assign IP addresses to hosts in the service-side network.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Device Templates**, and then click **Create Template**.

   In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

   a) From the **Create Template** drop-down list, choose **From Feature Template**.
   b) From the **Device Model** drop-down list, select the type of device for which you are creating the template.
   c) Click **Service VPN** or scroll to the **Service VPN** section.
   d) From **Additional VPN Templates**, click **VPN Interface**.
   e) From the **Sub-Templates** drop-down list, choose DHCP Server.
   f) From the **DHCP Server** drop-down list, click **Create Template**. The DHCP-Server template form is displayed.

      This form contains fields for naming the template, and fields for defining the DHCP Server parameters.

   g) In **Template Name**, enter a name for the template.

      The name can be up to 128 characters and can contain only alphanumeric characters.

   h) In **Template Description**, enter a description of the template.

      The description can be up to 2048 characters and can contain only alphanumeric characters.

   When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the **Scope** drop-down list.

3. Configure DHCP basic server functionality.

   Table 30:

Parameter Name	Description
Address Pool*	Enter the IPv4 prefix range, in the format *prefix/length*, for the pool of addresses in the service-side network for which the router interface acts as DHCP server.
Exclude Addresses	Enter one or more IP addresses to exclude from the DHCP address pool. To specify multiple individual addresses, list them separated by a comma. To specify a range of addresses, separate them with a hyphen.
Maximum Leases	Specify the number of IP addresses that can be assigned on this interface. Range: 0 through 4294967295
Lease Time	Specify how long a DHCP-assigned IP address is valid. Range: 0 through 4294967295 seconds

Parameter Name	Description
Offer Time	Specify how long the IP address offered to a DHCP client is reserved for that client. By default, an offered IP address is reserved indefinitely, until the DHCP server runs out of addresses. At that point, the address is offered to another client. Range: 0 through 4294967295 seconds Default: 600 seconds
Administrative State	Select Up to enable or Down to disable the DHCP functionality on the interface. By default, DHCP server functionality is disabled on an interface.

4. Configure a static lease to assign a static IP address to a client device on the service-side network.
   Table 31:

Parameter Name	Description
MAC Address	Enter the MAC address of the client to which the static IP address is being assigned.
IP Address	Enter the static IP address to assign to the client.
Hostname	Enter the hostname of the client device.

   To edit a static lease, click **pencil** icon.

   To remove a static lease, click **trash** icon.

   To save the feature template, click **Save**.

5. Configure advanced DHCP server options
   Table 32:

Parameter Name	Description
Interface MTU	Specify the maximum MTU size of packets on the interface. Range: 68 to 65535 bytes
Domain Name	Specify the domain name that the DHCP client uses to resolve hostnames.
Default Gateway	Enter the IP address of a default gateway in the service-side network.
DNS Servers	Enter one or more IP address for a DNS server in the service-side network. Separate multiple entries with a comma. You can specify up to eight addresses.

Parameter Name	Description
TFTP Servers	Enter the IP address of a TFTP server in the service-side network. You can specify one or two addresses. If two, separate them with a comma.

7 Dynamic On-Demand Tunnels

Topics:

- Feature history for dynamic on-demand tunnels
- Dynamic on-demand tunnels
- How on-demand tunnels work
- How on-demand tunnels work with a transport gateway
- Prerequisites for on-demand tunnels
- Restrictions for on-demand tunnels
- Configure on-demand tunnels
- Monitor the status of on-demand tunnels
- View OMP routes

Introduces dynamic on-demand tunnel technology, summarizing its role, core concepts, and integration within Cisco SD-WAN solutions.

Feature history for dynamic on-demand tunnels

Details the evolution and key feature updates associated with dynamic on-demand tunnels, highlighting version-specific enhancements and changes.

Table 33: Feature History Table

Feature Name	Release Number	Description
Dynamic On-Demand Tunnels	Cisco IOS XE Catalyst SD-WAN Release 17.3.1a Cisco SD-WAN Release 20.3.1	This feature enables you to configure an inactive state for tunnels between edge devices. This configuration reduces performance demands on devices and decreases network traffic.
Dynamic On-Demand Tunnels with Transport Gateways	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	A transport gateway can serve as the hub between two spoke devices. It provides the backup route that is necessary for spoke-to-spoke on-demand tunnels to operate. Using a transport gateway as the hub simplifies the process of enabling on-demand tunnels. This method does not require any changes to control policy on Cisco SD-WAN Controllers.

Dynamic on-demand tunnels

Explains the purpose, capabilities, and operational context of dynamic on-demand tunnels, providing foundational knowledge for deployment and management.

A Cisco Catalyst SD-WAN on-demand tunnel is a network feature that:

- automatically establishes secure tunnels between spoke devices when traffic starts,
- uses a configurable inactivity timer to remove tunnels after traffic stops, and
- conserves bandwidth and device performance when inactive.

On-demand tunnels enable efficient, temporary connections that optimize network resource usage.

Cisco Catalyst SD-WAN on-demand tunnels are triggered by device traffic and are removed after a set period of inactivity, ensuring inactive links do not consume bandwidth or affect performance.

How on-demand tunnels work

Describes the mechanisms and workflow of on-demand tunnels, clarifying operational sequences and tunnel establishment within the network.

Summary

When you configure a site to use dynamic tunnels, the on-demand functionality is enabled. In this mode of operation, Cisco Catalyst SD-WAN edge routers do not bring up direct tunnels to other sites that are also enabled with on-demand functionality.

Cisco Catalyst SD-WAN selects one or more edge routers (typically centrally located routers) to act as backup forwarding node(s), providing a secondary path for traffic between two nodes. The backup node(s) are not enabled for on-demand. All on-demand sites form static tunnels with the backup node(s). The backup node(s) provide a static backup route for traffic between two nodes that have on-demand enabled.

The first packet of traffic between two nodes is routed through the static backup path, and triggers the on-demand tunnel to become active between the sites. The backup path continues to forward traffic until the direct path becomes active.

All on-demand sites learn the TLOCs and prefixes of all other on-demand remote sites. The prefixes also have a backup path set up through Cisco Catalyst SD-WAN Controller control policy. So in the control plane, the on-demand tunnel network has the same state as a full-mesh tunnel network, including a backup path.

The control plane downloads to the data plane, routes, with the backup path and remote TLOCs that represent a potential direct path between any two sites, but it does not set up a direct path tunnel to remote TLOCs.

All prefixes learned from remote sites must have a backup path. A less-specific aggregate route from a hub site is not a valid backup path. The specific prefix advertised from the remote branch must also be advertised as a backup from the hub.

If this specific prefix is missing as a backup, the on-demand tunnel setup will fail.

Therefore, the backup path must be a static tunnel that is always UP. This static tunnel should include both the specific prefixes from the remote branch and any necessary aggregate routes to ensure proper tunnel establishment and maintenance.

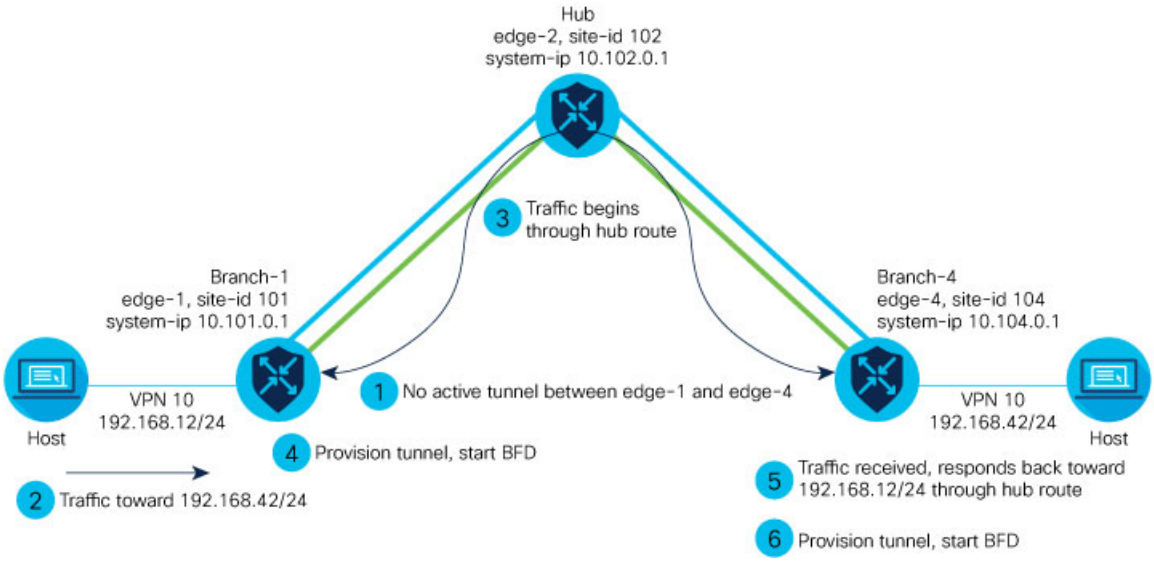
Traffic from either end of the on-demand tunnel triggers setting up the tunnel. This enables on-demand tunnels to accommodate network address translation (NAT) traversal.

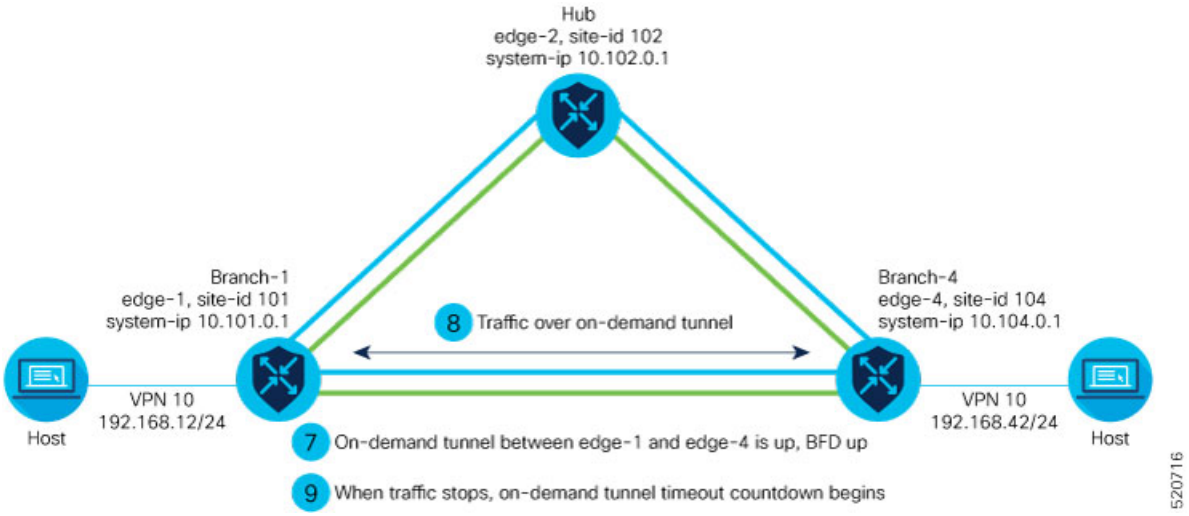
The on-demand tunnel feature introduces two states for the on-demand branch site:

- Inactive: The on-demand tunnel is not set up with the remote site. There is no active traffic to or from the remote site. Remote site TLOCs are inactive—no bidirectional forwarding detection (BFD) is set up, the prefix is installed with the inactive paths, and the backup path is set as the path to forward any traffic.
- Active: The on-demand direct site-to-site tunnel is set up to the remote site. There is active traffic to or from the remote site. This state is identical to the case of a typical tunnel, where the remote TLOCs have BFD set up, and the prefix is installed with the direct path tunnel. In this state, tunnel activity is tracked. If there is no traffic for the “idle time” duration (default 10 minutes), the direct site-to-site tunnel is removed and the state changes to Inactive.

Workflow

Figure 4: Process of On-Demand Tunnel Establishment Between Two Edge Routers





The steps below demonstrate what occurs between two edge routers with an on-demand tunnel configured.

1. There is no active tunnel between the two edge routers. edge-1 and edge-4 are in their inactive states.
2. The host behind edge-1 initiates traffic toward the host behind edge-4.
3. edge-1 forwards the traffic through the backup path using the hub or backup node to edge-4.
4. edge-1 provisions the on-demand tunnel and begins bidirectional forwarding detection (BFD). edge-4 is now in its active state on edge-1.
5. When edge-4 receives the return traffic for the host behind edge-1, it forwards the traffic through the backup path using the hub or backup node to edge-1.
6. edge-4 provisions the on-demand tunnel and begins BFD. edge-1 is now in active state on edge-4.
7. At this point, the on-demand tunnel between edge-1 and edge-4 is up, and BFD is up.
8. Traffic between the two edge devices takes the direct route through the on-demand tunnel.
9. Both edge-1 and edge-4 track the traffic activity on the on-demand tunnel in both directions.

   If there is no traffic for the idle timeout duration, the on-demand tunnel is deleted, and the edge-1 and edge-4 devices go back to the inactive state.

How on-demand tunnels work with a transport gateway

Outlines the processes involved when using on-demand tunnels alongside a transport gateway, addressing gateway interactions and associated workflows.

Summary

A transport gateway can serve as the hub between two spoke devices, providing the backup route that is necessary for spoke-to-spoke on-demand tunnels to operate. Using a transport gateway as the hub simplifies the process of enabling on-demand tunnels. This method does not require configuring control policy on Cisco SD-WAN Controllers.

Workflow

For information about configuration, see *Configure On-Demand Tunnels Using a Transport Gateway*.

Prerequisites for on-demand tunnels

Details essential requirements and system conditions needed before deploying dynamic on-demand tunnels, ensuring a supported environment for operation.

There are several prerequisites for using on-demand tunnels:

- Configure a Centralized Control Policy for On-Demand Tunnels
- Prerequisites: OMP Settings
- Prerequisites: Hub Device Traffic Engineering Service
- Prerequisites: Spoke Device ECMP Limit
- Prerequisites: OMP Settings
- Prerequisites: Hub Device Traffic Engineering Service
- Prerequisites: Spoke Device ECMP Limit

Prerequisites: OMP settings

When on-demand tunnels are enabled, spokes use backup paths through the hub, so a higher path limit is necessary. The direct paths as well as the backup paths need to be advertised. To accommodate this, increase the Cisco Catalyst SD-WAN Controller send-path-limit to advertise all available paths. We recommend to use the maximum possible value.

> **Note**
>
> If there are too many Hub TLOCs configured in the on-demand tunnel control policy, the recommended value for send-path-limit is not enough always. In such cases, the on-demand tunnel feature will not work at all.
>
> Starting from Cisco vManage Release 20.8.1 and Cisco IOS XE Catalyst SD-WAN Release 17.8.1a, the maximum send-path-limit is 32. In Cisco vManage Release 20.7.x and earlier releases, the maximum send-path-limit is 16.

For information about configuring Cisco SD-WAN Controller send-path-limit, see the routing configuration guides on the Cisco Catalyst SD-WAN Configuration Guides page.

Before you begin

The Cisco Catalyst SD-WAN Controller send-path-limit must be more than the default 4.

Procedure

1. Configure the OMP send path limit using a feature template

   For a Cisco Catalyst SD-WAN Controller in Managed mode, use this procedure to configure the OMP send path limit.

   To confirm that it is in Managed mode, from the Cisco SD-WAN Manager menu, choose **Configuration** > **Control Components**. For a **Controller** row, the **Managed By** column shows a template name.

   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Feature Templates.** In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.
   c) Edit or create an OMP template for device type Controller. In some earlier releases, the device type is called vSmart. Any functioning Cisco Catalyst SD-WAN deployment has at least one vSmart OMP template configured.

d) In **Basic Configuration**, set the **Number of Paths Advertised per Prefix** to 16 (recommended). From Cisco SD-WAN Release 20.8.1, the maximum is 32.

2. Configure the OMP send path limit for an edge device using CLI commands

   Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

   For an edge device, use these commands to configure an OMP send path limit.

   a) Enter OMP configuration mode.
   b) Set the send path limit. We recommend 16.

      From Cisco SD-WAN Release 20.8.1, the maximum is 32.

   Example

```
sdwan
omp
 send-path-limit 16
```

Prerequisites: Hub device traffic engineering service

Before you begin

On the hub device, the Traffic Engineering service (service TE) must be enabled.

This ensures that the Cisco Catalyst SD-WAN Overlay Management Protocol (OMP) on the spoke devices accepts the backup path through the hub, which is being added as an intermediate path between the two spoke devices. Without this, the backup path through the hub would be considered invalid and unresolved by the spoke devices.

Procedure

1. Enable the Traffic Engineering Service Using Cisco SD-WAN Manager
   a) In Cisco SD-WAN Manager, open **Configuration** > **Templates**
   b) Click **Feature Templates**.

      In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

   c) Click **Add Template**.
   d) Select a platform.
   e) From **VPN**, select **VPN**.
   f) Ensure that in **Basic Configuration**, the **VPN** field is set to 0.
   g) From **Service**, click **New Service** and select **TE**.
   h) Click **Add**, and then click **Update**. The TE service appears in the table of services.
   i) Apply the VPN-0 template to the hub
2. Enable the Traffic Engineering Service, Using CLI Commands (Cisco IOS XE Catalyst SD-WAN Devices)

   Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Example

```
sdwan
 service TE vrf global
exit
```

3. Enable the Traffic Engineering Service, Using CLI Commands (Cisco vEdge Devices)

Example

```
vpn 0
 service TE
 exit
```

Prerequisites: Spoke Device ECMP Limit

Before you begin

On spoke devices, the ECMP limit must be more than the default 4. Recommended: 16

When on-demand tunnels are enabled, spoke devices create both direct and backup paths. To accommodate the need for more paths, increase the ECMP limit.

Procedure

1. Configure the ECMP Limit Using a Configuration Group

   On the **Configuration** > **Configuration Groups** page, choose the SD-WAN solution type.

   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
   b) Do one of these:
      - Edit a profile directly:

        In the System Profile tab, create (**Add New)** or edit a System profile.

      - Edit a profile in a configuration group:

        Open a configuration group and edit the System profile.

   c) In the system profile, create or edit an OMP feature.
   d) In the **Basic Configuration** section, use the **ECMP Limit** field to configure the equal-cost multi-path (ECMP) limit as 16 (recommended).

2. Configure the ECMP Limit Using a Feature Template
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Feature Templates**.

      In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

   c) Click **Add Template**.
   d) Select a device and click **Cisco OMP**.
   e) In **Basic Configuration**, set the **ECMP Limit** field to 16 (recommended).

3. Configure the ECMP Limit Using CLI Commands

Use a CLI add-on profile, CLI add-on feature template, or CLI template to execute these CLI commands. By default, CLI templates and the CLI add-on profile execute commands in global configuration mode. For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Example

```
sdwan
omp
 no shutdown
 ecmp-limit       16
```

You can view the current ecmp-limit using the `show running-config omp` command.

Restrictions for on-demand tunnels

Outlines operational limitations, constraints, and unsupported features relevant to dynamic on-demand tunnels in Cisco SD-WAN deployments.

- PfR statistics collection starts fresh for each on-demand tunnel setup and does not cache statistics for deleted tunnels after idle timeout.
- Out-of-order (OOO) packets may occur when the router switches traffic from the backup path to the on-demand tunnel; the router forwards packets as received.
- Unidirectional and multicast flows do not trigger on-demand tunnel setup and continue to use the backup path.
- Do not configure a data policy that applies a **set tloc-list** action to an on-demand site TLOC; doing so will result in dropped traffic.
- If the Pair Wise Key (PWK) IPSEc feature is enabled, on-demand tunnels are not supported.
- All TLOCs in the system are reset (disabled and then enabled) when you execute **on-demand enable** or **no on-demand enable**.
- When provisioning on-demand tunnels, the edge device provisions tunnels to all TLOCs on the remote edge device; for multi-home sites, you must enable on-demand mode on all systems.
- The system keeps all edge devices using on-demand tunnels active if service or user traffic exists on any on-demand tunnel in either direction; tunnels can be enabled between two sites only if both are in on-demand mode.
- You must configure a backup path for all prefixes from on-demand remote sites; the backup path must always be UP. The setup or removal of on-demand tunnels does not affect overlay route (OMP) updates or service/LAN-side route updates (such as OSPF or BGP). If either site is not in on-demand mode, the system sets up static tunnels between the sites.

Configure on-demand tunnels

Guides configuration of dynamic on-demand tunnels, including setup using control policies, centralized control policy methods, and configuration via transport gateways, covering both group and template-based approaches.

The following procedures describe how to configure on-demand tunnels using different methods, including using control policy, or a simpler method using a transport gateway as a hub.

- Configure On-Demand Tunnels Using Control Policy
- Configure On-Demand Tunnels Using a Transport Gateway

- Enable On-Demand Tunnels on a Spoke Device Using a Configuration Group
- Enable On-Demand Tunnels on a Spoke Device Using a Template
- Enable On-Demand Tunnels Using a CLI Template

Configure on-demand tunnels using control policy

To configure on-demand tunnels using the control policy method, do the following:

Procedure

1. Configure a control policy, as described in Configure a centralized control policy for on-demand tunnels.
2. Enable on-demand tunnels n spoke devices, as described in Enable on-demand tunnels on a spoke device using a template and Enable on-demand tunnels using a CLI template.

Configure a centralized control policy for on-demand tunnels

Before you begin

This procedure configures a centralized control policy on a Cisco Catalyst SD-WAN Controller to enable on-demand tunnels.

- The Cisco Catalyst SD-WAN Controller centralized control policy must include the `tloc-action backup` action.

  This ensures that the backup path through the hub for communication between all of the spoke devices.
- The Cisco Catalyst SD-WAN Controllerr centralized control policy must accept all spoke prefix routes.
- The Cisco Catalyst SD-WAN Controller centralized control policy must accept TLOCs of all spokes.

  For information about configuring a Cisco Catalyst SD-WAN Controller **centralized control policy,** see the policies configuration guides on the Cisco Catalyst SD-WAN Configuration Guides page.
- When configuring on-demand tunnels using a transport gateway, do not use the control policy procedure described here. For information, see Configure On-Demand Tunnels Using a Transport Gateway.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
   a) Select **Centralized Policy**.
   b) Click **Add Policy**.
   c) In the left pane, click **Site**.
   d) Click **Next**.
   e) Click **Add Topology** and select **Custom Control (Route & TLOC)**.
   f) Enter a name and description for the topology.
   g) Click **Sequence Type**.
   h) In the **Add Control Policy** pop-up window, choose **Route**.
   i) Click **Sequence Rule** to create a sequence.
2. Click **Match**.
   a) Among the match conditions, click **Site**.
   b) In the **Match Conditions** area, click the **Site List** menu and choose a site list.
   c) Click **Actions**, and then **Accept**.

3. Among the actions, click **TLOC Action**.
   a) In the **Actions** area, click the **TLOC Action** menu and choose **Backup**.
   b) Among the actions, click **TLOC**.
   c) In the **Actions** area, click the **TLOC List** menu and choose or create a TLOC list.
   d) Click **Save Match and Actions**.
4. Click **Default Action**.
   a) In the **Default Action** area, click the pencil icon to edit.
   b) Near the **Actions** label, click **Accept**.
   c) Click **Save Control Policy**.
   d) Click **Next** twice.
   e) In the Topology tab, click **New Site/WAN Region List**.
   f) Click **Outbound Site List** and choose a site list that defines the sites at which you are enabling on-demand tunnels.
   g) Adjacent to the site list, click **Add**.
   h) Enter a name and description for the policy.
   i) Click **Save Policy**.

Configure centralized control policy for on-demand tunnels using a CLI policy

Before you begin

The Cisco Catalyst SD-WAN Controller must be managed by Cisco SD-WAN Manager.

Procedure

---

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Classic** > **Policies**.
2. Open **Centralized Policy**.
3. From **Custom Options**, choose **Centralized Policy** > **CLI Policy**.
4. Click **Add Policy**.
5. Enter the CLI commands for the policy.

Example

```
control-policy Dynamic-Tunnel-Control-Policy
    sequence 100
     match route
      site-list Branches
     !
     action accept
      set
       tloc-action backup
       tloc-list Hub-TLOCs
      !
     !
     sequence 200
      match tloc
     !
     action accept
    !
  default-action accept
 !
 lists
  site-list Branches
```

```
   site-id 200
   site-id 300
  !
  tloc-list Hub-TLOCs
   tloc 10.0.0.1 color mpls encap ipsec
   tloc 10.0.0.1 color public-internet encap ipsec
!
!
apply-policy
 site-list Branches
  control-policy Dynamic-Tunnel-Control-Policy out
 !
!
```

Configure on-demand tunnels using a transport gateway

Before you begin

- On Cisco SD-WAN Controllers, configure the send path limit, as described in Prerequisites: OMP settings.
- On spoke devices, configure the ECMP limit, as described in Prerequisites: Spoke Device ECMP Limit.
- When using a transport gateway as a hub to support on-demand tunnels, there is no need to create or modify a control policy.

  Do not use the procedure described in Configure a Centralized Control Policy for On-Demand Tunnels.

Procedure

1. Enable transport gateway functionality on a router serving as the hub, providing a backup route between spokes, as described in the Transport Gateway section of the *Cisco Catalyst SD-WAN Routing Configuration Guide*.
2. Enable on-demand tunnels and configure the idle timeout on spoke devices as described in Enable on-demand tunnels on a spoke device using a template.

Enable on-demand tunnels on a spoke device using a configuration group

Before you begin

On the **Configuration** > **Configuration Groups** page, choose the **SD-WAN** solution type.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Do one of these:
   - Edit a profile directly:

     In the System Profile tab, create (**Add New**) or edit a System profile.
   - Edit a profile in a configuration group:

     Open a configuration group and edit the System profile.
3. In the System profile, create (**Add New**) or edit a Basic feature.

4. In the **Advanced** section, use the **On Demand Tunnel** control to enable on-demand tunnels.

Enable on-demand tunnels on a spoke device using a template

Before you begin

- See the Prerequisites for On-Demand Tunnels.
- Do not enable on-demand on the hub device.
- On the spoke devices, enable on-demand at the system level. In the case of multi-homed sites, enable on-demand on all systems at the site.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**
2. Click **Feature Templates**.

> **Note**
>
> In Cisco vManage Release 20.7.x and earlier releases, **Feature Templates** is titled **Feature**.

3. Click **Add Template**.
4. Select a device.
5. From **Basic Information**, select **Cisco System**.
6. Click **Advanced**.
7. Enable **On-demand Tunnel.**
8. (optional) Configure the **On-demand Tunnel Idle Timeout** time. The default idle timeout value is 10 minutes. Range: 1 to 65535 minutes
9. Attach the System feature template to the device template for the spoke device.

Enable on-demand tunnels using a CLI template

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

By default, CLI templates execute commands in global configuration mode.

Before you begin

- See Prerequisites for On-Demand Tunnels.
- Do not enable on-demand on the hub device

Procedure

On the spoke devices, enable on-demand tunnels at the system level. In the case of multi-homed sites, enable on-demand on all systems in the site.

The default idle timeout value is 10 minutes. Range: 1 to 65535 minutes

Example

```
system
     on-demand enable
     on-demand idle-timeout 10
```

Monitor the status of on-demand tunnels

Explains monitoring procedures for dynamic on-demand tunnels, including viewing tunnel status and history in Cisco SD-WAN Manager and examining the route to destination devices.

Procedures for monitoring the status of on-demand tunnels.

- View the current status of on-demand tunnels using Cisco SD-WAN Manager on page 85
- View a chart of the on-demand tunnel status over time in Cisco SD-WAN Manager on page 85
- View the route to a destination device on page 86

View the current status of on-demand tunnels using Cisco SD-WAN Manager

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

   For Cisco Catalyst SD-WAN Control Components Release 20.6.x and earlier:

   From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network**.
2. Select a device.
3. Select **Real Time**.
4. For **Device Options**, select one of the following:
   - **On Demand Local**: Displays the status of on-demand tunnels on the specified device.
   - **On Demand Remote**: Displays the status of on-demand tunnels on the specified device, and on all connected devices.

The output is equivalent to executing the `show [sdwan] system on-demand [remote-system] [system-ip ip-address] CLI` command. It displays the status of on-demand tunnels.

View a chart of the on-demand tunnel status over time in Cisco SD-WAN Manager

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.

   Cisco Cisco Catalyst SD-WAN Control Components Release 20.6.x and earlier: From the Cisco SD-WAN Manager menu, choose **Monitor** > **Network** > > **.**
2. Select a device.
3. From WAN, choose Tunnel.

4. From the **Chart Options** drop-down list, select **On-Demand Tunnel Status.** The chart shows the status of tunnels as `ACTIVE` or `INACTIVE`. `INACTIVE` indicates that an on-demand tunnel is in its inactive mode.

View the route to a destination device

No Active On-Demand Tunnel

Active On-Demand Tunnel

Viewing the route between routers A and B can show whether the route is using an on-demand tunnel. On router A, use the `traceroute` command and enter router B as the destination. The command output shows whether the current route includes a hop at a hub device or whether the route is directly to the destination.

In the following examples, the router IP addresses are as follows:

- Router A: 10.1.1.1
- Router B: 10.1.1.2
- Hub device: 10.100.1.100

In the following example, there is no active on-demand tunnel between routers A and B, so the route includes the hub device. Note that it takes two hops to reach router B.

```
RouterA#traceroute vrf 1 10.1.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.1.1.2
VRF info: (vrf in name/id, vrf out name/id)
1 10.100.1.100 10 msec 8 msec 0 msec
2 10.1.1.2 2 msec * 1 msec
```

```
RouterA#traceroute vrf 1 10.1.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.1.1.2
VRF info: (vrf in name/id, vrf out name/id)
1 10.1.1.2 1 msec
```

In the following example, there is an active on-demand tunnel between routers A and B, so the route from router A and to router B is direct.

View OMP routes

Describes procedures for viewing and verifying OMP routes associated with dynamic on-demand tunnels, supporting network validation and troubleshooting activities.

Viewing OMP routes can show the status of on-demand tunnels between two routers. Use the show `sdwan omp routes` command and view the **STATUS** column. The following table shows the possible values for this column, depending on whether an on-demand tunnel is active or not between two routers:

Table 34: Status of Routes, with or without an Active On-Demand Tunnel Between Two Routers

On-Demand Tunnel Between Routers A and B	STATUS for OMP Routes Between Routers A and B	STATUS for Backup Routes (through the Hub)
Not active	I, U, IA (installed, unresolved, and inactive)	C, I, R (chosen, installed, and resolved)

On-Demand Tunnel Between Routers A and B	STATUS for OMP Routes Between Routers A and B	STATUS for Backup Routes (through the Hub)
Active	C, I, R (chosen, installed, and resolved)	R (resolved)

8 GRE-in-UDP

Topics:

- Feature history for GRE-in-UDP
- GRE-in-UDP
- Configure GRE-in-UDP using CLI commands

Introduces GRE-in-UDP, providing a reference overview of its protocol encapsulation and use cases within network infrastructure.

Feature history for GRE-in-UDP

Details the feature history for GRE-in-UDP, highlighting changes, enhancements, and support introductions across software releases.

Table 35: Feature History

Feature Name	Release Information	Description
GRE-in-UDP	Cisco IOS XE Catalyst SD-WAN Release 17.11.1a Cisco vManage Release 20.11.1	You can configure GRE encapsulation for UDP transport.

GRE-in-UDP

Explains GRE-in-UDP concepts, describing protocol functionality, encapsulation techniques, deployment scenarios, and operational benefits for flexible tunneling solutions.

A GRE-in-UDP protocol is a network protocol that

- encapsulates Generic Routing Encapsulation (GRE) tunnels within a User Datagram Protocol (UDP) packet to facilitate load balancing and improve performance over networks, and
- enables a router to encapsulate GRE packets—including source and destination port information—within a UDP header, send the UDP packet through the tunnel, and allow the destination device to de-encapsulate the UDP packet.

Cisco Catalyst SD-WAN supports generic routing encapsulation (GRE) with UDP for IPv4 and IPv6 traffic.

Configure GRE-in-UDP using CLI commands

Provides step-by-step instructions for configuring GRE-in-UDP through CLI commands, including setup procedures, example configurations, and verification methods to ensure proper tunnel operation.

Follow these steps to configure GRE-in-UDP using a CLI template.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*. By default, CLI templates execute commands in global config mode.

GRE-in-UDP is a protocol that encapsulates Generic Routing Encapsulation (GRE) tunnels within a User Datagram Protocol (UDP) packet to facilitate load balancing and improve performance over networks.

Procedure

1. For the desired interface, enter interface configuration mode.

```
sdwan
interface interface
```

2. Enter tunnel interface mode.

```
tunnel-interface
```

3. Configure GRE encapsulation.

```
encapsulation gre
```

4. Configure GRE-in-UDP as the encapsulation mode.

```
gre-in-udp
```

Here is a complete example of configuring GRE-in-UDP.

```
interface GigabitEthernet1
  tunnel-interface
   encapsulation gre
   color lte
   gre-in-udp
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   no allow-service stun
   allow-service https
   no allow-service snmp
   no allow-service bfd
  exit
```

9 IP DHCP Smart-Relay

Topics:

- Feature history for IP DHCP smart-relay
- IP DHCP smart-relay
- Benefits of IP DHCP smart-relay
- Configure IP DHCP smart-relay agent using a CLI template

Describes the IP DHCP smart-relay feature, summarizing key operational characteristics and technical specifications for relaying DHCP packets within IP networks.

Feature history for IP DHCP smart-relay

Details the feature history of IP DHCP smart-relay, highlighting major enhancements, platform support, and software release information.

Table 36: Feature History Table

Feature Name	Release Information	Feature Description
IP DHCP Smart-Relay	Cisco IOS XE Catalyst SD-WAN Release 17.13.1a Cisco Catalyst SD-WAN Manager Release 20.13.1	With this feature, you can set the gateway address to the secondary IP address using the DHCP relay agent, when there is no IP address and DHCP service information from the DHCP server. A DHCP relay agent is any host or IP router that forwards DHCP packets between clients and servers. This functionality is useful when the DHCP server cannot be configured to use secondary pools.

IP DHCP smart-relay

Outlines essential concepts of the IP DHCP smart-relay, including operational principles, deployment considerations, and use case scenarios.

A Dynamic Host Configuration Protocol (DHCP) relay agent is a network host that

- forwards DHCP packets between clients and servers on different physical subnets,
- generates new DHCP messages on another interface, setting the gateway IP address and optionally adding relay agent information (option 82), and
- supports unnumbered interfaces by automatically managing static host routes for DHCP clients.

Relay-agent forwarding differs from standard IP routing; instead of transparent packet switching, the relay agent receives DHCP messages, creates new messages with updated information, and forwards these between interfaces.

When a DHCP reply is received from the server, the relay agent removes option 82 (if present) before forwarding it to the client. With unnumbered interfaces, the relay agent borrows an IP address from another configured interface and dynamically adds or removes static host routes for DHCP clients as leases are assigned and released.

Supporting reference information

- DHCP relay agents are essential when DHCP clients and servers are not on the same subnet, as they enable message forwarding across networks.
- Relay agents specifically set the gateway IP address and can insert option 82 for added information in packets sent to the DHCP server.
- The Cisco IOS XE DHCP relay agent supports unnumbered interfaces, which helps conserve address space and network resources. Static host routes for DHCP clients on unnumbered interfaces are created and removed automatically based on the DHCP lease status.

Benefits of IP DHCP smart-relay

Explains the benefits of IP DHCP smart-relay, emphasizing improved flexibility, scalability, and efficiency in managing DHCP traffic across network segments.

- Automatically assigning IP addresses at each remote site substantially reduces internet access costs. Purchasing static IP addresses is considerably more expensive than using automatically allocated IP addresses.
- Simplifies configuration, reduces operational overhead and costs related to device configuration, and makes deployment easier for nontechnical users.
- The DHCP server maintains configurations for several subnets, so an administrator needs to update only a single, central server when configuration parameters change.

Configure IP DHCP smart-relay agent using a CLI template

Provides instructions for configuring the IP DHCP smart-relay agent with a CLI template, including step-by-step procedures to enable and verify the feature on supported devices.

To forward UDP broadcasts to the DHCP server, configure helper addresses on the interface. If you have configured the secondary addresses on that interface and you want the router to step through each IP network when forwarding DHCP requests, use the `ip dhcp smart-relay` command.

If the smart relay agent forwarding is not configured, all requests are forwarded using only the primary IP address on the interface.

If the `ip dhcp smart-relay` command is configured, the relay agent counts the number of times that the client retries sending a request to the DHCP server when there is no DHCPOFFER message from the DHCP server. After three retries, the relay agent sets the gateway address to the secondary address. If the DHCP server still does not respond after three more retries, then the next secondary address is used as the gateway address.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Before you begin

- To configure the IP DHCP smart-relay feature, configure the IP helper address on desired interfaces using `ip helper-address` command. You can use the `service dhcp` command to enable the DHCP service or the `no service dhcp` command to disable it, depending on the requirement.
- The Cisco DHCP relay agent is enabled on an interface only when the `ip helper-address` command is configured. This command enables the DHCP broadcast to be forwarded to the configured DHCP server.

Procedure

1. Enter SD-WAN configuration mode.

   Example

   ```
   sdwan
   ```

2. Enable DHCP server.

   Example

   ```
   service dhcp
   ```

3. In the SD-WAN configuration mode, configure an interface type such as **Gigabit Ethernet.**

   Example

   ```
   interface Gigabitethernet0/0
   ```

4. Enable the DHCP broadcast to be forwarded to the configured DHCP server.

   Example

   ```
    ip helper-address
   ```

5. Configure the DHCP relay agent to switch the gateway address to a secondary address when there is no DHCPOFFER message from a DHCP server.

   Example

   ```
   ip dhcp smart-relay
   ```

This is a DHCP smart-relay CLI configuration. In the example, the device forwards the DHCP broadcast received on Gigabitethernet interface 0/0 to the DHCP server (10.0.0.1), by inserting 192.168.255.254 in the gateway address field of the DHCP packet.

```
service dhcp
ip address 172.16.0.1 255.255.0.0
secondary ip address 192.168.255.254 255.255.0.0

interface GigabitEthernet0/0
ip helper-address 10.0.0.1
ip dhcp smart-relay
end
```

10 Layer 2 VPN

Topics:

- Feature history for L2VPN
- Layer 2 VPNs within the SD-WAN overlay network
- Supported platforms for Layer 2 VPN
- Restrictions for Layer 2 VPN
- Methods to configure Layer 2 VPN using CLI template
- Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template
- Configure a point-to-point Layer 2 VPN using CLI template
- Configure a point-to-multipoint Layer 2 VPN using CLI template
- Configure a Layer 2 VPN Switchport using CLI template
- Methods to verify Layer 2 VPN using CLI
- Monitor configured layer 2 VPN using CLI

Introduces Layer 2 VPN overview, highlighting key concepts, use cases, and fundamental capabilities within Cisco Catalyst SD-WAN deployments.

Feature history for L2VPN

Details the chronological development and enhancements of Layer 2 VPN features, providing a historical perspective on capability additions and changes.

This table describes the developments of this feature, by release.

Table 37: Feature History

Feature Name	Release Information	Description
Layer 2 (L2) VPN	Cisco Catalyst SD-WAN Manager Release 20.14.1 Cisco IOS XE Catalyst SD-WAN Release 17.14.1a	The feature adds Layer 2 VPN support on the Cisco Catalyst SD-WAN overlay network. It allows you to configure Layer 2 point-to-point and point-to-multipoint connections within the Cisco Catalyst SD-WAN fabric.
Layer 2 (L2) VPN Multihoming and Hub-and-Spoke Support	Cisco Catalyst SD-WAN Manager Release 20.15.1 Cisco IOS XE Catalyst SD-WAN Release 17.15.1a	With this feature, you can configure Layer 2 VPN on multiple devices on the same site in an active-standby configuration. This feature also enables Layer 2 connections using an indirect path, such as a hub, for point-to-multipoint connections within the Cisco Catalyst SD-WAN fabric.

Layer 2 VPNs within the SD-WAN overlay network

Explains Layer 2 VPN integration within the Cisco Catalyst SD-WAN overlay network, covering architecture, support models, and interoperability considerations.

Layer 2 VPN within the Cisco Catalyst SD-WAN overlay network is a network feature that

- enables Layer 2 connectivity across the SD-WAN fabric for legacy systems and non-IP applications,
- supports point-to-point (P2P) and point-to-multipoint (P2MP) L2VPN services with options for single homing, multihoming, and topologies including full mesh and hub-and-spoke, and
- provides MAC learning through OMP protocol (Control Plane), along with features such as ingress replication for broadcast, unknown-unicast, and multicast (BUM) traffic.

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, the following L2VPN features are supported:

- Point-to-point L2VPN Service (P2P)
- Point-to-Multipoint L2VPN Service (P2MP)
- Single homing
- Flood and Learn in WAN and LAN
- Ingress replication for Broadcast, Unknown-unicast and Multicast (BUM)
- Full mesh topology only

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, the following L2VPN features are supported:

- Multihoming for P2P and P2MP

- Hub-and-spoke topology support for L2VPN services
- The MAC learning mode (previously the Flood and Learn in WAN and LAN) is changed to learning through OMP protocol (that is, Control Plane).

Note

From Cisco IOS XE Catalyst SD-WAN Release 17.18.1a, you can adjust the TCP Maximum Segment Size (MSS) even for a TCP packet encapsulated in an MPLS label. You can set the TCP MSS per the Path Maximum Transmission Unit (PMTU) with 30 bytes to account for Layer 2 headers, such as Ethernet, VLAN tags and MPLS headers.

For IPv4, the TCP MSS is set per PMTU with 80 bytes for IPv4 and TCP headers and an additional 30 bytes for Layer 2 headers. For example, if the PMTU is 1438, the TCP MSS is set as 1328 (1438 - 80 - 30).

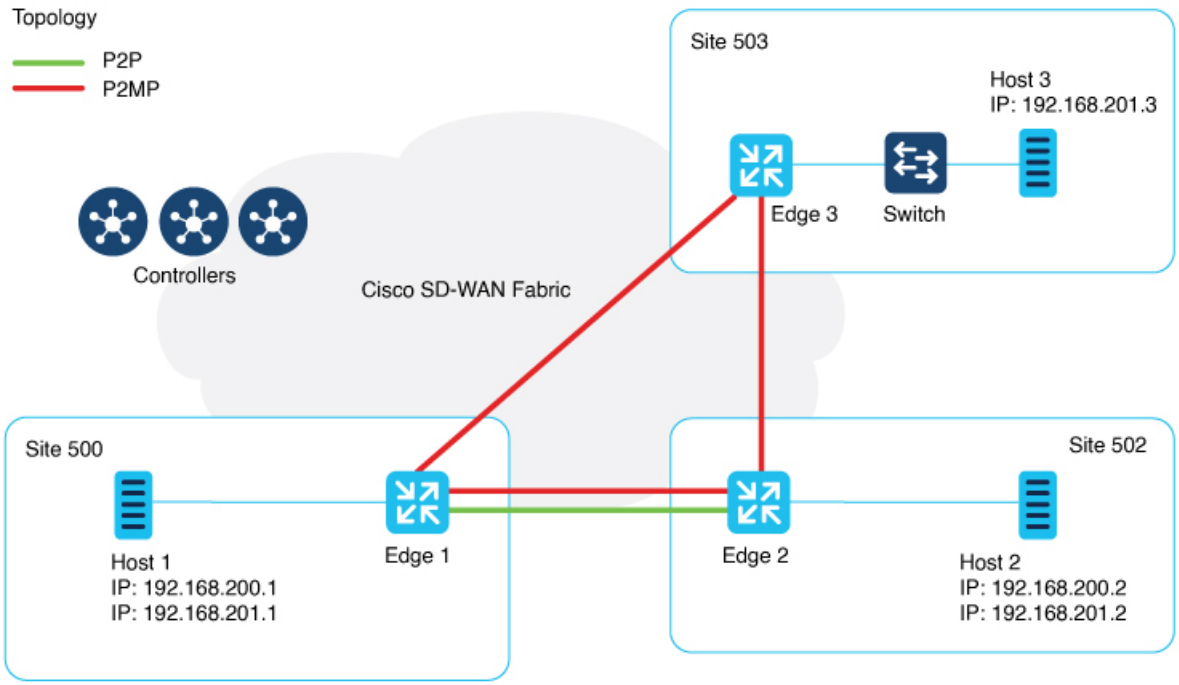
For IPv6, the TCP MSS is set per PMTU with 100 bytes for IPv6 and TCP headers and an additional 30 bytes for Layer 2 headers. For example, if the PMTU is 1438, the TCP MSS is set as 1308 (1438-100-30).

For more information about configuring TCP MSS, see Configure TCP MSS Using CLI

This change helps prevent drop in TCP sessions, improving overall network performance and reliability.

Network Topology for Layer 2 Connections

Figure 5: Topology



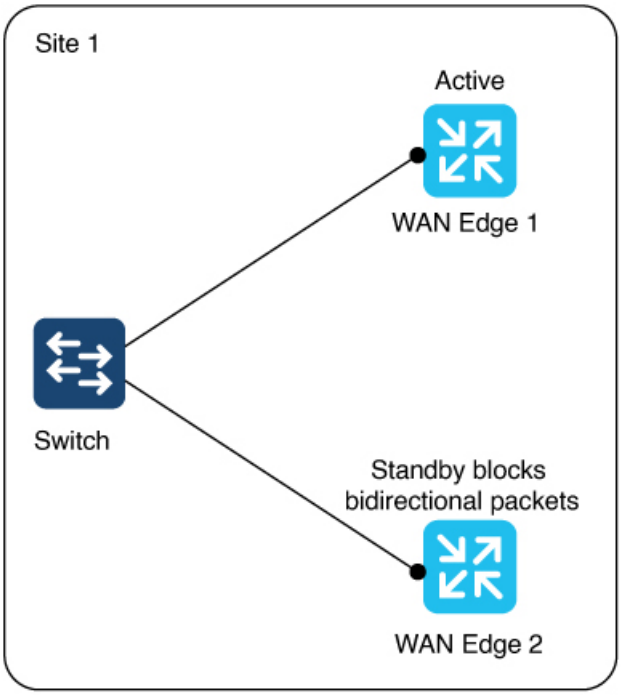
This illustration shows three sites and shows P2P (green line) and P2MP (red lines) connections between edge routers at the sites.

- Point-to-Point (P2P): Connects sites 500 and 502 with a dedicated Layer 2 VPN. The L2VPN connection between the two sites allows Host 1 and Host 2 to interact.

- Point-to-Multipoint (P2MP): Connects sites 500, 502, and 503 with Layer 2 VPN. Host 1 communicates with both Host 2 and Host 3 across a Layer 2 multipoint network.

The L2VPN connections use existing Cisco Catalyst SD-WAN tunnels.

Figure 6: Multihoming



The illustration shows two edge routers on the same site connected to a switch. For an (instance-id + vc), one router is active and the other is on standby. (instance-id +vc) maps to a bridge domain and a bridge-domain maps to a VLAN (or a VLAN range).

The router on standby blocks bidirectional traffic for that VLAN.

Multihoming supports L2VPN configuration on up to two edge devices on the same site, thereby providing redundancy for L2VPN service over SD-WAN.

Multihoming allows an active-standby scenario where one device is chosen as active and the other as standby. This provides automated failover. It determines which of the two edge devices should be active and which one should be on standby. When the OMP timer expires on the controller, it marks the L2VPN status route as stale, and notifies other edges.

Role determination for an Active and Standby device

The active and standby role between edge devices is automatically determined by with this algorithm: (SDWAN-Instance-ID + VC-ID) modular 2.

If the modular result is 0, the edge with lower system-ip is selected as the active device. The edge with the higher system-ip is selected as the standby device.

If the modular result is 1, the edge with higher system-ip is selected as the active device. The edge with the lower system-ip is selected as the standby device.

Example:

There are two WAN edge devices. WAN edge 1 has a system-ip of 172.16.255.10. WAN edge 2 has a system-ip of 172.16.255.11.

For sdwan-instance-id 100, vc-id 2, WAN edge 1 with the lower system-ip is selected as the active device. WAN edge 2 is the standby device.

For sdwan-instance-id 100, vc-id 1, WAN edge 2 with the higher system-ip is selected as the active device. WAN edge 1 is the standby device.

If a failure occurs on the service side of one of the edge devices, the controller is notified about a change to the L2VPN status route, and other edge routers can switchover traffic to the new active device.

Supported platforms for Layer 2 VPN

Lists platforms compatible with Layer 2 VPN, describing hardware and software requirements for successful deployment in Cisco Catalyst SD-WAN environments.

All Cisco IOS XE Catalyst SD-WAN devices.

Restrictions for Layer 2 VPN

Outlines operational limitations, deployment restrictions, and unsupported features associated with Layer 2 VPN to inform optimal usage and prevent configuration issues.

Supported configuration CLI templates for Layer 2 VPN

Only CLI template or CLI add-on template configuration is supported for Layer 2 VPN.

LAN side interface limitation in single homing and multihoming

For both single homing and multihoming, only one LAN side interface is supported in a bridge-domain.

Point-to-Multipoint (P2MP) requirements for hub and spoke connectivity

P2P configuration between two spokes is not supported. In such cases, use P2MP instead of P2P.

> **Note**
>
> P2P configuration between hub and spoke is supported.

Multihoming support only for dual homing

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, multihoming only supports dual homing.

Layer 2 VPN limitations for hub-and-spoke topology

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, hub-and-spoke topology is supported for Layer 2 VPN. It is limited by:

- No support for point-to-point Layer 2 VPN service between spokes.
- Support for up to 6000 spokes and 6000 sites within the same Layer 2 VPN in hub-and-spoke topology, and
- Support for only 256 sites within the same Layer 2 VPN in a non-hub-and-spoke design.

TCP performance degradation in some deployments

In SD-WAN Layer 2 VPN deployments using segment routing on cEdge devices running Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, the TCP SYN packets encapsulated with MPLS labels are not analyzed for Maximum Segment Size (MSS) adjustment. Therefore, MSS is not set correctly resulting in unexpected TCP fragmentation and potentially degraded TCP performance.

Methods to configure Layer 2 VPN using CLI template

Describes available CLI template methods for configuring Layer 2 VPN, emphasizing automation approaches and recommended practices for network deployment.

Using these procedures, configure a Layer 2 VPN on a Cisco Catalyst SD-WAN overlay network.

- Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template on page 102
- Configure a point-to-point Layer 2 VPN using CLI template on page 103
- Configure a point-to-multipoint Layer 2 VPN using CLI template on page 106
- Configure a Layer 2 VPN Switchport using CLI template on page 110

Configure a Layer 2 VPN on a Cisco IOS XE Catalyst SD-WAN device using CLI template

Provides step-by-step instructions to configure Layer 2 VPNs on Cisco devices using CLI templates for efficient service provisioning.

Before you begin

For information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Follow these steps to configure an L2VPN on a Cisco IOS XE Catalyst SD-WAN Device Using CLI Template

Procedure

1. Configure an L2VPN instance for P2P and P2MP connections.

```
l2vpn sdwan instance instance-id point-to-point
l2vpn sdwan instance instance-id multipoint
```

   The instance ID is a unique identifier for each L2VPN connection, and must not overlap or be shared with any Layer 3 VRFs in the SD-WAN fabric. For example, you cannot use L2VPN instance 10 and vrf definition 10.

2. Configure a bridge-domain.

```
bridge-domain bridge-id
```

3. Configure a Layer 2 interface on a Cisco IOS XE Catalyst SD-WAN device.

```
interface vlan-id
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

> **Note**
>
> A rewrite is used to modify the default VLAN tag. If you have not configured rewrite under service instance, dot1q must be the same at all sites participating in the Layer 2 network. The rewrite option in a Layer 2 configuration modifies the VLAN tags of packets as they ingress or egress an interface. To use the rewrite option, you need to configure Ethernet Virtual Connections (EVCs) on edge routers (Cisco ASR 1000 Series). For more information about configuring an EVC, see Configuring Ethernet Virtual Connections on a Cisco Router.

Configure a point-to-point Layer 2 VPN using CLI template

Guides configuration of point-to-point Layer 2 VPNs using CLI templates, including procedures for setting up edge routers at Site A and Site B to establish connectivity.

Before you begin

- You can use one L2VPN instance ID for one or more bridge domains. It must be the same at both ends of the circuit.

  To identify a particular bridge-domain, use Virtual Circuit (VC) ID. This ID is the identifier of the virtual circuit between the Cisco IOS XE Catalyst SD-WAN devices.
- To create a P2P pseudowire, L2VPN instance ID, and VC ID must be the same on different Cisco IOS XE Catalyst SD-WAN devices.
- Remote-site-id is only supported for P2P configuration.
- For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates for Cisco IOS XE Catalyst SD-WAN Devices

> **Note**
>
> By default, CLI templates execute commands in Global Configuration mode.

Follow these steps to configure P2P L2VPN services between two sites (Site A and Site B) on the Cisco Catalyst SD-WAN overlay network.

Procedure

1. Configure an edge router at Site A for point-to-point Layer 2 VPN using CLI template on page 103
2. Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple on page 104

Configure an edge router at Site A for point-to-point Layer 2 VPN using CLI template

Site A uses an edge router and connects the Ethernet interface to the L2 network that bridges to Site B.

Follow these steps to configure a an Edge Router at Site A for Point-to-Point Layer 2 VPN Using CLI Template

Procedure

1. Define the L2VPN instance for point-to-point service:

```
l2vpn sdwan instance instance-id point-to-point
```

2. Configure the Ethernet interface:

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
```

3. Define the bridge domain and associate it with the interface and L2VPN instance:

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id
virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, you can specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
 dual-homing
```

Example

The following configures Site A using Cisco Catalyst 8000V Edge Software to manage traffic through GigabitEthernet5, which is linked to the Layer 2 network that provides connectivity to Site B.

```
l2vpn sdwan instance 100 point-to-point

interface GigabitEthernet5
 service instance 100 ethernet
  encapsulation dot1q 2002
  !
bridge-domain 100
 member GigabitEthernet5 service-instance 100
 member sdwan-instance 100 remote-site 502 vc-id 100 single-homing
```

Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple

Site B uses an edge router and Switchport Ethernet interface.

Follow these steps to configure an edge router at Site B for point-to-point Layer 2 VPN.

Before you begin

Procedure

1. Define the L2VPN instance for point-to-point service.

```
l2vpn sdwan instance instance-id point-to-point
```

2. Define the VLAN for the L2VPN.

```
vlan vlan-id
 name l2vpn
```

3. Configure the VLAN interface.

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

4. Configure the Ethernet interface as an access port for VLAN.

```
interface interface-name
 switchport access vlan vlan-id
```

5. Define the bridge-domain for site B and associate it with the VLAN and L2VPN instance.

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id
virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
 dual-homing
```

Example

The following configures Switchport GigabitEthernet 0/1/7 at Site B to connect to the interface with a Cisco ISR1100-8P device.

```
l2vpn sdwan instance 100 point-to-point
vlan 2002
 name L2vpn
interface Vlan2002
 service instance 100 ethernet
  encapsulation dot1q 2002
  no shutdown
```

```
  !
interface GigabitEthernet 0/1/7
 switchport access vlan 2002
bridge-domain 100
 member Vlan2002 service-instance 100
 member sdwan-instance 100 remote-site 500 vc-id 100 single-homing
```

After configuring the point-to-point L2VPN service on both sites, you can integrate these configuration blocks into your CLI Template or CLI Add-On Feature Template. This template can then be used to deploy the configuration across the relevant devices in the Cisco Catalyst SD-WAN fabric. Verify the connectivity and functionality of the L2VPN service following the deployment to confirm that the bridge between site A and site B is operational.

Configure a point-to-multipoint Layer 2 VPN using CLI template

Explains configuration workflows for point-to-multipoint Layer 2 VPNs via CLI templates, detailing steps for edge routers at sites A, B, and C to support multipoint connectivity.

Before you begin

- For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

  By default, CLI templates execute commands in global config mode.
- One L2VPN instance ID can be used by one or more bridge domains. VC ID is used to identify a particular bridge-domain.
- L2VPN instance ID and VC ID must be the same on different edge devices.

Follow these steps to configure P2MP Layer 2 VPN over Cisco Catalyst SD-WAN overlay, connecting a local Layer 2 network at site A to multiple remote sites (B and C). Site A uses Gigabit Ethernet interface to connect to the Layer 2 network for bridging.

Procedure

1. Configure an edge router at sites A, B and C on page 106
2. Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple on page 104
3. Configure an edge router at Site C for point-to-point Layer 2 VPN using CLI template on page 108

Configure an edge router at sites A, B and C

Site A is using an edge router, where an Ethernet interface is connected to the Layer 2 network that bridges to Site B and Site C.

Follow these steps to configure an Edge Router at Sites A, B, and C.

Procedure

1. Define the L2VPN instance for the multipoint service on the data center router:

   l2vpn sdwan instance *instance-id* **multipoint**

2. Configure the Ethernet interface on the data center router:

```
interface interface-name
service instance instance-id ethernet
encapsulation dot1q vlan-id
```

3. Define the bridge-domain on the data center route and associate it with the interface and L2VPN instance:

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
 single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
 dual-homing
```

What to do next

See the topic, Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI Temple.

Configure an edge router at Site B for point-to-point Layer 2 VPN using CLI temple

Site B uses an edge router and Switchport Ethernet interface.

Follow these steps to configure an edge router at Site B for point-to-point Layer 2 VPN.

Before you begin

Procedure

---

1. Define the L2VPN instance for point-to-point service.

```
l2vpn sdwan instance instance-id point-to-point
```

2. Define the VLAN for the L2VPN.

```
vlan vlan-id
 name l2vpn
```

3. Configure the VLAN interface.

```
interface interface-name
 service instance instance-id ethernet
  encapsulation dot1q vlan-id
  no shutdown
```

4. Configure the Ethernet interface as an access port for VLAN.

```
interface interface-name
 switchport access vlan vlan-id
```

5. Define the bridge-domain for site B and associate it with the VLAN and L2VPN instance.

```
bridge-domain bridge-id
 member vlan-name service-instance instance-id
 member sdwan instance  instance-id  remote-site remote-site-id vc-id
virtual-circuit-id single-homing
```

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

```
bridge-domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id remote-site remote-site-id vc-id virtual-circuit-id
 dual-homing
```

Example

The following configures Switchport GigabitEthernet 0/1/7 at Site B to connect to the interface with a Cisco ISR1100-8P device.

```
l2vpn sdwan instance 100 point-to-point
vlan 2002
 name L2vpn
interface Vlan2002
 service instance 100 ethernet
  encapsulation dot1q 2002
  no shutdown
  !
interface GigabitEthernet 0/1/7
 switchport access vlan 2002
bridge-domain 100
 member Vlan2002 service-instance 100
 member sdwan-instance 100 remote-site 500 vc-id 100 single-homing
```

After configuring the point-to-point L2VPN service on both sites, you can integrate these configuration blocks into your CLI Template or CLI Add-On Feature Template. This template can then be used to deploy the configuration across the relevant devices in the Cisco Catalyst SD-WAN fabric. Verify the connectivity and functionality of the L2VPN service following the deployment to confirm that the bridge between site A and site B is operational.

Configure an edge router at Site C for point-to-point Layer 2 VPN using CLI template

Before you begin

Repeat the same steps as for branch router C, substituting the specific interface used on site B.

Follow these steps to configure an edge router at Site C for point-to-point Layer 2 VPN.

Procedure

1. Define the L2VPN instance for multipoint service on the branch router:

   l2vpn sdwan instance *instance-id* **multipoint**

2. Define the VLAN for the L2VPN on the branch router:

   vlan *vlan-id*
   name L2vpn

3. Configure the VLAN interface on the branch router:

   interface *interface-name*
   service instance *instance-id* **ethernet**
   encapsulation dot1q *vlan-id*
   no shutdown

4. Configure the Ethernet interface on the branch router as an access port for VLAN:

   interface *interface-name*
   switchport access vlan *vlan-id*

5. Define the bridge-domain on the branch router and associate it with the VLAN and L2VPN instance:

   bridge-domain *bridge-id*
   member *vlan-name* **service-instance** *instance-id*
   member sdwan instance *instance-id* **vc-id** *virtual-circuit-id* **single-homing**

   Starting from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, specify the homing type as dual homing to enable multihoming.

   bridge-domain *bridge-id*
   member *vlan-name* **service-instance** *instance-id*
   member sdwan instance *instance-id* **remote-site** *remote-site-id* **vc-id** *virtual-circuit-id*
   dual-homing

Example

This section provides an example configuration for P2MP L2VPN service within the Cisco Catalyst SD-WAN overlay network, connecting a local Layer 2 network at site A to multiple remote sites (B and C). Site A uses GigabitEthernet6 interface to connect to the L2 network for bridging.

Verify the connectivity and functionality of the P2MP L2VPN service and ensure that all sites are correctly bridged.

Site A is using a Cisco Catalyst 8000V edge router, where GigabitEthernet6 is connected to the Layer 2 network that bridges to site B and site C.

```
l2vpn sdwan instance 200 multipoint

vlan 2001
 name L2MPvpn

interface Vlan2001
 service instance 200 ethernet
```

```
  encapsulation dot1q 2001
  no shutdown
  !
interface GigabitEthernet 0/1/6
 switchport access vlan 2001

bridge-domain 200
 member Vlan2001 service-instance 200
 member sdwan-instance 200 vc-id 200 single-homing
```

Configure branch router C:

Repeat the same steps as for branch router B, substituting the specific interface used on router 503. In this example, we have used the GigabitEthernet 0/1/6 interface.

```
l2vpn sdwan instance 200 multipoint

vlan 2001
 name L2MPvpn

interface Vlan2001
 service instance 200 ethernet
  encapsulation dot1q 2001
  no shutdown
  !
bridge-domain 200
 member Vlan2001 service-instance 200
 member sdwan-instance 200 vc-id 200 single-homing
```

Configure a Layer 2 VPN Switchport using CLI template

Provides instructions to configure switchports for Layer 2 VPN services using CLI templates to enable robust edge device connectivity.

If your device such as Cisco ISR1121-8P or similar has embedded switchports and you want to use one of them for the L2VPN services, configure a VLAN interface first and then assign that VLAN to your switchport as described in this section.

To support a Layer 2 switchport, configure a service instance in the VLAN interface. In the VLAN interface, a packet always has the dot1q tag even when the Layer 2 switchport is configured with switchport mode access. Therefore, the dot1q tag is mandatory in the service instance of the VLAN interface.

This following section provides steps to configure a Layer 2 switchport for P2MP (applicable for devices with embedded switchports). You can also configure a Layer 2 switchport for P2P by updating the Layer 2 VPN instance command.

Site A is using an edge router, where the Ethernet interface is connected to the Layer 2 network that bridges to Site B and Site C.

Follow these steps to configure a Layer 2 VPN Switchport using CLI template.

Procedure

1. Define the Layer 2 VPN instance for multipoint service on the branch routers:

   l2vpn sdwan instance *instance-id* **multipoint**

2. Define the VLAN for the Layer 2 VPN on the branch routers:

```
vlan vlan-id
name l2vpn
```

3. Configure the Ethernet interface on the routers:

```
interface interface-name
```

4. Set the switch port access VLAN and switchport mode to access to accept traffic only from the specified VLAN:

```
switchport access Vlan vlan-id
```

5. Configure the VLAN interface on a router and disable the IP address assignment

```
interface interface-name
no ip address
service instance instance-id ethernet
encapsulation dot1q vlan-id
```

6. Define the bridge-domain on the data center router and associate it with the interface and L2VPN instance:

```
bridge domain bridge-id
member vlan-name service-instance instance-id
member sdwan instance instance-id vc-id virtual-circuit-id single homing
```

The following configures a Layer 2 VPN Switchport to integrate a multipoint SD-WAN instance and bridge-domain. This configuration sets up GigabitEthernet0/1/2 as an access port for VLAN 201.

```
l2vpn sdwan instance 200 multipoint

interface GigabitEthernet0/1/2
 switchport access Vlan 201
 switchport mode access

interface Vlan201
 no ip address
 service instance 200 ethernet
  encapsulation dot1q 201
  !

bridge-domain 201
 member Vlan201 service-instance 200
 member sdwan-instance 200 vc-id 201 single-homing
```

Methods to verify Layer 2 VPN using CLI

Describes methods and CLI commands to verify Layer 2 VPN status, ensuring correct configuration and operational integrity.

To verify Layer 2 VPN using CLI, use these methods.

1. View a Layer 2 VPN status on page 112
2. View L2VPN information learned through OMP route on a Cisco SD-WAN Controller on page 112

3. View Bridge-domain information on page 113
4. View Cisco Catalyst SD-WAN Flood List Information and Packet Counters in Data Plane on page 114
5. View packet counters in data plane on page 114

View a Layer 2 VPN status

To view the remote peer information, system IP, status, and related information, use the **show l2vpn sdwan [instance** *instance-id]***[vc-id** *vc-id]* command.

This is an example for a Cisco IOS XE Catalyst SD-WAN device.

```
Device# show l2vpn sdwan instance 13 vc-id 13
VC_ID: 13 Bridge-domain: 13
Local l2vpn status: UP
Local Pseudoports: GigabitEthernet7 service instance 13
```

View L2VPN information learned through OMP route on a Cisco SD-WAN Controller

To view the specific L2-route or path learned in the specific VPN and virtual circuit, use the **show sdwan omp l2-routes[vpn** *vpn-id*] [**vc-id** *vc-id*] command. If the **vpn** and **vc-id** are not included, the command shows Layer 2 routes learned through OMP from all VPNs across the Cisco Catalyst SD-WAN fabric.

This is a sample output from the **show omp l2-routes** command displaying Layer 2 routes learned through OMP for Cisco SD-WAN Controllers.

```
Device# show omp l2-routes | tab
C   -> chosen
I   -> installed
Red -> redistributed
Rej -> rejected
L   -> looped
R   -> resolved
S   -> stale
Ext -> extranet
Inv -> invalid
Stg -> staged
IA  -> On-demand inactive
U   -> TLOC unresolved


                                                                  REMOTE
                                          ROUTE                     IP
     SITE                      PATH                              SITE
VPN    VC ID        ORIGINATOR       TYPE   MAC ADDRESS          ADDRESS  VPN TYPE
    ID    FROM PEER        ID     LABEL    STATUS    ID
-------------------------------------------------------------------------------
12     12           172.16.255.15    vpn    0000.0000.0000   ::       p2p
     500   172.16.255.15    66     1004     C,R       501

           172.16.255.15    69     1004     C,R       501

           172.16.255.20    1      1004     C,R       501

           172.16.255.20    2      1004     C,R       501
12     12           172.16.255.27    vpn    0000.0000.0000   ::       p2p
     501   172.16.255.20    1      1014     C,R       500

           172.16.255.27    70     1014     C,R       500
13     13          172.16.255.15    vpn    0000.0000.0000   ::       multipoint
```

```
  500   172.16.255.15    66     1006     C,R       -

           172.16.255.15    69     1006     C,R       -

           172.16.255.20    1      1006     C,R       -

           172.16.255.20    2      1006     C,R       -
13     13          172.16.255.27    vpn    0000.0000.0000   ::       multipoint
  501   172.16.255.20    1      1016     C,R       -

           172.16.255.27    70     1016     C,R       -
13     13          172.16.255.32    vpn    0000.0000.0000   ::       multipoint
  503   172.16.255.20    1      1007     C,R       -

           172.16.255.32    71     1007     C,R       -
14     1           172.16.255.27    vpn    0000.0000.0000   ::       multipoint
  501   172.16.255.20    1      1018     C,R       -

           172.16.255.27    70     1018     C,R       -
15     1            172.16.255.15    vpn    0000.0000.0000   ::       p2p
     500   172.16.255.15    66     1020     C,R       501

           172.16.255.15    69     1020     C,R       501

           172.16.255.20    1      1020     C,R       501

           172.16.255.20    2      1020     C,R       501
15     1            172.16.255.27    vpn    0000.0000.0000   ::       p2p
     501   172.16.255.20    1      1020     C,R       500

           172.16.255.27    70     1020     C,R       500
```

View Bridge-domain information

To verify information related to bridge domains within the context of Forwarding Table Management Daemon (FTMD), use the **show platform software sdwan ftmd bridge-domain** command on a device.

This is a sample output from the **show platform software sdwan ftmd bridge-domain** command that displays information related to bridge domains within the context of Forwarding Table Management Daemon (FTMD).

```
Device# show platform software sdwan ftmd bridge-domain
L2vpn Bridge-domain 12 Table:
  sdwan efp dpidx: 4210708(0x404014)
  Label: 1004 lbl-nhop-id: 196611 (binosId=0xf830003f)
  Bum Label: 1005 bum-lbl-nhop-id: 196612 (binosId=0xf830004f)
  Remote Site Table(1 entries in total):
    remote-site-id: 501 sla-nhop-id: 29 (binosId=0xf80001df)

L2vpn Bridge-domain 13 Table:
  sdwan efp dpidx: 4210709(0x404015)
  Label: 1006 lbl-nhop-id: 196613 (binosId=0xf830005f)
  Bum Label: 1007 bum-lbl-nhop-id: 196614 (binosId=0xf830006f)
  Remote Site Table(2 entries in total):
    remote-site-id: 501 sla-nhop-id: 30 (binosId=0xf80001ef)
remote-site-id: 503 sla-nhop-id: 33 (binosId=0xf800021f)
```

View Cisco Catalyst SD-WAN Flood List Information and Packet Counters in Data Plane

To verify information related to Cisco Catalyst SD-WAN flood list information, use the **show platform hardware qfp active feature bridge-domain datapath** *bridge-domain-id* **sdwan-flood-list** command.

This is a sample output from the **show platform hardware qfp active feature bridge-domain datapath** *bridge-domain-id* **sdwan-flood-list** command that displays the Cisco Catalyst SD-WAN flood list information.

```
Device#show platform software sdwan ftmd bridge-domain
L2vpn Bridge-domain 12 Table:
  sdwan efp dpidx: 4210708(0x404014)
  Label: 1004 lbl-nhop-id: 196611 (binosId=0xf830003f)
  Bum Label: 1005 bum-lbl-nhop-id: 196612 (binosId=0xf830004f)
  Remote Site Table(1 entries in total):
    remote-site-id: 501 sla-nhop-id: 29 (binosId=0xf80001df)

L2vpn Bridge-domain 13 Table:
  sdwan efp dpidx: 4210709(0x404015)
  Label: 1006 lbl-nhop-id: 196613 (binosId=0xf830005f)
  Bum Label: 1007 bum-lbl-nhop-id: 196614 (binosId=0xf830006f)
  Remote Site Table(2 entries in total):
    remote-site-id: 501 sla-nhop-id: 30 (binosId=0xf80001ef)
remote-site-id: 503 sla-nhop-id: 33 (binosId=0xf800021f)
```

View packet counters in data plane

To verify information related to a QuantumFlow Processor (QFP) hardware module packet counters for a specific bridge domain within the data path, use the **show platform hardware qfp active feature bridge-domain datapath** *bridge-id* command.

This is a sample output from the **show platform hardware qfp active feature bridge-domain datapath** *bridge-id* command to display a QFP hardware module packet counters for a specific bridge domain within the data path.

```
Device# show platform hardware qfp active feature bridge-domain datapath 200
QFP L2BD Bridge Domain information



BD id                  : 200

State enabled          : Yes

Aging timeout (sec)    : 300

Aging active entry     : Yes

Max mac limit          : 65536

Unkwn mac limit flood  : Yes

mac_learn_enabled      : Yes

mac_learn_controled    : No

Unknown unicast olist  : Yes

otv_aed_enabled : No

otv_enabled : No
```

```
mcast_snooping_enabled : No

Feature : sdwan

SISF snoop protocols   : None

Sdwan instance id      : 200

Mac learned            : 0

BDI outer vtag         : 00000000

BDI inner vtag         : 00000000


Replication tree info:

  Global replication    : depth encode 0X1000001, (head 0XE4E90000)

  Split-horizon-group 0 : depth encode 00000000, (head 00000000)

  Split-horizon-group 1 : depth encode 00000000, (head 00000000)
Bridge Domain statistics


Total bridged                pkts : 0          bytes: 0

Total unknown unicast        pkts : 0          bytes: 0

Total broadcasted            pkts : 0          bytes: 0

Total to BDI                 pkts : 0          bytes: 0

Total injected               pkts : 0          bytes: 0

Total mac-sec violation drop pkts : 0          bytes: 0

Total mac-sec move drop      pkts : 0          bytes: 0

Total mac-sec unknown drop   pkts : 0          bytes: 0

Total source filter drop     pkts : 0          bytes: 0

Total bfib policy drop       pkts : 0          bytes: 0

Total replication start drop pkts : 0          bytes: 0

Total recycle tail drop      pkts : 0          bytes: 0

Total static MAC move drop   pkts : 0          bytes: 0

Total BD disabled drop       pkts : 0          bytes: 0

Total STP state drop         pkts : 0          bytes: 0

Total UUF suppression drop   pkts : 0          bytes: 0

Total sisf ctrl punt         pkts : 0          bytes: 0

Total sisf ctrl drop         pkts : 0          bytes: 0
```

```
Total p2p lan to wan         pkts : 0          bytes: 0

Total p2p wan to lan         pkts : 0          bytes: 0
```

Monitor configured layer 2 VPN using CLI

Introduces CLI-based monitoring techniques for configured Layer 2 VPNs, enabling administrators to assess ongoing operational status and performance.

This is a sample output from the **show l2vpn sdwan all** command. The following examples show the configuration and status information for Layer 2 VPN instances within a Cisco Catalyst SD-WAN overlay network. The output includes details for both point-to-point (P2P) and point-to-multipoint (P2MP) topologies.

Example 1: The example shows the L2VPN SD-WAN instance for instance 100 for point-to-point connectivity.

```
Device# show l2vpn sdwan all
L2VPN sdwan Instance : 100
VPN Type : point-to-point
  VC_ID: 100 Bridge-domain: 100 UP
    Local l2vpn status: UP
    Local Pseudoports: GigabitEthernet5 service instance 100
    Remote Site: 53
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.53       DOWN         00:15:04   public-internet ipsec    1023
   N/A
```

Example 2: The example shows all the Layer 2 VPN SD-WAN instance for instance 200 for point-to-point connectivity.

```
Device# show l2vpn sdwan all
L2VPN sdwan Instance : 200
VPN Type : multipoint
IP Local-learning    : Disabled
Flooding Suppression : Disabled
  VC_ID: 200 Bridge-domain: 200 UP
    Local l2vpn status: UP
    Local Pseudoports: GigabitEthernet5 service instance 200
    Remote Site: 50
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.50       UP           00:04:14   public-internet ipsec    1008
   N/A

    Remote Site: 53
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.53       UP           00:15:00   public-internet ipsec    1025
   N/A
```

This is a sample output from the **show l2vpn sdwan instance** *instance-id* **vc-id** *vc-id***peers** command. The following examples show information about a specific Cisco Catalyst SD-WAN Layer 2 VPN instance (instance 200) and its associated virtual circuit (vc-id 200), including details about its peer connections.

```
show l2vpn sdwan instance instance-id vc-id vc-id peers
```

Example 1

```
Device1# show l2vpn sdwan instance 200 vc-id 200 peers
    Remote Site: 50   MACs Learn: 0
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.50       UP           00:19:54   public-internet ipsec    1008
   N/A
    Remote Site: 53   MACs Learn: 0
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.53       UP           00:30:40   public-internet ipsec    1025
   N/A
```

Example 2

```
Device# show l2vpn sdwan instance 200 vc-id 200 peers
     Remote Site: 1   MACs Learn: 0
      System IP          status       up/down    color           encap    label
  DF
      10.100.31.1        UP           00:30:13   public-internet ipsec    1014
   N/A
```

11 IPv6 Functionality

Topics:

- Configure IPv6 functionality

Explains how to configure IPv6 functionality across network components, guiding you through setup of interfaces, OMP, BGP, VRRP, SNMP, DHCP relay agents, ACL, QoS, logging hosts, prefix lists, data prefixes, and centralized or localized policies using templates and CLI commands.

Configure IPv6 functionality

Guides you through configuring IPv6 functionality, including essential setup steps and options for enabling IPv6 on network devices.

Configure IPv6 functionality for an Interface or Subinterface using templates

Before you begin

Perform these steps to configure IPv6 functionality for an interface or subinterface template.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco VPN Interface Ethernet** from the list of templates.
4. From **Basic Configuration**, click **IPv6** and configure these parameters.

Field	Description
Static	Selected by default because IPv6 addresses are static.
IPv6 Address	IPv6 address of the interface or subinterface.

Configure IPv6 functionality for an interface or subinterface using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure according to this example.

```
interface GigabitEthernet1
  no shutdown
  ipv6 address 2001:DB8:1::1/64
  ipv6 enable
```

Configure IPv6 functionality for OMP using templates

Before you begin

Perform these steps to configure IPv6 functionality for OMP.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco OMP** from the list of templates.

4. In the **Advertise** section, select **IPv6** and configure these parameters.

Field	Description
Connected	Click **Off** to disable advertising connected routes to OMP. By default, connected routes are advertised to OMP.
Static	Click **Off** to disable advertising static routes to OMP. By default static routes are advertised to OMP.
BGP	Click **Off** to advertise BGP routes to OMP. By default, BGP routes are not advertised to OMP.

Configure IPv6 functionality for OMP using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Enable service VRF for IPv6, according to this example.

```
config-transaction
vrf definition 1
  rd 1:1
  address-family ipv6
```

3. Enable OMP, according to this example.

   OMP supports global IPv6 configuration. In addition, per VRF level configuration is allowed. Per VRF level configuration overrides global configuration.

```
config-transaction
sdwan
  omp
  !
  address-family ipv6
   advertise bgp
   advertise connected

   address-family ipv6 vrf 1
    advertise static
```

4. Global configuration is the default configuration, so IPv6 is enabled by default for OMP. To disable IPv6 OMP route redistribution for a particular VRF, configure the redistribution protocol to **no**.

```
config-transaction
sdwan
  omp
  !
  address-family ipv6
   advertise bgp
   advertise connected

   address-family ipv6 vrf 1
    no advertise connected
```

```
    no advertise static
    no advertise bgp
```

Configure IPv6 functionality for BGP using templates

Before you begin

Perform these steps to configure IPv6 functionality for BGP.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco BGP** from the list of templates.
4. In the **Unicast Address Family** section, select **IPv6** and configure these parameters.

Tab	Field	Description
	Maximum Paths	Maximum number of parallel IBGP paths that can be installed into a route table to enable IBGP multipath load sharing. Range: 0 to 32
	Address Family	BGP IPv6 unicast address family.
RE-DISTRIBUTE		Click the **Redistribute** tab, and then click **Add New Redistribute**.
	Protocol	Select the protocols from which to redistribute routes into BGP, for all BGP sessions. Options are Connected, NAT, OMP, OSPF, and Static. At a minimum, select these: • For service-side BGP routing, select OMP. By default, OMP routes are not redistributed into BGP. • For transport-side BGP routing, select Connected, and then under Route Policy, specify a route policy that has BGP advertise the loopback interface address to its neighbors.
	Route Policy	Name of the route policy to apply to redistributed routes. Click **Add** to save the redistribution information.

Tab	Field	Description
NETWORK		Click the **Network** tab, and then click **Add New Network**.
	Network Prefix	Network prefix in the format of prefix/length, for BGP to advertise.
		Click **Add** to save the network prefix.
AGGREGATE ADDRESS		Click the **Aggregate Address** tab, and then click **Add New Aggregate Address**.
	Aggregate Prefix	Prefix of the addresses to aggregate for all BGP sessions, in the format prefix/length.
	AS Set Path	Click **On** to generate set path information for the aggregated prefixes.
	Summary Only	Click **On** to filter out more specific routes from BGP updates.
		Click **Add** to save the aggregate address.

5. In the **Neighbor** section, select **IPv6**, create a new neighbor or edit an existing one, and then configure these parameters.

Field	Description
IPv6 Address*	IPv6 address of the BGP neighbor.
Description	Description of the BGP neighbor.
Remote AS*	AS number of the remote BGP peer.
Address Family	Select **Global** from the drop-down list, click **On** and select the address family. Enter the address family information.
Shutdown	To shut down a BGP neighbor when you push the template, select **Global** from the drop-down list and then click **Yes**. Default: Off

Configure IPv6 functionality for BGP using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.

2. Configure according to this example.

```
config-transaction
router bgp 1
 bgp log-neighbor-changes
  address-family ipv6 unicast vrf 1
  neighbor 2001:DB8:19::1 remote-as 2
  neighbor 2001:DB8:19::1 activate
  neighbor 2001:DB8:19::1 advertisement-interval 1
  neighbor 2001:DB8:19::1 password cisco
  redistribute omp
  redistribute static
  exit-address-family
```

Configure IPv6 functionality for VRRP using templates

Before you begin

Perform these steps to configure IPv6 functionality for Virtual Router Redundancy Protocol (VRRP).

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco VPN Interface Ethernet** from the list of templates.
4. In the **VRRP** section, select **IPv6**.
5. Click **New VRRP** and configure these parameters.

Field	Description
Group ID	Virtual router ID, which represents a group of routers. Range: 1 to 255
Priority	Priority level of the router within a VRRP group. Range: 1 to 254 Default: 100
Timer	Not used.
Track OMP	Select **On** to track the Overlay Management Protocol (OMP) session running on the WAN connection when determining the primary VRRP virtual router. Default: Off
Track Prefix List	Value to track a list of IPv6 remote prefixes. This value is an alphanumeric string that is configured under Policy.
Link Local IPv6 Address	Virtual link local IPv6 address, which represents the link local address of the group. The address should be in standard link local address format. For example, FE80::AB8.

Field	Description
Global IPv6 Address	Virtual global unicast IPv6 address, which represents the global address of the group. The address should be an IPv6 global prefix address that has the same mask as the interface forwarding address on which the VRRP group is configured. Example: 2001::2/124 Maximum: 3 global IPv6 addresses

Configure IPv6 functionality for VRRP using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure VRRP, according to this example.

```
config-transaction
interface GigabitEthernet1

 vrrp 10 address-family ipv6
   priority 20
   track omp shutdown
   address FE80::10:100:1 primary
   address 2001:10:100::1/64

Prefix-list tracking
track 1 ipv6 route 1:1::1/128
 reachability
 ipv6 vrf 1

 track 2 ipv6 route 2:2::2/128
 reachability
 ipv6 vrf 2

 track 20 list boolean or
  object 1
  object 2

vrrp 10 address-family ipv6
  track 20 shutdown
```

Configure IPv6 functionality for SNMP using templates

Before you begin

Configure the SNMP community and trap target group.

Perform these steps to configure IPv6 functionality for SNMP.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco SNMP** from the list of templates.
4. In the **Trap** section, create or edit an SNMP trap target.

   Note the prerequisites for this procedure.
5. Configure these parameters.

Field	Description
VPN ID	Number of the VPN to use to reach the trap server. Range: 0 to 65530
IP Address	IP address of the SNMP server.
UDP Port	UDP port number for connecting to the SNMP server. Range: 1 to 65535
Trap Group Name	Name of a trap group configured in the **Group** tab.
User Name	Name of a community configured in the **Community** tab.
Source Interface	Interface to use to send traps to the SNMP server that is receiving the trap information.

Configure IPv6 functionality for SNMP using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure SNMP, according to these examples.

   This example permits any SNMP to access all objects with read-only permission using the community string named public. The device also sends Border Gateway Protocol (BGP) traps IPv6 host 3ffe:b00:c18:1::3/127 using SNMP v1.The community string named public is sent with the traps.

```
Device# config-transaction
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host 3ffe:b00:c18:1::3/127 public
```

   In this example, the SNMP context A is associated with the views in SNMPv2c group GROUP1 and the IPv6 named access list public2.

```
Device# config-transaction
Device(config)# snmp-server context A
Device(config)# snmp mib community-map commA context A target-list comm AVpn
Device(config)# snmp mib target list commAVpn vrf CustomerA
```

```
Device(config)# snmp-server view viewA ciscoPingMIB included
Device(config)# snmp-server view viewA ipForward included
Device(config)# snmp-server group GROUP1 v2c contextA read viewA write viewA
 notify access ipv6 public2
```

This example configures the IPv6 host as the notification server.

```
Device> enable
Device# config-transaction
Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2
Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6
Device(config)# snmp-server group publicv2c access ipv6 public2
Device(config)# snmp-server hosthost1.com2c vrf trap-vrf mgr
Device(config)# snmp-server user user1 bldg1 remote3ffe:b00:c18:1::3/127 v2c
 access ipv6 public2
Device(config)# snmp-server enable traps bgp
Device(config)# exit
```

Configure IPv6 functionality for a DHCP relay agent using templates

Before you begin

Perform these steps to configure IPv6 functionality for a DHCP relay agent.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco VPN Interface Ethernet** from the list of templates.
4. In the **Basic Configuration** section, select **IPv6**.
5. In the **DHCP Helper** area, click **Add** and configure these parameters.

Field	Description
DHCPv6 Helper #	IP address of the DHCP helper
DHCPv6 Helper VPN	VPN ID of the VPN source interface for the DHCP helper.

Configure IPv6 functionality for a DHCP relay agent using CLI commands

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a DHCP relay agent, according to this example.

```
device-configuration
interface GigabitEthernet8
 vrf forwarding 2
```

```
no ip address
ipv6 address 2001:A14:99::F/64
ipv6 dhcp relay destination vrf 1 2001:A14:19::12 GigabitEthernet2
```

Configure IPv6 functionality for ACL and QoS using templates

Before you begin

Perform these steps to configure IPv6 functionality for access control lists (ACL) and quality of service (QoS).

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco VPN Interface Ethernet** from the list of templates.
4. In the **ACL/QoS** section, configure these parameters.

Parameter Name	Description
Ingress ACL - IPv6	Click **On** to enable the IPv6 ingress access list.
IPv6 Ingress Access List	Enter the name of the IPv6 ingress access list.
Egress ACL - IPv6	Click **On** to enable the IPv6 egress access list.
IPv6 Egress Access List	Enter the name of the IPv6 egress access list.

Configure IPv6 functionality for ACL and QoS using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for access control lists (ACL) and quality of service (QoS).

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure ACL according to this example.

```
Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv6_acl
Device(config-access-list-ipv6_acl)#    sequence 11
Device(config-sequence-11)#     match
Device(config-match)#       source-ip 2001:380:1::64/128
Device(config-match)#      destination-ip 2001:3c0:1::64/128
Device(config-match)#      source-port      4000
Device(config-match)#      destination-port 3000
Device(config-match)#      traffic-class    6
Device(config-match)#      next-header      6
Device(config-match)#      packet-length    1000
Device(config-match)#     action accept
```

```
Device(config-action)#

Device(config)# sdwan interface GigabitEthernet6 ipv6 access-list ipv6_acl
in
Device(config-interface-GigabitEthernet6)#
Device(config-interface-GigabitEthernet6)#

Device(config)# policy lists data-ipv6-prefix-list source_ipv6_list
Device(config-data-ipv6-prefix-list-source_ipv6_list)# ipv6-prefix
2001:380:1::/64

Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv_ipv6_prefix
Device(config-access-list-ipv_ipv6_prefix)# sequence 11
Device(config-sequence-11)#     match
Device(config-match)#       source-data-prefix-list data-ipv6-prefix-list
Device(config-match)#       destination-data-prefix-list source_ipv6_list
Device(config-match)#      destination-ip 2001:3c0:1::64/128
Device(config-match)#      source-port      4000
Device(config-match)#      destination-port 3000
Device(config-match)#      traffic-class    6
Device(config-match)#      next-header      6
Device(config-match)#      packet-length    1000
Device(config-match)#     !
Device(config-match)#     action accept
```

3. Configure QoS according to this example.

```
class-map match-any class0
match qos-group 0
class-map match-any class1
match qos-group 1
!
policy-map qos_map_for_data_policy
class class0
  bandwidth percent 10
  random-detect
class class1
  bandwidth percent 10
  random-detect

policy
no app-visibility
class-map
  class class0 queue 0
  class class1 queue 1
!
ipv6
  access-list fwd_class_data_policy
   sequence 5
    match
     traffic-class 0
    !
    action accept
     count fwd_class_data_policycnt_5
     class class0
    !
   sequence 6
    match
     traffic-class 1
    !
```

```
    action accept
     count fwd_class_data_policycnt_6
     class class1
!
default-action drop
```

Configure IPv6 functionality for a logging host using templates

Before you begin

Perform these steps to configure IPv6 functionality for a logging host.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template** to select an appropriate device model.
3. Select **Cisco Logging** from the list of templates.
4. From **Server**, click **IPv6** and configure these parameters.

Field	Description
IPv6 Hostname/IPv6 Address	Host name or IP address of the server to direct the logging information.
VPN ID	VPN ID of the VPN source interface.
Source Interface	Name of the source interface.
Priority	Maximum severity of messages that are logged.

Configure IPv6 functionality for a logging host using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a logging host.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a logging host, according to this example.

```
config-transaction
Device(config)# logging host ipv6
AAAA:BBBB:CCCC:DDDD::FFFF
```

> **Note**
>
> Creating and deleting the logging host configurations in same transaction causes unexpected behavior. For example, deleting **logging host** *ipv6-address* and creating **logging host** *ipv6-address* **vrf** *vrf-name* configuration in same transaction causes both configurations to disappear from the device. Send the two requests in separate transactions.

Configure IPv6 functionality for a prefix list using templates

Before you begin

Perform these steps to configure IPv6 functionality for a prefix list.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. From the **Custom Options** drop-down list, select **Lists**. You can make this selection for a centralized policy or a localized policy.
3. Select **Prefix** from the list on the left and then select **New Prefix List**.
4. Click **IPv6** and enter the IPv6 address in **Add Prefix**.

> **Note**
>
> In SD-WAN Manager Release 20.9.5, changing or removing an IPv6 prefix can trigger a **Sync Now** prompt and cause the associated policy to be pushed again, even when the generated configuration has no changes. Beginning with release SD-WAN Manager 20.12.1, SD-WAN Manager no longer processes IP addresses and prefix lists, and this behavior does not occur.

Configure IPv6 functionality for a prefix list using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a prefix list.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a prefix list, according to this example.

```
config-transaction
Device(config)# policy
Device(config-policy)# ipv6
Device(config-ipv6)# access-list ipv6_acl
Device(config-access-list-ipv6_acl)#  sequence 11
```

```
Device(config-sequence-11)#  match
Device(config-match)# source-ip 2001:DB8:1::64/128
Device(config-match)# destination-ip 2001:DB8:1::64/128
```

Configure IPv6 functionality for a data prefix using templates

Before you begin

Perform these steps to configure IPv6 functionality for a data prefix.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. From the **Custom Options** drop-down list, select **Lists**. You can make this selection for a centralized policy or a localized policy.
3. Select **Data Prefix** from the list on the left and then select **New Data Prefix List**.
4. From **Internet Protocol**, click **IPv6** and enter the IPv6 address in **Add Prefix**.

Configure IPv6 functionality for a data prefix using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a data prefix.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a data prefix, according to this example.

```
Device(config)# policy lists data-ipv6-prefix-list source_ipv6_list
Device(config-data-ipv6-prefix-list-source_ipv6_list)# ipv6-prefix
2001:DB8:1::/64
```

Configure IPv6 functionality for a centralized policy using templates

Before you begin

Perform these steps to configure IPv6 functionality for a centralized policy.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. From the **Custom Options** drop-down list, select **Traffic Policy** under **Centralized Policy**.
3. Select **Traffic Data**.
4. Select **Add Policy** and click **Create New**.
5. Click **Sequence Type** and then select **Traffic Engineering**.
6. Click **Sequence Rule**.

7. From the **Protocol** drop-down list, select **IPv6** to apply the policy only to IPv6 address families, or select **Both** to apply the policy IPv4 and IPv6 address families.
8. Click **Sequence Type** and then select **QoS**.
9. Click **Sequence Rule**.
10. From the **Protocol** drop-down list, click **IPv6** to apply the policy only to IPv6 address families, or select **Both** to apply the policy IPv4 and IPv6 address families.

Configure IPv6 functionality for a centralized policy using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a centralized policy.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a IPv6 in a centralized policy, according to this example.

```
config-transaction
(config)# policy
(config-policy)# lists ipv6-prefix-list foo ipv6-prefix 1::1/64
                  ipv6-prefix-list ipv6-1
                  ipv6-prefix 1::1/128
```

Configure IPv6 functionality for a localized policy using templates

Before you begin

Perform these steps to configure IPv6 functionality for a localized policy.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Policies**.
2. From the **Custom Options** drop-down list, select **Access Control Lists** under **Localized Policy**.
3. Click **Add Access Control List Policy** and choose **Add IPv6 ACL Policy**. The policy you create applies only to IPv6 address families.

Configure IPv6 functionality for a localized policy using CLI commands

Before you begin

Perform these steps to configure IPv6 functionality for a localized policy.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Configure a IPv6 in a localized policy, according to this example.

The example matches IPv6 routes that have addresses specified by the prefix list called marketing.

```
config-transaction
Device(config)# route-map name
Device(config-route-map)# match ipv6 address prefix-list marketing
```

12 IPv6 in a Dual Stack Environment

Topics:

- Dual stack environment with IPv6 as the preferred address family
- Dual stack environment with IPv6 as the preferred address family
- Methods for configuring a dual stack environment with IPv6 as the preferred address family
- Monitor the use of IPv6 as the preferred address family in a dual stack environment, in SD-WAN Manager
- Monitoring IPv6 as the preferred address family in a dual stack environment

Introduces the concept and requirements of leveraging IPv6 within a dual stack network, establishing foundational context for subsequent configuration and monitoring procedures.

Dual stack environment with IPv6 as the preferred address family

Explains the architecture and benefits of prioritizing IPv6 in dual stack environments, detailing network behaviors and operational considerations for implementation.

Table 38: Feature History

Feature Name	Release Information	Description
IPv6 as Preferred Address Family in a Dual Stack Environment	Cisco IOS XE Catalyst SD-WAN Release 17.10.1a Cisco Catalyst SD-WAN Control Components Release 20.10.1	You can select IPv6 as the preferred address family for control and data connections in a dual stack network environment. For Cisco SD-WAN Manager and Cisco Catalyst SD-WAN Controller, configure IPv6 as the preferred address family by using the feature template or the CLI template. For Cisco IOS XE Catalyst SD-WAN devices, configure IPv6 as the preferred address family using the configuration groups, the Quick Connect workflow, or by CLI commands executed through a CLI add-on profile or template.

Dual stack environment with IPv6 as the preferred address family

Describes essential concepts and principles for using IPv6 as the primary address family in dual stack setups, covering deployment scenarios and network impacts.

A dual stack environment is a network that supports both IPv4 and IPv6 addressing, and where both can be used simultaneously. In a dual stack Cisco Catalyst SD-WAN environment, you can select a preferred address family, either IPv4 or IPv6, for establishing control and data connections.

Control connections

You can select a preferred address family, IPv4 or IPv6, to establish control and data connections in a dual stack network environment. Use the **Dual Stack IPv6 Default** drop-down list in Cisco SD-WAN Manager to set IPv6 or IPv4.

- **Dual Stack IPv6 Default**: **True**:

  The device establishes IPv6 control connections with SD-WAN Manager and the SD-WAN Controller that the device is connected to.

- **Dual Stack IPv6 Default**: **False**:

  The device establishes IPv4 control connections with SD-WAN Manager and the SD-WAN Controller that the device is connected to.

Data connections

Data connections or Bidirectional Forwarding Detection (BFD) sessions are established based on the IPv6 option set in local, remote devices. In a dual stack network environment, if you choose **True** for a local or remote device, the BFD session is an IPv6 connection. Otherwise, it is IPv4.

- **Dual Stack IPv6 Default**: **True**:

  Establishes IPv6 BFD sessions.

- **Dual Stack IPv6 Default**: **False**:

  Establishes IPv4 BFD sessions.

SD-WAN Validator

The **Dual Stack IPv6 Default** drop-down list options applies to devices, SD-WAN Manager, and SD-WAN Controller, but not to SD-WAN Validator.

The connections from SD-WAN Manager, SD-WAN Controller, and devices to the SD-WAN Validator are always dual (IPv4 and IPv6) in a dual stack network environment, regardless of the **Dual Stack IPv6 Default** setting.

NAT44 and NAT66

You can configure an IPv6 connection on devices that are at sites behind NAT44 and NAT66.

Benefits of setting IPv6 as the preferred address family

You have the option to migrate from IPv4 to IPv6, which allows you to have more IP addresses compared to IPv4. With IPv6, there can be no depletion of IP addresses.

Methods for configuring a dual stack environment with IPv6 as the preferred address family

Guides configuration of dual stack environments with IPv6 preference, providing instructions for device setup via configuration groups, Quick Connect workflows, CLI commands, and templates for SD-WAN Manager and Controller platforms.

These are methods of configuring devices and SD-WAN Control Components for IPv6 connections.

Configuring devices for IPv6 connectivity

- Configure IPv6 as the preferred address family for devices, using a configuration group on page 138
- Configure IPv6 as the preferred address family for devices, using the Quick Connect workflow on page 138
- Configure IPv6 as the preferred address family for devices, using CLI commands on page 139

Configuring SD-WAN Control Components for IPv6 connectivity

- Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using CLI commands on page 140
- Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using templates on page 140

Configure IPv6 as the preferred address family for devices, using a configuration group

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using a configuration group.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Click **Configuration Groups**.
3. Create a new configuration group or open an existing one to display the full details of the configuration group.
4. In the **Deployment** area, view the devices associated with the configuration group. You can add devices if desired.
5. Click **Deploy** to deploy the configuration to the associated devices.
6. In **Process Overview**, click **Next**.
7. On the **Selected Devices to Deploy** page, select or deselect devices and click **Next**.
8. For **Dual Stack IPv6 Default**, select **True** to set IPv6 as the default connection type, and click **Next**.

   Here's how **Dual Stack IPv6 Default** affects control connections and BFD sessions:

   - **True**: Devices establish an IPv6 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv6.
   - **False**: Devices establish an IPv4 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv4.

   The connections from devices to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.
9. On the summary page, click **Deploy**.

What to do next

For more information on using configuration groups, see Configuration Groups and Feature Profiles.

Configure IPv6 as the preferred address family for devices, using the Quick Connect workflow

For information about the workflow, see Quick Connect Workflow.

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using the Quick Connect workflow.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Workflows** > **Quick Connect**.
2. On the **Process Overview** page, click **Next**.
3. Select an option to sync your devices, and click **Next**.
4. On the **Selected devices to bring up** page, select devices, and click **Next**.
5. In the **Dual Stack IPv6 Default** drop-down list, select **True** to set IPv6 as a default connection. Click **Apply**, and then click **Next**.

   Here's how **Dual Stack IPv6 Default** affects control connections and BFD sessions:

- **True**: Devices establish an IPv6 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv6.
- **False**: Devices establish an IPv4 control connection with SD-WAN Manager and the SD-WAN Controller they are connected to. BFD sessions use IPv4.

The connections from devices to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.

6. On the **Summary** page, click **Deploy**.

Configure IPv6 as the preferred address family for devices, using CLI commands

Before you begin

Follow these steps to configure IPv6 as the preferred address family for Cisco IOS XE Catalyst SD-WAN devices in a dual stack environment, using CLI commands.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Enable IPv6 on the tunnel interface.

```
interface tunnel1
no shutdown
ipv6 enable
```

3. Enable IPv6.

```
system
ipv6-strict-control true
```

This example configures IPv6 as the preferred address family for devices.

```
interface Tunnel1
 no shutdown
 ip unnumbered GigabitEthernet1
 tunnel source GigabitEthernet1
 tunnel mode sdwan
 ipv6 enable
exit

system
gps-location latitude 32.0
gps-location longitude -100.0
system-ip 10.16.255.14
domain-id 1
site-id 400
ipv6-strict-control true
admin-tech-on-failure
organization-name "Cisco"
vbond vbond
```

Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using CLI commands

Before you begin

Follow these steps to configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller in a dual stack environment, using CLI commands.

Procedure

1. Create a CLI add-on profile or CLI add-on template.
2. Enable IPv6.

```
system
ipv6-strict-control true
```

This example configures IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controllers.

```
system
host-name vm9
system-ip 10.16.255.19
site-id 400
ipv6-strict-control true
port-offset 9
no daemon-restart
admin-tech-on-failure
no vrrp-advt-with-phymac
organization-name "Cisco"
vbond vbond
```

Configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller, using templates

Before you begin

Follow these steps to configure IPv6 as the preferred address family for SD-WAN Manager and SD-WAN Controller in a dual stack environment, using templates.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**, and click **Add Template**.
3. In the list of devices, select **Controller**.
4. Select **System** from the list of templates.
5. In the **Basic Information** section, for the **Dual Stack IPv6 Default** field, click **On** to enable.

   This sets IPv6 as a default connection.

   Here's how **Dual Stack IPv6 Default** affects connections:

   - **On**: SD-WAN Manager and SD-WAN Controllers establish an IPv6 control connection with other SD-WAN Manager and SD-WAN Controller instances.
   - **Off**: SD-WAN Manager and SD-WAN Controllers establish IPv4 control connections.

The connections from SD-WAN Manager and SD-WAN Controller instances to SD-WAN Validator are always dual (IPv4 and IPv6) in a dual IP stack environment, regardless of how**Dual Stack IPv6 Default** is configured.

6. Click **Save**.

Monitor the use of IPv6 as the preferred address family in a dual stack environment, in SD-WAN Manager

Details monitoring procedures for tracking IPv6 utilization as the primary address family in dual stack environments using SD-WAN Manager tools and dashboards.

After you configure IPv6 as the preferred address family for connections, the BFD connections will be up and running in SD-WAN Manager.

Before you begin

Follow these steps to view the BFD connections in SD-WAN Manager.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.
2. Verify the status of the connection in the **BFD** column.

Monitoring IPv6 as the preferred address family in a dual stack environment

Describes strategies and tools for observing and analyzing IPv6 activity within dual stack networks, focusing on verification and ongoing operational checks.

These commands are useful for monitoring the use of IPv6 addressing in a dual stack environment.

Cisco IOS XE Catalyst SD-WAN devices

- **show sdwan control connections**
- **show sdwan control local-properties**
- **show sdwan bfd sessions**
- **show sdwan omp tlocs**
- **show sdwan bfd tloc-summary-list**

SD-WAN Manager and SD-WAN Controller

- **show control connections**
- **show control local-properties**

13 DHCP for IPv6

Topics:

- Prerequisites for DHCP for IPv6
- Restrictions for DHCP for IPv6
- DHCP for IPv6
- Use cases for DHCP for IPv6
- Methods for configuring DHCP for IPv6 using CLI commands
- Methods for verifying a DHCPv6 client and server configuration using CLI commands

Introduces DHCP for IPv6, presenting a high-level reference to its features, historical context, and relevance within IPv6 networks.

Table 39: Feature History

Feature Name	Release Information	Description
DHCP for IPv6	Cisco IOS XE Catalyst SD-WAN Release 17.7.1a Cisco vManage Release 20.7.1	This feature allows you to configure DHCP for IPv6 (DHCPv6) on Cisco IOS XE Catalyst SD-WAN devices to assign IPv6 addresses to hosts on an IPv6-enabled network. Assigning of IPv6 addresses is accomplished using SLAAC, DHCPv6, DHCPv6 Prefix Delegation, or DHCPv6 Relay. A Cisco IOS XE Catalyst SD-WAN device can be configured for DHCPv6 as a DHCP server, DHCP client, or as a DHCP relay agent.

Prerequisites for DHCP for IPv6

Details essential prerequisites required before deploying DHCP for IPv6, including necessary configurations, dependencies, and foundational requirements.

Ensure basic IPv6 connectivity for assigning IPv6 addresses to hosts connected to the devices.

Restrictions for DHCP for IPv6

Explains restrictions and limitations associated with implementing DHCP for IPv6, highlighting operational boundaries, feature constraints, and compatibility

These are restrictions that apply to DHCP for IPv6.

- DHCP for IPv6 is supported only through CLI configuration.
- A unique DHCPv6 pool name must be provided for each VRF.

DHCP for IPv6

Describes the concept and operational principles of DHCP for IPv6, providing an overview of its roles, benefits, and integration within IPv6 environments.

DHCP for IPv6 (DHCPv6) is a protocol used by IPv6 devices to obtain IP addresses and configuration parameters from DHCPv6 servers. It operates similarly to DHCP for IPv4 but uses IPv6-specific mechanisms and messages.

You can configure Dynamic Host Configuration Protocol (DHCP) for IPv6 to assign addresses on an IPv6-enabled network. Alternatively, you can also configure Stateless Address Autoconfiguration (SLAAC) to assign addresses on an IPv6-enabled network.

SLAAC

The most common method for IPv6 client address assignment is SLAAC. SLAAC provides simple plug-and-play connectivity where hosts self-assign an address based on the IPv6 prefix.

SLAAC is configured like this:

- Host sends a router solicitation message.
- Hosts waits for a Router Advertisement (RA) message.
- Hosts take the first 64 bits of the IPv6 prefix from the RAmessage and combines it with the 64 bit EUI-64 address (in the case of ethernet, this is created from the MAC Address) to create a global unicast message. The host also uses the source IP address, in the IP header, of the RA message, as its default gateway.
- Duplicate Address Detection (DAD) is performed by IPv6 clients in order to ensure that random addresses that are picked do not collide with other clients.
- The choice of algorithm is up to the client and is often configurable.

The last 64 bits of the IP v6 address can be learned based on these two algorithms:

- EUI-64 which is based on the MAC address of the interface, or
- Private addresses that are randomly generated.

SLAAC and DHCPv6

DHCPv6	IPv6 devices use multicast to acquire IP addresses and to find DHCPv6 servers. The basic DHCPv6 client-server concept is similar to DHCP for IPv4. If a client wants to receive configuration parameters, it sends out a request on the attached local network to detect available DHCPv6 servers. The server responds with the requested information in a Reply message. The DHCPv6 client knows whether to use DHCPv6 based upon the instruction from a router on its link-local network. The default gateway has two configurable bits in its RA available for this purpose: • O bit–When this bit is set, the client can use DHCPv6 to retrieve other configuration parameters (for example, TFTP server addressor DNS server address) but not the client's IP address. • M bit–When this bit is set, the client can use DHCPv6 to retrieve a managed IPv6 address and other configuration parameters from a DHCPv6 server.
Stateless DHCP	Stateless DHCPv6 is a combination of SLAAC and DHCPv6. With this option SLAAC is still used to retrieve an IP address while DHCP is used to obtain additional information such as TFTP server addressor, DNS server address. In this case, the device sends an RA with the O bit set but does not set the M bit. This is known as Stateless DHCPv6 because the DHCPv6 server does not have to track the client address bindings.
Stateful DHCP	Stateful DCHPv6 functions exactly the same as DHCP IPv4 in which hosts receive both their IPv6 address and additional parameters from the DHCP server. When a device sends an RA with the M bit set, this indicates that clients must use DHCP to obtain their IP addresses. When the M bit is set, the setting of the O bit is irrelevant because the DHCP server also returns other configuration information together with the addresses. This is known as Stateful DHCPv6 because the DHCPv6 server tracks the client address bindings.

DHCPv6 prefix delegation

The DHCPv6 prefix delegation feature is a stateful mode of operation for simple delegation of prefixes from a delegating edge device (DHCP server) to requesting edge device (DHCP clients).

DHCPv6 prefix delegation feature is ideal for situations where:

- A delegating edge device that does not have the information about the topology of the networks to which the requesting edge device is attached to.
- A delegating edge device does not require other information apart from the identity of the requesting edge device to choose a prefix for delegation. This mechanism is appropriate for use by an ISP to delegate a prefix to a subscriber. After the ISP has delegated prefixes to a subscriber, the subscriber may further subnet and assign prefixes to the links within the subscriber's network.

DHCPv6 relay

A DHCPv6 relay agent is an edge device, residing on the client's network, is used to relay messages between the client and the server when a DHCPv6 server is not in the same network as the DHCPv6 clients.

Benefits

Configuring DHCP for IPv6 allows you to have more IP address compared to IPv4. With IPv6, there can be no depletion of IP addresses.

Use cases for DHCP for IPv6

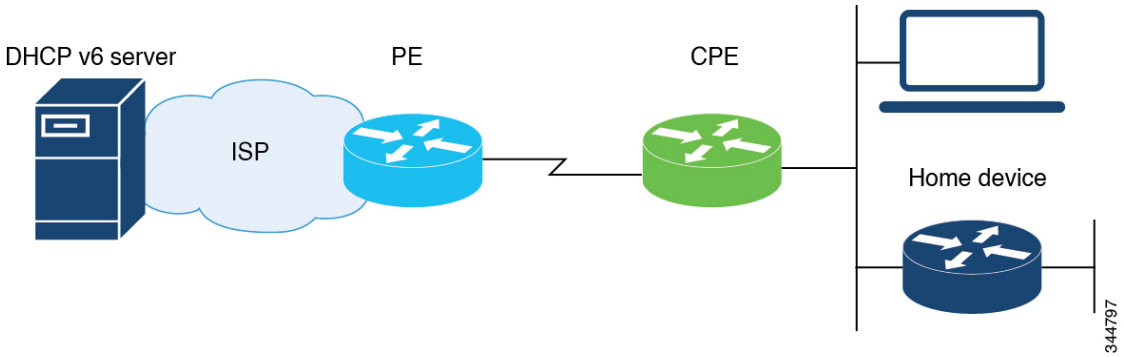
Outlines typical use cases for DHCPv6, illustrating scenarios where DHCP for IPv6 is deployed to meet various network needs and operational requirements.

These are use cases for DHCP for IPv6.

Cisco IOS XE Catalyst SD-WAN devices can be configured for DHCPv6 as a server, client, or a relay agent. As a server, a device can be configured for SLAAC, stateless DHCP, or for prefix delegation.

SLAAC with DHCP

The figure below shows a typical broadband deployment.



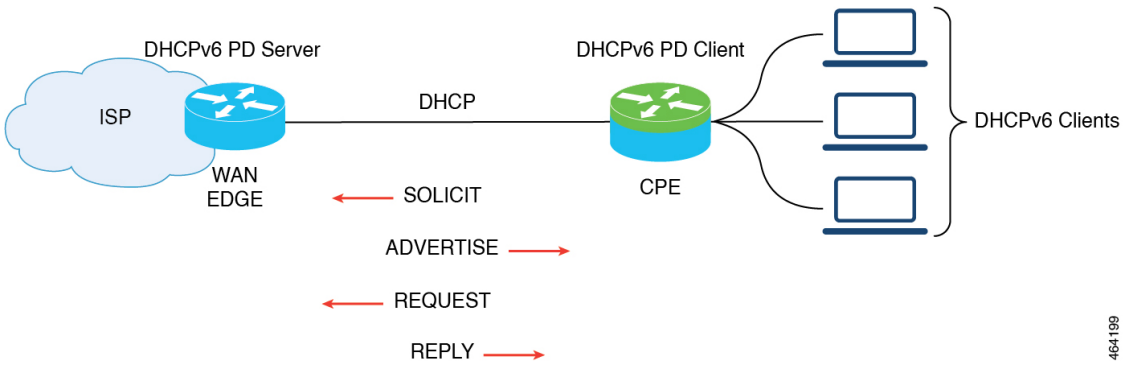
A Cisco IOS XE Catalyst SD-WAN device deployed on a customer premises (CPE) and connected to a ISP edge (PE) device can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server might provide configuration parameters such as Domain Name System (DNS) server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. Such information can be specific to ISPs.

In addition to being a DHCPv6 client (toward the ISP), the CPE can act as a DHCPv6 server to the home network. For example, neighbor discovery followed by a stateless or stateful DHCPv6 client can occur on the link between the CPE and the home devices. In some cases, the information to be provided to the home network is the same as that obtained from the ISP-side DHCPv6 server. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 Prefix Delegation

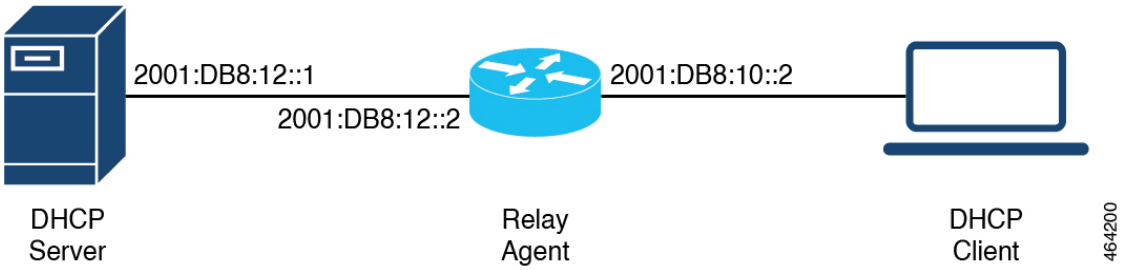
The model of operation for prefix delegation is as follows. In this sample topology, an edge device is configured as a DHCP server which is provisioned with prefixes to be delegated to a DHCP client. A Cisco IOS XE Catalyst SD-WAN device is configured as a DHCP client and requests prefix(es) from the server. The server chooses prefix(es) for delegation and responds with prefix(es) to the DHCP client. The DHCP client is then responsible for the delegated prefix(es).

For example, the client might assign a subnet from a delegated prefix to one of its interfaces and begin sending Router Advertisements for the prefix on that link. Each prefix has an associated preferred lifetime and valid lifetime, which constitute an agreement about the length of time over which the client is allowed to use the prefix. A client can request an extension of the lifetimes on a delegated prefix and is required to terminate the use of a delegated prefix if the valid lifetime of the prefix expires.



DHCPv6 Relay

In this sample topology, the DHCP server is not in the same network as DHCP client. A Cisco IOS XE Catalyst SD-WAN device residing on the client's network acts as a relay agent to relay messages between the client and the server.



Methods for configuring DHCP for IPv6 using CLI commands

Provides instructions for configuring DHCP for IPv6 through CLI commands, detailing the step-by-step procedures to set up client and server functions in IPv6 networks.

These are methods for configuring DHCP for IPv6 (DHCPv6) using CLI commands.

Use a CLI add-on profile or template to configure devices from Cisco SD-WAN Manager using CLI commands. The sections here provide examples of various DHCP configurations.

Configure SLAAC

This example shows how to configure SLAAC on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address autoconfig
device(config-if)# ipv6 enable
device(config-if)# end
```

This example shows how to configure SLAAC on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# end
```

Configure SLAAC and DHCPv6 pool for options

This example shows how to configure SLAAC and DHCPv6 pool on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address autoconfig
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure SLAAC and DHCPv6 pool on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd other-config-flag
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
```

Configure DHCPv6 (stateful) address assignment

This example shows how to configure DHCPv6 address assignment on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 address dhcp
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure DHCPv6 address assignment on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd managed-config-flag
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# address prefix 2010:AB8:0:1::1/64 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
```

```
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
```

Configure DHCPv6 with prefix delegation (stateful)

This example shows how to configure DHCPv6 with prefix delegation on the client side.

```
device(config)# interface GigabitEthernet0/0/2
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 dhcp client pd prefix_from_provider
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# end
```

This example shows how to configure DHCPv6 with prefix delegation on the server side.

```
device(config)# interface GigabitEthernet1
device(config-if)# ipv6 address 2010:AB8:0:1::1/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 nd autoconfig default-route
device(config-if)# ipv6 nd managed-config-flag
device(config-if)# ipv6 nd ra interval 20
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# prefix-delegation pool dhcpv6-pool1 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:DB8:3000:3000::42
device(config-dhcpv6)# domain-name example.com
device(config-dhcpv6)# vendor-specific 100
device(config-dhcpv6)# suboption 1 address 2001:CC:1234:44::10
device(config-dhcpv6)# suboption 2 ascii "ip phone"
device(config)# ipv6 local pool dhcpv6-pool1 2001:DB8:1200::/40 48
```

Configure DHCPv6 with relay

This example shows how to configure DHCPv6 with relay on the client side.

```
device(config)# interface GigabitEthernet3
device(config-if)# ipv6 address dhcp
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp client pd pr-from-pd
device(config-if)# ipv6 dhcp client request vendor
device(config-if)# no mop enabled
device(config-if)# no mop sysid
device(config-if)# end
```

This example shows the configurations on the client facing a WAN edge device that acts as the relay agent.

```
device(config)# interface TenGigabitEthernet0/0/5
device(config-if)# vrf forwarding 10
device(config-if)# load-interval 30
device(config-if)# ipv6 address 2001:BB:1000::10/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp relay destination 2001:BB8:1200::2
```

```
device(config-if)# ipv6 dhcp relay option vpn
device(config-if)# end
```

This example shows the configurations on the server facing WAN edge device.

```
device(config)# interface GigabitEthernet0/0/3
device(config-if)# vrf forwarding 10
device(config-if)# no ip address
device(config-if)# negotiation auto
device(config-if)# ipv6 address 2001:BB8:1200::1/64
device(config-if)# ipv6 enable
device(config-if)# end
```

This example shows how to configure DHCPv6 with relay on the server side.

```
device(config)# interface GigabitEthernet2
device(config-if)# ipv6 address 2001:BB8:1200::2/64
device(config-if)# ipv6 enable
device(config-if)# ipv6 dhcp server dhcpv6
device(config-if)# end

device(config)# ipv6 dhcp pool dhcpv6
device(config-dhcpv6)# prefix-delegation pool dhcpv6-pool10 lifetime infinite
 infinite
device(config-dhcpv6)# address prefix 2001:BB:1000::/64 lifetime 200 200
device(config-dhcpv6)# dns-server 2001:BB:1200::42
device(config-dhcpv6)# domain-name relay.com
device(config)# ipv6 local pool dhcpv6-pool10 8001:ABCD::/40 48
```

Methods for verifying a DHCPv6 client and server configuration using CLI commands

Guides users through verifying DHCPv6 client and server configurations using CLI commands, covering essential validation steps to ensure correct deployment and operation.

These are methods of verifying DHCP for IPv6 (DHCPv6) configurations.

Verify DHCPv6 interface information

This is a sample output from the **show ipv6 dhcp interface** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is IDLE
  Address State is OPEN
  Renew for address will be sent in 00:01:09
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      IA NA: IA ID 0x00080001, T1 100, T2 160
        Address: 2010:AB8:0:1:95D1:CFC:F227:23FB/128
                 preferred lifetime 200, valid lifetime 200
                 expires at Oct 26 2021 07:28 AM (170 seconds)
      DNS server: 2001:DB8:3000:3000::42
```

```
      Domain name: example.com
      Information refresh time: 0
      Vendor-specific Information options:
          Enterprise-ID: 100
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

This is a sample output from the **show ipv6 dhcp interface** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is OPEN
  Renew will be sent in 00:01:34
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00080001, T1 100, T2 160
        Prefix: 2001:DB8:1202::/48
                preferred lifetime 200, valid lifetime 200
                expires at Oct 26 2021 07:30 AM (194 seconds)
      DNS server: 2001:DB8:3000:3000::42
      Domain name: example.com
      Information refresh time: 0
  Prefix name: prefix_from_server
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

This is a sample output from the **show ipv6 dhcp interface** command that provides details about SLAAC with DHCP.

```
Device# show ipv6 dhcp interface GigabitEthernet0/0/2
GigabitEthernet0/0/2 is in client mode
  Prefix State is IDLE (0)
  Information refresh timer expires in 23:59:49
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::250:56FF:FEBD:DBD1
    DUID: 00030001001EBD43F800
    Preference: 0
    Configuration parameters:
      DNS server: 2001:DB8:3000:3000::42
      Domain name: example.com
      Information refresh time: 0
      Vendor-specific Information options:
          Enterprise-ID: 100
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

View DHCPv6 Pool information

This is a sample output from the **show ipv6 dhcp pool** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp pool
DHCPv6 pool: relay_server
  VRF 10
  Prefix pool: dhcpv6-pool2
  Address allocation prefix: 5001:DB8:1234:42::/64 valid 20000 preferred 20000
 (1 in use, 0 conflicts)
```

```
              preferred lifetime 200, valid lifetime 200
  DNS server: 2001:BB8:3000:3000::42
  Domain name: relay.com
  Information refresh: 60
  Vendor-specific Information options:
  Enterprise-ID: 10
    suboption 1 address 2001:DB8:1234:42::10
    suboption 2 ascii 'ip phone'
  Active clients: 1
  Pool is configured to include all configuration options in REPLY
```

This is a sample output from the **show ipv6 dhcp pool** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp pool
DHCPv6 pool: relay_server
  VRF 10
  Prefix pool: dhcpv6-pool2
  Address allocation prefix: 5001:DB8:1234:42::/64 valid 20000 preferred 20000
 (0 in use, 0 conflicts)
              preferred lifetime 200, valid lifetime 200
  DNS server: 2001:BB8:3000:3000::42
  Domain name: relay.com
  Information refresh: 60
  Vendor-specific Information options:
  Enterprise-ID: 10
    suboption 1 address 2001:DB8:1234:42::10
    suboption 2 ascii 'ip phone'
  Active clients: 1
  Pool is configured to include all configuration options in REPLY
```

View DHCPv6 bindings

This is a sample output from the **show ipv6 dhcp binding** command that provides details about DHCPv6 address allocation.

```
Device# show ipv6 dhcp binding
Client: FE80::250:56FF:FEBD:8261
  DUID: 00030001001EE6DBF500
  Username : unassigned
  VRF : 10
  IA NA: IA ID 0x00080001, T1 10000, T2 16000
    Address: 5001:DB8:1234:42:500C:B3FA:54A7:F63D
            preferred lifetime 20000, valid lifetime 20000
            expires at Oct 26 2021 01:17 PM (19925 seconds)
```

This is a sample output from the **show ipv6 dhcp binding** command that provides details about DHCPv6 prefix delegation.

```
Device# show ipv6 dhcp binding
Client: FE80::250:56FF:FEBD:8261
  DUID: 00030001001EE6DBF500
  Username : unassigned
  VRF : 10
  Interface : GigabitEthernet0/0/3
  IA PD: IA ID 0x00080001, T1 100, T2 160
    Prefix: 2001:BB8:1602::/48
            preferred lifetime 200, valid lifetime 200
            expires at Oct 26 2021 08:01 AM (173 seconds)
```

View DHCPv6 database

This is a sample output from the **show ipv6 dhcp database** command.

```
Device# show ipv6 dhcp database
Database agent bootflash:
  write delay: 300 seconds, transfer timeout: 300 seconds
  last written at Oct 26 2021 08:01 AM, write timer expires in 250 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 2
  failed write times 0
```

View DHCPv6 relay bindings

This is a sample output from the **show ipv6 dhcp relay bindings** command that provides details about DHCPv6 relay.

```
Device# show ipv6 dhcp relay binding

Relay Bindings associated with default vrf:

Relay Bindings associated with vrf 10:
Prefix: 2001:AA8:1100::/48 (GigabitEthernet3)
  DUID: 00030001001E49674C00
  IAID: 851969
  lifetime: INFINITE
  expiration: INFINITE
Summary:
  Total number of Relay bindings = 1
  Total number of IAPD bindings = 1
  Total number of IANA bindings = 0
  Total number of Relay bindings added by Bulk lease = 0
```

14 Per Packet Load Balancing

Topics:

- Feature history for per packet load balancing
- Per packet load balancing
- Configure PPL
- Monitor PPL

Introduces Per Packet Load Balancing in Cisco SD-WAN, presenting an overview of the feature, its operational mechanism, benefits, supported platforms, typical use cases, and critical restrictions for deployment.

Feature history for per packet load balancing

Lists the development milestones and release information for PPL, including feature description.

This table describes the developments of this feature, by release.

Table 40: Feature history

Feature name	Release information	Description
Per packet load balancing	Cisco IOS XE Catalyst SD-WAN Release 26.1.x Cisco Catalyst SD-WAN Manager Release 26.1.x	With this feature, Cisco SD-WAN sends packets from a single flow across multiple WAN links, maximizing bandwidth and maintaining performance by reordering packets at the destination.

Per packet load balancing

Describes how PPL improves network bandwidth and performance by distributing and reordering packets across multiple WAN links, unlike traditional load balancing methods.

Per-packet load balancing (PPL) or adaptive single flow distribution is a bandwidth aggregation feature that

- distributes packets from a single flow across multiple WAN links,
- maximizes the use of all available paths, and
- maintains data integrity by reordering packets at the destination.

Traditional load balancing vs PPL

Traditional load balancing uses one network link for each data stream, which can limit overall bandwidth use. PPL shares traffic across multiple available links, allowing you to get the most out of your network and improve performance.

Reorder packets

When a device distributes packets over multiple links, packets may arrive at the destination out of order. The PPL feature automatically reorders these packets on a best-effort basis and releases any buffered packets based on either a time threshold or a memory threshold.

Key concepts for understanding PPL

Explains how PPL efficiently distributes traffic across multiple WAN tunnels by dynamically selecting the best paths based on latency, flowlets, and network policies.

Sender and receiver

The sender is the WAN edge device that sends PPL traffic. The receiver is the WAN edge device that receives this traffic and may reorder packets if they arrive out of order.

Traffic selection

Defines which specific traffic from the service VPNs use PPL, enabling detailed control through network policies. You can configure this using data policies in the "from-service" direction, and any traffic matched in the policy's match clause becomes eligible for PPL.

Inter-tunnel latency measurement

PPL measures the delay across all available tunnels and selects the tunnel with the lowest latency as a reference point. It compares other tunnels to this reference and includes only those within a set latency range for traffic distribution. This method is a Cisco proprietary algorithm. It ensures efficient and reliable path selection for your network traffic.

Flow splitting

Instead of splitting traffic into individual packets, PPL groups packets into flowlets, which are small sets of packets. You create flowlets either after sending a set number of packets or when you detect natural breaks (packet gaps) in the traffic.

Packet gap

Packet gap is a method used in PPL to detect natural breaks in a flow of traffic. When there is an idle period between bursts of packets that exceeds the maximum latency difference among available tunnels, a new flowlet is started and sent on a different tunnel.

Candidate tunnel group (CTG)

The group of tunnels available for PPL, selected first by “local colors” and then filtered using metrics like latency and loss. A maximum of eight tunnels can be part of the CTG.

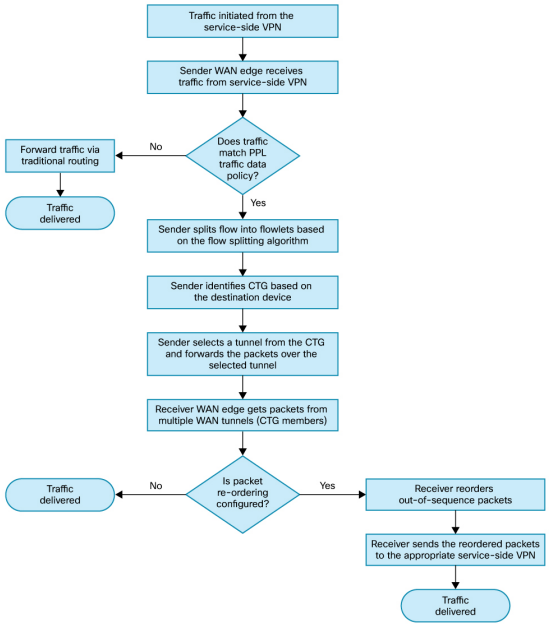
Path selection

Determines how each flowlet is sent across the reasonably available tunnels in the CTG, using round-robin distribution.

Sequence of events for PPL

The sender device receives traffic from service-side VPNs and checks it against the traffic data policy for PPL rules. If matched, the sender splits the flow into flowlets and forwards them over reasonably available tunnels in the Candidate Tunnel Group. The receiver device collects and re-orders the flowlets as needed, then forwards the reordered flows into the appropriate service VPNs.

See the illustration to understand the sequence of events.



Benefits of PPL in Cisco SD-WAN

Benefits of PPL:

- Adaptibility: Adjusts quickly to changes in network conditions to keep performance steady.

- Bandwidth and resource utilization: Combines the capacity of multiple WAN links for higher data transfer rates.

  Uses all available network paths to make the optimal usage of bandwidth.

- Performance: Increases speed and efficiency for elephant flows.

  Supports applications that need more bandwidth than a single link can provide.

Use cases for PPL

Scenario 1: Managing AI traffic between offices and data centers

AI tasks often involve sending large amounts of information from local offices to a central data center for processing. At the same time, users need quick, real-time responses from AI tools. This traffic is unpredictable and comes in sudden bursts. When all this data is forced through a single network path, it can create network congestion, leading to slow AI performance and delayed data uploads.

How does PPL help?

PPL breaks these large data streams into smaller, manageable pieces and sends them across all available network connections at once. This prevents any single connection from becoming overloaded.

Scenario 2: High-volume workload transfers and backups in enterprise environments

Enterprise environments routinely perform large-scale data transfers for activities such as workload upgrades, Windows operating system or application updates, data center backups, and disaster recovery operations. These scenarios often require moving significant amounts of data between endpoints. Such transfers can easily exceed the capacity of individual WAN circuits, especially when using multiple mid-sized links.

How does PPL help?

PPL splits large, single flows into multiple flowlets and distributes them across all available WAN circuits. This approach maximizes total available bandwidth, prevents any single link from being overwhelmed, and ensures that even deprioritized or non-critical traffic, like updates and backups, completes efficiently without impacting business-critical applications.

Supported platforms for PPL

These tables outline the supported hardware models and their corresponding DRAM requirements for Receiver and Sender PPL roles.

Table 41: Receiver PPL supported platforms

Model	Minimum DRAM
C8500-12X4QC	32 GB
C8500-20X6C	64 GB
C8570-G2	32 GB
C8300-2N2S-6T	16 GB
C8300-2N2S-4T2X	16 GB
C8455-G2 (Supported in Cisco IOS XE Catalyst SD-WAN Release 26.2.x and Cisco Catalyst SD-WAN Manager Release 26.2.x onwards.)	32 GB

Table 42: Sender PPL supported platforms

Model	Minimum DRAM
C8300-2N2S-4T2X	16 GB
C8300-2N2S-6T	16 GB
C8500L-8S4X	15 GB
C8475-G2	32 GB
C8455-G2	32 GB
C8500-12X4QC	16 GB
C8500-12X	16 GB
C8550-G2	32 GB

Note

Receiver platforms can act as senders, but sender platforms cannot act as receivers.

Restrictions for PPL

Restrictions for PPL:

- Maximum senders: A receiver can only support up to 256 senders.
- Maximum receivers: A sender can only support up to 512 receivers.
- PPL data policy: You can configure PPL data policy action only in the ‘from-service’ direction.

  Do not use packet duplication (packet dup) or forward error correction (FEC) actions with PPL.
- SLA classes: Limit SLA classes to 14 to reserve one for PPL. If 15 SLA classes are configured, remove one and reload the device before configuring PPL.
- Multicast traffic: Multicast traffic is not supported with PPL.
- Service insertion: Service insertion is not supported with PPL.
- Parallel routing configurations: Sender cannot be configured with any parallel routing decisions such as next-hop, and remote-TLOC through data policy when PPL is enabled.

Configure PPL

Provides configuration procedures for Per Packet Load Balancing, including setup using configuration groups, data policy definitions with policy groups, and adjustment of packet and latency parameters via CLI commands.

Use one of these procedures to configure PPL.

- Configure PPL using configuration groups
- Configure a data policy with load balancing using policy groups
- Configure maximum packet value in PPL using CLI commands

- Configure latency offset value in PPL using CLI commands

Configure PPL using configuration groups

Use these steps to configure PPL using configuration groups.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration groups**.
2. Create or edit a System profile.
3. Navigate to Basic Settings and select **Per Packet Load Balancing**.
4. Enable **PPL** on the Sender and configure its parameters mentioned below.
5. Enable packet reordering on the receiver.

Table 43: PPL fields

Field	Description
Local colors	Specify the WAN links (such as MPLS, Internet, and LTE) used to distribute traffic within the network.
Flow splitting	Specify how to divide traffic among the available tunnels. **Packets:** Sends a set number of packets (default: 1024) through one tunnel before switching to the next flowlet. This cycle repeats continuously to balance the network load. To set a maximum packet value, refer toConfigure maximum packet value in PPL using CLI commands **Packet gap:** Switches tunnels based on the time interval between packets. If the time gap between bursts exceeds the maximum latency difference among tunnels, a new flowlet is sent on a different tunnel. This helps distribute traffic more effectively during natural pauses in data transmission.
Candidate tunnel group	Dynamically defines the group of tunnels used in a PPL network. All tunnels: All available tunnels become part of PPL network. Latency offset: Includes only those tunnels whose latency is within a configurable threshold of the lowest-latency tunnel. To set latency offset value, refer to Configure latency offset value in PPL using CLI commands. Adaptive cluster: Groups tunnels dynamically into clusters based on real-time loss and latency measurements. Tunnels with similar loss (primary factor) and latency (secondary factor) are grouped together for packet distribution.
Reorder	Enables packet reordering at the receiving end to reduce out-of-order delivery. If you disable this option, ensure that your application can independently handle packet reordering.

Configure a data policy with load balancing using policy groups

Use these steps to configure data policy with load balancing.

Procedure

1. From the Cisco SD-WAN Manager, choose **Configuration** > **Policy Groups** > **Application Priority and SLA**.
2. Create or edit an existing policy.
3. Add or create a new Traffic Policy within the policy.
4. Set the direction to **From Service**.
5. In the Match Clause, specify the traffic to be matched, such as the DSCP source.
6. In the Action section, select the **radio button** to enable Load Balancing .
7. From the **Load Balancing Algorithm** drop-down list, select **per packet**.
8. Select **Save**.

Configure maximum packet value in PPL using CLI commands

Use these steps to configure the maximum packet value in PPL.

Procedure

1. Enter configuration mode.

   Example

   ```
   config-transaction
   ```

2. Set the maximum number of packets in a flowlet.

   Example

   ```
   sdwan ppl flow-splitting max-pkts 2048
   ```

   Enter a value between 1024 and 10240 to define the maximum packet count for a flowlet.

Here's the complete configuration example for configuring maximum packet value in PPL.

```
sdwan
 ppl
  enable
  flow-splitting
   max-pkts 2048
   !
  !
 !
```

Configure latency offset value in PPL using CLI commands

Use these steps to configure the latency offset value in PPL.

Procedure

1. Enter configuration mode.

   Example

   ```
   config-transaction
   ```

2. Configure the latency offset value for the CTG algorithm.

   Example

   ```
   sdwan ppl ctg algo latency-offset 15
   ```

   Enter the desired latency offset value to be used by the PPL CTG algorithm.

Here's the complete configuration example for configuring maximum packet value in PPL.

```
sdwan
 ppl
  ctg
    algo
     latency-offset 15
     !
    !
  !
!
```

Monitor PPL

Guides monitoring of Per Packet Load Balancing, enabling verification of feature status, operational metrics, and effective traffic distribution within Cisco SD-WAN networks.

Use these show commands to monitor PPL.

To verify the configured flow splitting and Candidate Tunnel Group (CTG) algorithms on a sender, use the **show platform software sdwan ftmd ppl cfg** command.

```
sender# show platform software sdwan ftmd ppl cfg
PPL Config
 Enable             : true
 Local Color List   : all-colors
 Path Selection     : round-robin
 OWL
   Interval : 20 sec
   DSCP     : 48 (0x30)
 Flow Splitting
   Algo     : pkt-gap adaptive

 CTG
   Algo     : adaptive-cluster
```

To verify if the PPL configuration flag is enabled for specific sites on a device, use the **show sdwan ftm site-db** command.

```
sender# show sdwan ftm site-db


  Site ID: 400, OD: No, status: Active, sys_cnt: 2, status_change: cnt
 2time 04:14:19:532, last_stats 0, is_pending_tloc_re
play=0, replay_cnt=1 Idle Timer: Not inited

  Remote    system: 172.16.255.41, OD: No, status: Active, tloc_cnt: 4,
 tun_cnt: 16, glean_ipc_cnt: 0, is_pending_tloc_repl
ay=0, replay_cnt=103, pkt_ro_ctx_id: 1, ppl_ctx_id: 1 ppl_cfg: True
ppl_adaptive_flowlet_gap: 1000
        TLOC   :: 172.16.255.41 : bronze : ipsec, nh_index: 32772, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : private2 : ipsec, nh_index: 32776, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : mpls : ipsec, nh_index: 32782, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.41 : biz-internet : ipsec, nh_index: 32784, OD:
No, status: Inactive
<snip>
  Remote    system: 172.16.255.40, OD: No, status: Active, tloc_cnt: 5,
 tun_cnt: 20, glean_ipc_cnt: 0, is_pending_tloc_repl
ay=0, replay_cnt=177, pkt_ro_ctx_id: 2, ppl_ctx_id: 2 ppl_cfg: True
 ppl_adaptive_flowlet_gap: 1000
        TLOC   :: 172.16.255.40 : mpls : ipsec, nh_index: 32773, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : private1 : ipsec, nh_index: 32775, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : biz-internet : ipsec, nh_index: 32777, OD:
No,
 status: Inactive
        TLOC   :: 172.16.255.40 : private2 : ipsec, nh_index: 32783, OD: No,
 status: Inactive
        TLOC   :: 172.16.255.40 : bronze : ipsec, nh_index: 32785, OD: No,
 status: Inactive
        Tunnel : 10.1.61.40/12346->10.1.41.40/12346 proto 0x32, idx 11,
sys 172.16.255.40, nh_index 65546
```

To verify Candidate Tunnel Group (CTG) selection for a destination site, use the **show sdwan bfd sessions alt** command.

```
sender# show sdwan bfd sessions alt
*Sus = Suspend
*GREinUDP = GREinUDP encap
*EAAR  = Enhanced Application-Aware Routing
*PPL = Per Packet Loadbalancing
*NA  = Flag Not Set
                                          SOURCE TLOC      REMOTE TLOC
                           DST PUBLIC               DST PUBLIC
SYSTEM IP        SITE ID     STATE       COLOR            COLOR
SOURCE IP                     IP                      PORT        ENCAP  BFD-LD
      FLAGS                UPTIME
------------------------------------------------------------------------------
172.16.255.40    400          up          mpls             biz-internet
10.1.61.40                    10.1.40.40              12346       ipsec  20019
       PPL                   13:01:01:32
172.16.255.40    400          up          mpls             mpls
10.1.61.40                    10.1.41.40              12346       ipsec  20022
       NA                    13:01:01:31
```

```
172.16.255.40    400         up          mpls             private1
10.1.61.40                   10.1.42.40              12346       ipsec  20020
       PPL                  13:01:01:31
172.16.255.40    400         up          mpls             bronze
10.1.61.40                   10.1.43.40              12346       ipsec  20021
       PPL                  13:01:01:32
172.16.255.40    400         up          biz-internet     biz-internet
20.1.60.40                   10.1.40.40              12346       ipsec  20001
       PPL                  13:01:01:45
172.16.255.40    400         up          biz-internet     mpls
20.1.60.40                   10.1.41.40              12346       ipsec  20004
       PPL                  13:01:01:45
172.16.255.40    400         up          biz-internet     private1
20.1.60.40                   10.1.42.40              12346       ipsec  20002
       PPL                  13:01:01:45
172.16.255.40    400         up          biz-internet     bronze
20.1.60.40                   10.1.43.40              12346       ipsec  20003
       PPL                  13:01:01:45
172.16.255.40    400         up          biz-internet     private2
20.1.60.40                   10.1.44.40              12346       ipsec  20006
       PPL                  13:01:01:45
```

To verify absolute Candidate Tunnel Group (CTG) data and counters, use the **show platform software sdwan ftmd ppl ctg** command.

```
sender# show platform software sdwan ftmd ppl ctg
OWL-REQ RX TS: Timestamp at which the remote device received the OWL request.
 In brackets: Latency relative to tunnel with lowest OWL.
--------------------------------------------------------------------------------
LOCAL COLOR     SYSTEM-IP       REMOTE COLOR    BFD-LD    PPL-ID    PKTRO-ID
   EPOCH-ID    OWL-REQ RX TS                PKT-TX      BYTES-TX    PKT-RX
   BYTES-RX
--------------------------------------------------------------------------------
biz-internet    172.16.255.40   biz-internet    20001     1         65535
   55905       1119300231(0)                0           0           0
    0
biz-internet    172.16.255.40   private1        20002     1         65535
   55905       1119300231(0)                0           0           0
    0
biz-internet    172.16.255.40   bronze          20003     1         65535
   55905       1119300231(0)                0           0           0
    0
biz-internet    172.16.255.40   mpls            20004     1         65535
   55905       1119300231(0)                0           0           0
    0
biz-internet    172.16.255.40   private2        20006     1         65535
   55905       1119300231(0)                0           0           0
    0
mpls            172.16.255.40   biz-internet    20019     1         65535
   55905       1119300231(0)                0           0           0
    0
mpls            172.16.255.40   private1        20020     1         65535
   55905       1119300231(0)                0           0           0
    0
mpls            172.16.255.40   bronze          20021     1         65535
   55905       1119300231(0)                0           0           0
    0
biz-internet    172.16.255.41   bronze          20005     4         65535
   55905       1119139948(0)                0           0           0
```

```
    0
biz-internet    172.16.255.41   biz-internet    20007     4          65535
   55905       1119139948(0)                 0           0           0
    0
biz-internet    172.16.255.41   private2        20008     4          65535
   55905       1119139948(0)                 0           0           0
    0
biz-internet    172.16.255.41   mpls            20009     4          65535
   55905       1119139948(0)                 0           0           0
    0
mpls            172.16.255.41   bronze          20023     4          65535
   55905       1119139948(0)                 0           0           0
    0
mpls            172.16.255.41   biz-internet    20025     4          65535
   55905       1119139948(0)                 0           0           0
    0
mpls            172.16.255.41   private2        20026     4          65535
   55905       1119139948(0)                 0           0           0
    0
mpls            172.16.255.41   mpls            20027     4          65535
   55905       1119139948(0)                 0           0           0
    0
```

To verify the cumulative tunnel statistics, use the **show sdwan tunnel statistics table** command.

```
sender# show sdwan tunnel statistics ppl table
TUNNEL                                SOURCE  DEST
 PROTOCOL  SOURCE IP   DEST IP     PORT    PORT   ppl-tx-ipv4-pkts
ppl-tx-ipv6-pkts  ppl-tx-ipv4-octets  ppl-tx-ipv6-octets  ppl-rx-ipv4-pkts
ppl-rx-ipv6-pkts  ppl-rx-ipv4-octets  ppl-rx-ipv6-octets
ipsec     10.1.61.40  10.1.40.40  12346   12346  1024              0
        1224704             0                   0                0
       0                   0
ipsec     10.1.61.40  10.1.42.40  12346   12346  1024              0
        1224704             0                   0                0
       0                   0
ipsec     10.1.61.40  10.1.43.40  12346   12346  1024              0
        1224704              0                   0                0
        0                   0
ipsec     20.1.60.40  10.1.40.40  12346   12346  1024             0
      1224704              0                   0                0
      0                   0
ipsec     20.1.60.40  10.1.41.40  12346   12346  1024              0
        1224704              0                   0                0
        0                   0
ipsec     20.1.60.40  10.1.42.40  12346   12346  1024              0
        1224704             0                   0                0
       0                   0
ipsec     20.1.60.40  10.1.43.40  12346   12346  1024              0
        1224704             0                   0                0
       0                   0
ipsec     20.1.60.40  10.1.44.40  12346   12346  1024              0
        1224704             0                   0                0
       0                   0
```

To verify the configured packet reordering parameters on a receiver, use the **show platform software pkt-reorder cfg** command.

```
receiver# show platform software pkt-reorder cfg
Enable                : true
Max-allowed-memory    : 40%
```

```
Packets-per-context    : 8192
Out-of-order-tolerance : 2048
Dropped-packet-timeout : 100 ms
Idle-timeout           : 1000 ms
Dispact-size           : 2048
```

To check if the packet re-order context ID is correctly assigned for remote senders on a receiver device, use the **show sdwan ftm site-db** command.

```
receiver# show sdwan ftm site-db 300
OD Tunnel Idle Time   : 10 min
 Site ID: 300, OD: Yes, status: Inactive, sys_cnt: 1, status_change:
cnt ltime 08:59:05:953, last_stats 0, is_pending_tloc
_replay=0, replay_cnt=2 Idle Timer: Not running, time_to_expiry: 0 sec
 Remote   system: 172.16.255.80, OD: Yes, status: Inactive,
tloc_cnt: 4, tun_cnt: 20, glean_ipc_cnt: 0, is_pending_tloc_r
eplay=0, replay_cnt=1010, pkt_ro_ctx_id: 7, ppl_ctx_id: 5 ppl_cfg:
True ppl_adaptive_flowlet_gap: 1000
      TLOC  :: 172.16.255.80 : privatel : ipsec, nh_index: 32800,
OD: Yes, status: Inactive
      TLOC  :: 172.16.255.80 : bronze : ipsec, nh_index: 32801, OD:
Yes, status: Inactive
      TLOC  :: 172.16.255.80 : mpls : ipsec, nh_index: 32803, OD:
Yes, status: Inactive
      TLOC  :: 172.16.255.80 : biz-internet : ipsec, nh_index:
32804, OD: Yes, status: Inactive
      Tunnel : 10.1.41.40/12346->10.1.62.40/12346 proto 0x32, idx
960, sys 172.16.255.80, nh_index 66495
```

To view the reordering statistics of a PPL network, use the **show platform software pkt-reorder ctx sdwan** command.

```
receiver# show platform software pkt-reorder ctx sdwan
Context ID Module IP Address Total Packet Time Exceeded Too Old In Window Wrap
 DM Distance Mem Usage Total Mem Usage
1 sdwan 172.16.255.80 0 0 0 0 0 0 0 0
2 sdwan 172.16.255.81 132,213,316 403 130,239,435 829,845 1
43,955,369,702,422,670 0 0
3 sdwan 172.16.255.83 0 0 0 0 0 0 0 0
4 sdwan 172.16.255.82 0 0 0 0 0 0 0 0
```

15 PPPoE

Topics:

- Feature history for PPPoE
- PPPoE
- PPPoE over ATM
- Supported platforms for PPPoE over ATM
- Configure PPPoE using templates
- Configure PPPoE over ATM using templates
- Configuration examples for PPPoE

Introduces PPPoE functionality, describing its purpose, core features, and relevance in network environments.

Feature history for PPPoE

Details the feature history of PPPoE, highlighting major enhancements and changes across software releases.

Table 44: Feature History

Feature Name	Release Information	Description
Configure PPPoE over ATM	Cisco IOS XE Catalyst SD-WAN Release 17.4.1a Cisco vManage Release 20.4.1	This feature provides support for configuring PPPoEoA on Cisco IOS XE Catalyst SD-WAN devices. PPPoEoA uses AAL5MUX encapsulation which delivers better efficiency compared to other encapsulation methods.

PPPoE

Explains the underlying concepts of PPPoE, including operational principles and benefits within network infrastructures.

A Point-to-Point Protocol over Ethernet (PPPoE) is a network protocol that

- connects multiple users over an Ethernet local area network to a remote site through common customer premises equipment,
- is commonly used in broadband aggregation such as by digital subscriber line (DSL), and
- provides authentication with the CHAP or PAP protocol.

In the Cisco Catalyst SD-WAN overlay network, CCisco IOS XE Catalyst SD-WAN devices can run the PPPoE client. The PPPoE server component is not supported. It is recommended that you configure quality of service (QoS) and shaping rate on a PPPoE Dialer interface. Queuing based QoS policies on both Dialer interface and PPPoE-enabled physical interface at the same time, is not supported.

PPPoE-enabled physical interfaces are supported on ATM PVCs and Ethernet interfaces. A dialer interface must be used for cloning virtual access. Multiple PPPoE client sessions can be configured on an Ethernet interface, but each session must use a separate dialer interface and a separate dialer pool.

The Cisco Catalyst SD-WAN implementation of PPPoE does not support the Compression Control Protocol (CCP) options, as defined in RFC 1962.

PPPoE over ATM

Describes PPPoE deployment over ATM, outlining integration methods and the benefits of using ATM as the underlying transport.

A PPPoEoA is a protocol encapsulation method that

- uses ATM Adaptation Layer 5 Multiplexed Encapsulation (AAL5MUX) to carry PPPoE over ATM permanent virtual circuits (PVCs),
- provides efficiency gain over AAL5 LLC/SNAP encapsulation, and
- reduces Subnetwork Access Protocol (SNAP) encapsulation bandwidth usage by using multiplexed (MUX) encapsulation to reduce the number of cells needed to carry voice packets.

You can configure PPPoE over ATM interfaces (PPPoEoA) on Cisco IOS XE Catalyst SD-WAN devices that support ADSL. Deploying the PPPoEoA over ATM AAL5MUX feature in a VoIP environment results in improved throughput and bandwidth usage.

Supported platforms for PPPoE over ATM

Lists and explains the platforms that support PPPoE over ATM, helping users identify compatible hardware and software environments.

The following platforms support PPPoE over ATM:

- Cisco 1100 4G/6G Series Integrated Services routers.
- Cisco1100 Series Integrated Service routers.
- Cisco1109 Series Integrated Service routers.
- Cisco111x Series Integrated Service routers.
- Cisco1111x Series Integrated Service routers.
- Cisco1120 Series Integrated Service routers.
- Cisco1160 Series Integrated Service routers.

Configure PPPoE using templates

Provides instructions for configuring PPPoE sessions using templates, covering step-by-step procedures for streamlined deployment.

To use Cisco SD-WAN Manager templates to configure PPPoE on Cisco IOS XE Catalyst SD-WAN device, you create three feature templates and one device template:

- Create a VPN-Interface-PPP feature template to configure PPP parameters for the PPP virtual interface.
- Create a VPN-Interface-PPP-Ethernet feature template to configure a PPPoE-enabled interface.
- Optionally, create a VPN feature template to modify the default configuration of VPN 0.
- Create a device template that incorporates the VPN-Interface-PPP, VPN-Interface-PPP-Ethernet, and VPN feature templates.

Follow these steps to configure PPPoE using feature templates.

Procedure

1. Create a VPN-Interface-PPP feature template to configure PPP parameters for the PPP virtual interface.
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Feature Templates**, and click **Add Template**.

      In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

   c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
   d) Choose the **VPN-Interface-PPP** template.
   e) In the template, configure the following parameters:

Table 45:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
Shutdown	Click **No** to enable the PPP virtual interface.
Interface Name	Enter the number of the PPP interface. It can be from 1 through 31.
Description (optional)	Enter a description for the PPP virtual interface.
Authentication Protocol	Select either CHAP or PAP to configure one authentication protocol, or select PAP and CHAP to configure both. For CHAP, enter the hostname and password provided by your ISP. For PAP, enter the username and password provided by your ISP. If you are configuring both PAP and CHAP, to use the same username and password for both, click Same Credentials for PAP and CHAP.
AC Name (optional)	Select the PPP tab, and in the AC Name field, enter the name of the the name of the access concentrator used by PPPoE to route connections to the Internet.
IP MTU	Click **Advanced**, and in the IP MTU field, ensure that the IP MTU is at least 8 bytes less than the MTU on the physical interface. The maximum MTU for a PPP interface is 1492 bytes. If the PPPoE server does not specify a maximum receive unit (MRU), the MTU value for the PPP interface is used as the MRU. Starting from Cisco vManage Release 20.9.1, there is 8 bytes overheads deduced based on the specified IP MTU value when configuration is pushed to the device.
Save	To save the feature template, click **Save**.

2. Create a VPN-Interface-PPP-Ethernet feature template to enable the PPPoE client on the physical interfaces.
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Feature Templates**, and click **Add Template**.

      In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

   c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
   d) Choose the **VPN-Interface-PPP-Ethernet** template.
   e) In the template, configure the following parameters:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.

Parameter Field	Procedure
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
Shutdown	Click **No** to enable the PPPoE-enabled interface.
Interface Name	Enter the name of the physical interface in VPN 0 to associate with the PPP interface.
Description (optional)	Enter a description for the PPPoE-enabled interface.
IP Confguration	Assign an IP address to the physical interface: • To use DHCP, select Dynamic. The default administrative distance of routes learned from DHCP is 1. • To configure the IP address directly, enter of the IPv4 address of the interface.
DHCP Helper (optional)	Enter up to four IP addresses for DHCP servers in the network.
Save	To save the feature template, click **Save**.

3. Create a VPN feature template to configure the PPPoE-enabled interface in VPN 0, the transport VPN.
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Feature Templates**, and click **Add Template**.

      In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

   c) Choose Cisco IOS XE Catalyst SD-WAN device Cloud or a router model.
   d) Choose the **VPN** template.
   e) In the template, configure the following parameters:

Parameter Field	Procedure
Template Name	Enter a name for the template. It can be up to 128 alphanumeric characters.
Description	Enter a description for the template. It can be up to 2048 alphanumeric characters.
VPN Identifier	Enter VPN identifier 0.
Name	Enter aname for the VPN.
Other interface parameters	Configure the desired interface properties.
Save	To save the feature template, click **Save**.

4. Create a device template that incorporates the VPN-Interface-PPP, VPN-Interface-PPP-Ethernet, and VPN feature templates.
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Device Templates**, and then click **Create Template**.

In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

c) From the **Create Template** drop-down list, choose From Feature Template.

From the **Device Model** drop-down list, choose the type of device for which you are creating the device template.

Cisco SD-WAN Manager displays the feature templates for the device type you selected. Required templates are indicated with an asterisk (*).

Enter a name and description for the device template. These fields are mandatory. The template name cannot contain special characters.

d) In **Transport & Management VPN**, under **VPN 0**, from the drop-down list of available templates, select the desired feature template. The list of available templates are the ones that you have previously created.
e) In **Additional VPN 0 Templates**, click the plus sign (+) next to **VPN Interface PPP**.
f) From **VPN-Interface-PPP** and **VPN-Interface-PPP-Ethernet** fields, select the feature templates to use.
g) To configure multiple PPPoE-enabled interfaces in VPN 0, click the plus sign (+) next to Sub-Templates.
h) To include additional feature templates in the device template, in the remaining sections, select the feature templates in turn, and from the drop-down list of available templates, select the desired template. The list of available templates are the ones that you have previously created. Ensure that you select templates for all mandatory feature templates and for any desired optional feature templates.
i) To create the device template, click **Create**.

5. Attach a device template to a device.
   a) From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
   b) Click **Device Templates**, and then click **Create Template**.

   In Cisco vManage Release 20.7.1 and earlier releases, **Device Templates** is titled **Device**.

   c) Choose a template.
   d) Click ..., and click **Attach Device**.
   e) Search for a device or select a device from the Available Device(s) column to the left.
   f) Click the arrow pointing right to move the device to the Selected Device(s) column on the right.
   g) Click **Attach**.

Configure PPPoE over ATM using templates

Guides users through configuring PPPoE over ATM using templates, detailing task steps to ensure correct and efficient setup.

Follow these steps to configure PPPoE using the device CLI template in Cisco SD-WAN Manager.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. From **Device Templates**, click **Create Template**.

   In Cisco vManage Release 20.7.x and earlier releases, **Device Templates** is titled **Device**.

3. From the **Create Template** drop-down list, select **CLI Template**.
4. From the **Device Model** drop-down list, select the type of device for which you are creating the template.
   a) In **Template Name**, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.
   b) In **Template Description**, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

5. Choose **Device configuration**.

   Using this option, you can provide IOS-XE configuration commands that appear in the output of the **show sdwan running-config** command.

6. (Optional) To load the running config of a connected device, select it from the Load Running config from reachable device list and click **Search**.
7. In CLI Configuration, enter the configuration either by typing it, cutting and pasting it, or uploading a file.

```
Device(config)# interface atm number
Device(config)# no ip address
Device(config)# interface atm number point-to-point
Device(config)# no atm enable-ilmi-trap
Device(config)# encapsulation aal5mux  pppoe-client
Device(config)# pppoe-client dial-pool-number number
Device(config)# interface Dialer dialer-rotary-group-number
Device(config)# mtu bytes
Device(config)# ip address negotiated
Device(config-if)# encapsulation encapsulation-type
Device(config)# load-interval seconds
Device(config)# dialer pool number
Device(config)# dialer-group group-number
Device(config)# ppp mtu adaptive
Device(config)# ppp chap hostname hostname
Device(config)# ppp chap password secret
Device(config)# ppp ipcp address required
Device(config)# ppp link reorders
```

8. To convert an actual configuration value to a variable, select the value and click Create Variable. Enter the variable name, and click Create Variable. You can also type the variable name directly, in the format {{*variable-name*}}; for example, {{hostname}}.
9. Click **Add**. The new device template is displayed in the Device Template table. The **Type** column shows **CLI** to indicate that the device template was created from CLI text.

Configuration examples for PPPoE

Demonstrates a sample configuration for a PPPoE server, offering practical examples to assist in implementing and verifying server setup.

PPPoE server on IPv4 interfaces

This example shows configuring PPPoE server on IPv4 interfaces:

```
!
interface Dialer100
 mtu 1492
 ip address negotiated
 encapsulation ppp
 ip tcp adjust-mss 1460
 dialer pool 100
 dialer down-with-vInterface
 ppp authentication chap callin
 ppp chap hostname cisco
 ppp chap password 7 1511021F07257A767B
 ppp ipcp route default
```

To replace a template configured with PPPoE as WAN interface with a regular interface in Dialer100, remove the IP address assigned to the dialer interface using the **no ip address** command, and then add a new IP address for the dialer interface.

PPPoE over ATM interfaces

This example shows configuring PPPoE over ATM interfaces.

```
Device(config)# interface ATM0/1/0
Device(config)# no ip address
Device(config)# no atm enable-ilmi-trap
!
Device(config)# interface ATM0/1/0.10 point-to-point
Device(config)# no atm enable-ilmi-trap
Device(config)# cdp enable
Device(config)# pvc 22/62
Device(config)# ubr 1045
Device(config-if)# encapsulation aal5mux pppoe-client
Device(config)# pppoe-client dial-pool-number 120
!
!
Device(config)# interface Dialer 120
Device(config)# mtu 1492
Device(config)# ip address negotiated
Device(config)# ip nat outside
Device(config-if)# encapsulation ppp
Device(config)# load-interval 30
Device(config)# dialer pool 120
Device(config)# dialer-group 1
Device(config)# ppp mtu adaptive
Device(config)# ppp chap hostname test@cisco.com
Device(config)# ppp chap password 0 cisco
Device(config)# ppp ipcp address required
Device(config)# ppp link reorders
!
```

16 TCP MSS and Clear Dont Fragment

Topics:

- Feature history for TCP MSS and clear dont fragment
- TCP MSS
- Restrictions for TCP MSS and clear dont fragment
- Configure TCP MSS and clear dont fragment
- Verify TCP MSS and dont clear fragment configurations

Introduces TCP MSS and clear don't fragment concepts, explaining their purpose and role in controlling packet size and fragmentation within network infrastructure.

Feature history for TCP MSS and clear dont fragment

Details the evolution and key enhancements of the TCP MSS and clear don't fragment features, highlighting major updates and functional milestones released across versions.

Table 46: Feature History

Feature Name	Release Information	Description
Configure TCP MSS	Cisco IOS XE Catalyst SD-WAN Release 17.5.1a Cisco vManage Release 20.5.1	This feature adds support for TCP MSS adjustment on Cisco IOS XE Catalyst SD-WAN devices on both directions of the Cisco Catalyst SD-WAN tunnel interface.
Configure Clear Don’t Fragment Option	Cisco IOS XE Catalyst SD-WAN Release 17.5.1a Cisco vManage Release 20.5.1	This feature provides the option to clear the Don't Fragment bit in the IPv4 packet header for packets being sent out on a Cisco Catalyst SD-WAN tunnel . When you clear the Don't Fragment configuration, packets larger than the interface MTU are fragmented before being sent.

TCP MSS

Explains the principles of TCP MSS, describing its impact on TCP packet sizing and outlining considerations for optimizing data transmission efficiency.

A TCP maximum segment size (MSS) is a parameter that

- specifies the largest amount of data, in bytes, that a communications device can receive in a single TCP segment without counting the TCP header or the IP header, and
- is specified as TCP MSS, initially in the TCP SYN packet during the TCP handshake.

TCP MSS for SYN packets

Small MSS values reduce or eliminate IP fragmentation, which results in higher overhead. You can configure the MSS of TCP SYN packets passing through a device. By default, the device dynamically adjusts the MSS based on the interface or tunnel maximum transmission unit (MTU) to ensure that TCP SYN packets are never fragmented. For data sent over an interface, the device calculates the MSS by adding the interface MTU, the IP header length, and the maximum TCP header length.

Restrictions for TCP MSS and clear dont fragment

Describes limitations and restrictions associated with TCP MSS and clear don't fragment, outlining operational boundaries and unsupported scenarios for these features.

TCP MSS

- TCP MSS values can be adjusted for Cisco Catalyst SD-WAN tunnel interfaces only.

From Cisco IOS XE Catalyst SD-WAN Release 17.9.1aand Cisco vManage Release 20.9.1, you can adjust the TCP MSS value for a service VPN or for Network Address Translation (NAT) Direct Internet Access (DIA) use cases. Adjusting the TCP MSS value helps prevent TCP sessions from being dropped.

For more information on NAT DIA, see the *Cisco Catalyst SD-WAN NAT Configuration Guide*.

- From Cisco IOS XE Catalyst SD-WAN Release 17.18.1a, for an IPv4 SD-WAN tunnel carrying IPv6 traffic, the system uses the IPv6 TCP MSS value. Also, for an IPv6 tunnel carrying IPv4 traffic, the system uses the IPv4 TCP MSS value. This upgrade aligns MSS adjustment with the traffic protocol and tunnel configurations, ensuring accurate and efficient handling of traffic.

Clear dont fragment

- The option **Clear Dont Fragment** is available for Cisco Catalyst SD-WAN tunnel interfaces only.

Configure TCP MSS and clear dont fragment

Provides comprehensive instructions for configuring TCP MSS and clear don't fragment using both CLI commands and template-based methods, enabling flexible deployment and management.

Use one of these methods to configure TCP MSS and clear dont fragment:

- CLI commands
- Templates

Configure TCP MSS and clear dont fragment using CLI commands

Follow these steps to configure TCP MSS and clear dont fragment using CLI commands:

Procedure

1. Configure an interface type and enter the interface configuration mode.

   Example

   ```
   interface Tunnel 1
   ```

2. Enable IP processing on an interface without assigning an explicit IP address to the interface.

   Example

   ```
   ip unnumbered GigabitEthernet1
   ```

3. Configure TCP MSS and clear dont fragment.
   - Enable a maximum segment size (MSS) for TCP connections.

   Example

   ```
   ip tcp adjust-mss 1460
   ```

   - Clear the Don't Fragment (DF) bit in the IPv4 packet header for packets being transmitted out the interface.

Example

```
ip clear-dont-fragment
```

The following is an example configuration of TCP MSS:

```
Device(config)#interface Tunnel 1
Device(config-if)#ip unnumbered GigabitEthernet1
Device(config-if)#ip tcp adjust-mss 1460
```

The following is an example to configure Clear Dont Fragment option:

```
Device(config)#interface Tunnel 1
Device(config-if)#ip unnumbered GigabitEthernet1
Device(config-if)#ip clear-dont-fragment
```

Configure TCP MSS and clear dont fragment using templates

Follow these steps to configure TCP MSS and clear dont fragment using feature templates.

Procedure

---

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.

   > **Note**
   >
   > In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is titled **Feature**.

3. Create a new CLI add-on feature template or edit one of the following templates. You can use any of the following feature templates to configure TCP MSS and clear dont fragment:
   - VPN Ethernet Interface
   - VPN Interface DSL IPoE
   - VPN Interface DSL PPoA
   - VPN Interface DSL PPPoE
   - VPN Interface Multilink
   - VPN Interface T1/E1
   - Cellular Interfaces
4. Click **Tunnel**.
5. Specify the MSS of TPC SYN packets passing through the Cisco IOS XE Catalyst SD-WAN device to configure TCP MSS in the tunnel TCP MSS. By default, the MSS is dynamically adjusted based on the interface or tunnel MTU so that TCP SYN packets are not fragmented.

- Range: 552 to 1460 bytes
- Default: None

TCP MSS affects any packet that contains an initial TCP header that flows through the router. When configured, TCP MSS is examined against the MSS exchanged in the three-way handshake. The MSS in the header is lowered if the configured setting is lower than the MSS in the header. If the MSS header value is already lower than the TCP MSS, it flows through unmodified. The host at the end of the tunnel uses the lower setting of the two hosts. If the TCP MSS is to be configured, it should be set at 40 bytes lower than the minimum path MTU.

6. Click the **Clear-Dont-Fragment** option to clear the Dont Fragment bit in the IPv4 packet header for packets being transmitted out the interface. When the **Don't Fragment** bit is cleared, packets larger than that interface's MTU are fragmented before being sent.

   Clear-Dont-Fragment clears the **Don't Fragment** bit when there is fragmentation needed and the **Don't Fragment** bit is set. For packets that don't require fragmentation, the **Don't Fragment** bit is not affected.

7. Click **Save** or **Update**.

Verify TCP MSS and dont clear fragment configurations

Outlines verification steps to confirm TCP MSS and don't clear fragment configurations, detailing command outputs and expected results to validate correct implementation.

TCP MSS configuration

The following is sample output of the **show platform hardware qfp active feature sdwan datapath session summary** command:

```
Device# show platform hardware qfp active feature sdwan datapath session summary

Src IP          Dst IP          Src Port Dst Port   Encap    Uidb     Bfd
Discrim PMTU

------          ------          -------- -------    ------   -------
----------- -----

10.1.15.25      10.1.14.14      12347    12346      IPSEC    65526    10007
     1446

10.1.15.25      10.0.5.21       12347    12357      IPSEC    65526    10009
     1446

10.1.15.25      10.0.5.11       12347    12347      IPSEC    65526    10008
     1446

10.1.15.25      10.1.16.16      12347    12366      IPSEC    65526    10006
     1446
```

Dont fragment configuration

The following is sample output of the **show platform software interface rp active name Tunnel1** command to verify if **Clear-dont-fragment** is enabled or not.

```
Device# show platform software interface rp active name Tunnel1 | include dont
IP Clear-dont-fragment: TRUE
```

The following is sample output of the **show running-config interface Tunnel1** command that displays the running configuration when Clear-dont-fragment is enabled.

```
Device# show running-config interface Tunnel1
Building configuration...

Current configuration : 132 bytes
!
interface Tunnel1
ip unnumbered GigabitEthernet1
ip clear-dont-fragment
tunnel source GigabitEthernet1
tunnel mode sdwan
end
```

17 Track Static Routes for Service VPNs

Topics:

- Feature history of track static routes for service VPNs
- Track static routes for service VPNs
- Supported platforms
- Restrictions for IPv4 static route tracking
- Configure tracker group using a configuration group
- Create a static route tracker
- Configure a next hop static route with tracker
- Monitor static route tracker configuration
- Configure static routes using CLI
- Configuration examples static route tracking
- Verify static route tracking configuration

Introduces static route tracking within service VPNs, providing an overview of the feature's purpose, operational principles, and role in ensuring dynamic failover and resilient network connectivity.

Feature history of track static routes for service VPNs

Details the evolution of static route tracking for service VPNs, describing feature milestones, software releases, and key enhancements that impact configuration and deployment options over time.

This table describes the developments of this feature, by release.

Table 47: Feature history

Feature Name	Release Information	Description
Static Route Tracker for Service VPNs	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	This feature enables you to configure IPv4 static route endpoint tracking for service VPNs. For static routes, endpoint tracking determines whether the configured endpoint is reachable before adding that route to the route table of the device.
TCP/UDP Endpoint Tracker and Dual Endpoint Static Route Tracker for Cisco IOS XE Catalyst SD-WAN devices	Cisco IOS XE Catalyst SD-WAN Release 17.9.1a Cisco vManage Release 20.9.1	This feature enables you to configure the TCP/UDP static route endpoint trackers. Using this feature you can also configure IPv4, TCP/UDP dual endpoint static-route tracker groups for service VPNs to enhance the reliability of probes.

Track static routes for service VPNs

Explains the concept of static route tracking for service VPNs, focusing on operational mechanisms, use cases, and its importance for routing reliability in network environments.

Tracking static routes for service VPNs allows you to monitor the reachability of the configured next-hop endpoint IP address. You can verify the endpoint before the device adds the static route to its routing table. This is particularly important in service VPNs where static routes are advertised over the Overlay Management Protocol (OMP). The static route tracker periodically sends Internet Control Message Protocol (ICMP) ping probes to the configured endpoint IP address. If the endpoint is unreachable (no response to probes), the static route is excluded from the routing table and is not advertised to OMP, preventing traffic blackholing. You can configure backup routes with higher administrative distance to provide alternate paths.. This mechanism enhances route reliability and network stability in SD-WAN environments. The tracker sends periodic probes (ICMP echo requests, TCP, or UDP probes) to the static route's next-hop IP address.

- If the endpoint is unreachable, the static route is removed from the routing table and not advertised to OMP.
- Configure backup static routes with higher administrative distance for failover.
- Only one endpoint tracker is supported per static route per next-hop.
- IPv6 static routes are not supported for tracking.
- You can configure dual endpoint tracker groups to enhance probe reliability.
- This feature is supported on Cisco IOS XE Catalyst SD-WAN platforms such as ASR 1000, ISR 1000 and 4000, and CSR 1000 series routers.
- This feature ensures that static routes in service VPNs are advertised only when their next-hop endpoints are reachable. As a result, it prevents traffic blackholing and improves network resilience.

From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure TCP or UDP individual endpoint trackers and configure a tracker group with dual endpoints (using two trackers), and associate the trackers and tracker group to a static route. Dual endpoints help avoid false negatives that route unavailability might introduce.

Starting with Cisco IOS XE Catalyst SD-WAN Release 17.15.1a, Cisco SD-WAN Manager reports only UP/DOWN status changes. It does not report RTT value changes. This optimization improves efficiency in extensive networks.

Supported platforms

Lists supported hardware and software platforms for static route tracking within service VPNs, ensuring users can determine compatibility and implementation feasibility across network infrastructures.

These devices support static route tracking for service VPNs:

- Cisco ASR 1000 Series Aggregated Services Routers
- Cisco ISR 1000 Series-Integrated Services Routers
- Cisco ISR 4000 Series Integrated Services Routers
- Cisco CSR 1000 Series Cloud Service Routers

Restrictions for IPv4 static route tracking

Outlines limitations, prerequisites, and operational restrictions associated with IPv4 static route tracking within service VPNs, enabling users to design compliant and effective configurations.

- Only one endpoint tracker is supported per static route per next-hop address.
- IPv6 static routes are not supported.
- To configure a static route with tracker:
  1. Delete any existing static route, if it is already configured without a tracker. Plan for any connectivity downtime that might occur during this step for static route advertisement.
  2. Configure a new static route with tracker using the same prefix and next-hop as the deleted static route.
- To add a new tracker after you reach maximum tracker limit per router:
  1. Delete an old tracker and attach the template to the device.
  2. Add a new tracker and attach the device to the template again.
- UDP tracker endpoint enabled with IP SLA UDP packet responder is supported only on Cisco IOS XE Catalyst SD-WAN devices.
- You cannot link the same endpoint-tracker to static routes in different VPNs. Endpoint-tracker is identified by a name and can be used for multiple static routes in a single VPN.

Configure tracker group using a configuration group

Guides through the process of configuring tracker groups using configuration groups, offering step-by-step instructions to establish route tracking associations for improved failover and network resilience.

Before you begin

On the **Configuration** > **Configuration Groups** page, choose **SD-WAN** as the solution type.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Create and configure a tracker group in a Service Profile.

Table 48: Tracker Group

Field	Description
Tracker Elements*	This field is displayed only if you chose **Tracker Type** as the **Tracker Group**. Add the existing interface tracker names, separated with a space. When you add this tracker to the template, the tracker group is associated with these individual trackers, and you can then associate the tracker group to an interface.
Tracker Boolean	This field is displayed only if you chose **Tracker Type** as the **Tracker Group**. Select **AND** or **OR**. **OR** is the default boolean operation. An **OR** ensures that the transport interface status is reported as active if either one of the associated trackers of the tracker group reports that the interface is active. If you select the **AND** operation, the transport-interface status is reported as active if both the associated trackers of the tracker group report that the interface is active.

What to do next

Also see Deploy a configuration group.

Create a static route tracker

Provides instructions to create and configure a static route tracker, ensuring continuous monitoring and automated response to network events that affect static route availability.

Before you begin

Use the **System Template** to create a tracker for static routes.

Delete existing static routes, if any, before you create a static route tracker. Configure a new static route tracker using the same prefix and next hop as the deleted static route.

Procedure

1. From Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Navigate to the **Cisco System** template for the device.

   For information about creating a System template, see Create System Template.
4. Click **Tracker**. Click **New Endpoint Tracker** to configure the tracker parameters.

Table 49: Tracker Parameters

Field	Description
Name	Name of the tracker. The name can be up to 128 alphanumeric characters.
Threshold	Wait time for the probe to return a response before declaring that the configured endpoint is down. The range is from 100 to 1000 milliseconds. The default is 300 milliseconds.
Interval	Time interval between probes to determine the status of the configured endpoint. The default is 60 seconds (1 minute). The range is from 20 to 600 seconds.
Multiplier	Number of times probes are sent before declaring that the endpoint is down. The range is from 1 to 10. The default is 3.
Tracker Type	From the drop-down, choose Global. From the Tracker Type field drop-down, choose Static Route. From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure a tracker group with dual endpoints on Cisco IOS XE Catalyst SD-WAN devices. You can associate this tracker group to a static route.
Endpoint Type	Choose endpoint type IP address. **Note** Configuring the tracker type Static Route using endpoint URL or endpoint DNS name is not supported.
End-Point Type: IP Address	IP address of the static route end point. This is the destination on the internet to which the router sends probes to determine the status of the route.

5. Click **Add**.
6. Click **Save**.

   Complete all mandatory actions before saving the template.

7. To create a tracker group, click **Tracker Groups** > **New Endpoint Tracker Groups** and configure the tracker parameters.

   Ensure that you have created two trackers to form a tracker group.

Table 50: Tracker Group Parameters

Fields	Description
Name	Name of the tracker group.

Fields	Description
Tracker Type	From the drop-down, choose **Global**. From the Tracker Type field drop-down, choose **Static Route**. From Cisco IOS XE Catalyst SD-WAN Release 17.7.1a, you can configure a tracker group with dual endpoints on Cisco IOS XE Catalyst SD-WAN devices and associate this tracker group to a static route.
Tracker Elements	This field is displayed only if you chose **Tracker-group** as the tracker type. Add the existing interface tracker names (separated by a space). When you add this tracker to the template, the tracker group is associated with these individual trackers, and you can then associate the tracker group to a static route.
Tracker Boolean	From the drop-down list, choose **Global**. This field is displayed only if you chose **tracker-group** as the **Tracker Type**. By default, the **OR** option is selected. Choose **AND** or **OR**. **OR** ensures that the static route status is reported as active if either one of the associated trackers of the tracker group report that the route is active. If you select **AND**, the static route status is reported as active if both the associated trackers of the tracker group report that the route is active.

8. Click **Add**.
9. Click **Save**.

   Complete all mandatory actions before saving the template.

Configure a next hop static route with tracker

Use the VPN template to associate a tracker to a static route next hop. You can apply only one tracker per static route next hop.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates** and navigate to **Cisco VPN Template** for the device.

   For information about creating a VPN template, see Create VPN Template.
3. Enter **Template Name** and **Description** as required.
4. In Basic Configuration, by default, VPN is set to 0. Set a VPN value within (1-511, 513-65530) range for service VPNs, for service-side data traffic on Cisco IOS XE Catalyst SD-WAN devices.

   You can configure static route tracker only on service VPNs.
5. Click **IPv4 Route** and **New IPv4 Route**.
6. In the **IPv4 Prefix** field, enter a value.
7. Click **Next Hop**. Click **Add Next Hop** with **Tracker** and enter values for the fields listed in the table.

Parameter Name	Description
Address	Specify the next-hop IPv4 address.

Parameter Name	Description
Distance	Specify the administrative distance for the route.
Tracker	Enter the name of the gateway tracker to determine whether the next hop is reachable before adding that route to the route table of the device.
Add Next Hop with Tracker.	Enter the name of the gateway tracker with the next hop address to determine whether the next hop is reachable before adding that route to the route table of the device.

8. Click **Add** and **Save** to create the static route with the next-hop tracker.

   You need to fill all the mandatory fields in the form to save the VPN template.

Monitor static route tracker configuration

Explains monitoring methods for static route tracker configurations, including steps to verify operational status and ensure effective route tracking within service VPNs.

To view information about a static tracker on a transport interface:

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Monitor** > **Devices**.
2. Choose a device from the list of devices.
3. Click **Real Time**.
4. From the **Device Options** drop-down list, choose **Endpoint Tracker Info**.

Configure static routes using CLI

Provides CLI-based instructions for configuring static routes within service VPNs, covering command usage and verification to support consistent and efficient route management.

These sections provide information about how to configure static routes using the CLI.

You can configure static route tracking using the Cisco SD-WAN Manager CLI Add-on feature templates and CLI device templates. For more information on configuring using CLI templates, see CLI Templates.

Configure a static route tracker

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
```

```
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure a static route tracker with TCP port as the endpoint

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> tcp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure a static route tracker with UDP port as the endpoint

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> udp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name> endpoint-tracker
```

Configure tracker groups

You can create tracker groups to probe static routes from Cisco IOS XE Catalyst SD-WAN Release 17.7.1a and Cisco vManage Release 20.7.1.

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name1>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> tcp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name1> endpoint-tracker
```

```
Device# config-transaction
Device(config)# endpoint-tracker <tracker-name2>
Device(config-endpoint-tracker)# tracker-type <tracker-type>
Device(config-endpoint-tracker)# endpoint-ip <ip-address> udp <port-number>
Device(config-endpoint-tracker)# threshold <value>
Device(config-endpoint-tracker)# multiplier <value>
Device(config-endpoint-tracker)# interval <value>
Device(config-endpoint-tracker)# exit
Device(config)# track <tracker-name2> endpoint-tracker
```

```
Device(config)# endpoint-tracker <static-tracker-group>
Device(config-endpoint-tracker)# tracker-type tracker-group
Device(config-endpoint-tracker)# tracker-elements <tracker-name1>
```

```
<tracker-name2>
Device(config-endpoint-tracker)# boolean {and | or}
Device(config-endpoint-tracker)# exit
Device(config)# track <static-tracker-group> endpoint-tracker

Device(config)# ip route vrf <vrf-name> <prefix> <mask> <nexthop-ipaddress>
<administrative-distance> track name <static-tracker-group>
```

Note

- Use the **ip route** command to bind a tracker or tracker group with a static route and to configure a backup route for administrative distance that is higher than the default value of 1.
- You can apply only one tracker to an endpoint.
- A tracker group can have a mix of endpoint trackers. For example, you can create a tracker group with an IP address tracker and UDP tracker.

Configuration examples static route tracking

Presents practical configuration examples for static route tracking, illustrating typical deployments, troubleshooting scenarios, and best practices for efficient setup.

Configure Tracker

This example shows how to configure a single static route tracker:

```
config-transaction
!
 endpoint-tracker tracker1
!
  tracker-type static-route
  endpoint-ip 10.1.1.1
  threshold 100
  multiplier 5
  interval 20
  exit
!
track tracker1 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name tracker1
```

This example shows how to configure a tracker with TCP port as endpoint:

```
config-transaction
!
 endpoint-tracker tcp-10001
!
  tracker-type static-route
  endpoint-ip 10.0.0.1 tcp 10001
  threshold    100
  interval     10
```

```
  multiplier   1
  exit
!
track tcp-10001 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name tcp-10001
```

This example shows how to configure a tracker with UDP port as endpoint:

```
config-transaction
!
  endpoint-tracker udp-10001
!
    tracker-type static-route
    endpoint-ip 10.0.0.1 udp 10001
    threshold    100
    interval     10
    multiplier   1
    exit
!
track udp-10001 endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name udp-10001
```

Configure Tracker Groups

This example shows how to configure a tracker group with two trackers (two endpoints). You can create tracker groups to probe static routes.

```
config-transaction
!
 endpoint-tracker tcp-10001
!
    tracker-type static-route
    endpoint-ip 10.1.1.1 tcp 10001
    threshold 100
    multiplier 5
    interval 20
    track tcp-10001 endpoint-tracker
!
 endpoint-tracker udp-10002
!
   tracker-type static-route
   endpoint-ip 10.2.2.2 udp 10002
   threshold 100
   multiplier 5
   interval 20
   track udp-10002 endpoint-tracker
!
endpoint-tracker static-tracker-group
!
  tracker-type tracker-group
  tracker-elements tcp-10001 udp-10002
  boolean and
  track static-tracker-group endpoint-tracker
!
ip route vrf 1 192.168.0.0 255.255.0.0 10.1.19.16 100 track name
static-tracker-group
```

Note

- You must configure an administrative distance when you are configuring through CLI templates.
- Use the **ip route** command to bind the tracker or tracker group with a static route and to configure a backup route for administrative distance when it is higher than the default value of 1.
- You can apply only one tracker to an endpoint.

Verify static route tracking configuration

Describes commands and procedures for verifying static route tracking configurations, ensuring successful deployment, operational accuracy, and reliable network failover.

Use these commands to verify if the configuration is committed. The sample configuration shows tracker definition for a static route tracker and it's application to an IPv4 static route:

```
Device# show running-config | sec endpoint-tracker
endpoint-tracker tracker1
endpoint-ip 10.1.1.1
interval 60
multiplier 5
tracker-type static-route
endpoint-tracker tracker2
endpoint-ip 10.1.1.12
interval 40
multiplier 2
tracker-type static-route
track tracker2 endpoint-tracker
track tracker1 endpoint-tracker
```

Use this command to verify the IPv4 route:

```
Device# show running-config | inc ip route
ip route vrf 1 10.1.1.11 255.255.0.0 10.20.2.17 track name tracker2
ip route vrf 1 10.1.1.12 255.255.0.0 10.20.24.17 track name tracker1
```

The sample output from the **show endpoint-tracker static-route** command displaying individual static route tracker status:

```
Device#  show endpoint-tracker static-route
Tracker Name   Status     RTT (in msec) Probe ID
tcp-10001         UP         3             1
udp-10002         UP         1             6
```

The sample output from the **show endpoint-tracker tracker-group** command displaying tracker group status:

```
Device# show endpoint-tracker group
Tracker Name               Element trackers name    Status         RTT in msec
  Probe ID
group-tcp-10001-udp-10002  tcp-10001, udp-10002     UP(UP AND UP)  5, 1
    9, 10
```

The sample output from the **show endpoint-tracker records** command displaying tracker/tracker group configuration:

```
Device# show endpoint-tracker records
Record Name                Endpoint                EndPoint Type Threshold(ms)
 Multiplier Interval(s) Tracker-Type
group-tcp-10001-udp-10002  tcp-10001 AND udp-10002  N/A           N/A
   N/A        N/A         static-tracker-group
tcp-10001                  10.1.1.1                TCP           100
   1          20          static-route
udp-10002                  10.2.2.2                UDP           100
   1          20          static-route
```

The sample output from the **show ip static route vrf** command:

```
Device# show ip static route vrf 1
Codes: M - Manual static, A - AAA download, N - IP NAT, D - DHCP,
       G - GPRS, V - Crypto VPN, C - CASA, P - Channel interface processor,
       B - BootP, S - Service selection gateway
       DN - Default Network, T - Tracking object
       L - TL1, E - OER, I - iEdge
       D1 - Dot1x Vlan Network, K - MWAM Route
       PP - PPP default route, MR - MRIPv6, SS - SSLVPN
       H - IPe Host, ID - IPe Domain Broadcast
       U - User GPRS, TE - MPLS Traffic-eng, LI - LIIN
       IR - ICMP Redirect, Vx - VXLAN static route
       LT - Cellular LTE, Ev - L2EVPN static route
Codes in []: A - active, N - non-active, B - BFD-tracked, D - Not Tracked, P
- permanent, -T Default Track
Codes in (): UP - up, DN - Down, AD-DN - Admin-Down, DL - Deleted
Static local RIB for 1
T  192.168.0.0 [1/0] via 10.1.19.16 [A]
```

18 VDSL and GSHDSL

Topics:

- VDSL and GSHDSL
- GSHDSL EFM-ATM NIM
- Cisco VDSL examples
- Cisco GSHDSL examples

Introduces VDSL and GSHDSL technologies, detailing their operational principles, capabilities, and relevance within Cisco networking solutions.

VDSL and GSHDSL

Explains technical concepts and use cases for VDSL and GSHDSL, highlighting their applications and benefits in network architectures.

Very High-Speed Digital Subscriber Line (VDSL) is a digital subscriber line (DSL) technology that provides high-speed data transmission over existing copper telephone wires. It is an advanced version of earlier DSL standards like ADSL, offering significantly faster speeds by utilizing higher frequency ranges

Symmetric High-Speed Digital Subscriber Line (G.SHDSL) is a data communications technology designed for symmetric high-speed data transmission over copper twisted pairs. While the query mentioned G.HDSL, G.SHDSL (standardized as ITU-T G.991.2) is the more modern and widely adopted international standard for symmetric DSL. G.HDSL (G.991.1) is an older standard.

Key characteristics of VDSL

- Speed: VDSL can offer speeds up to 52 Mbit/s downstream and 16 Mbit/s upstream. Its successor, VDSL2 (ITU-T G.993.2), can provide data rates exceeding 100 Mbit/s simultaneously in both upstream and downstream directions, and VDSL2+ can reach 300+ Mbit/s.
- Technology: It uses frequencies up to 12 MHz (VDSL) or up to 30 MHz (VDSL2) to achieve these higher speeds.
- Applications: VDSL is capable of supporting applications such as high-definition television (HDTV), voice over IP (VoIP), and general high-speed internet access over a single connection.
- Distance Sensitivity: The performance of VDSL degrades as the distance from the service provider's equipment increases.

For related information, see VDSL Commands.

Key characteristics of G.SHDSL

G.SHDSL is an international standard that allows devices to send and receive high-speed symmetrical data streams over a single pair of copper wires. This section provides information about the Cisco G.SHDSL EFM/ATM NIM and provides guidelines for configuring G.SHDSL in SD-WAN mode.

- Symmetry: Unlike asymmetric DSL (ADSL), G.SHDSL provides symmetrical transmit and receive data rates, meaning the upload and download speeds are equal.
- Speed: It offers symmetric rates typically ranging from 192 kbit/s to 15296 kbit/s on a single twisted pair.
- Applications: G.SHDSL is often used for high-speed commercial broadband services, enabling applications that require significant data transfer in both directions, such as LAN remote access, web hosting, and combining multiple voice and data channels.
- Compatibility: It is compatible with other DSL technologies and can extend transmission distances while maintaining high data rates.

For related information, see Configuring Cisco G.SHDSL HWICs in Cisco Access Routers and VDSL Commands.

VDSL configuration guidelines

This table provides usage information and guidelines for configuring asymmetric DSL (ADSL2/2+) and VDSL for supported Integrated Services Router Network Interface Modules (ISR NIMs) in SD-WAN mode. VDSL2 and ADSL2/2+ provide highly reliable WAN connections for remote sites.

Function	Command	Guidelines
Configure operating mode	Device# **configure terminal** Device(config)# **controller VDSL** *slot/subslot/port* Device(config)# **operating mode auto**	To switch from operating mode auto adsl1 (adsl2+/ or vdsl2) to operating mode auto ads2+ (adsl1 or vdsl2), switch to operating mode auto first. Before you change the operating mode, ensure that line-mode is changed to line-mode single-wire line 0.
Enable DSL on a line	Device(config)# **line-mode single-wire line** *line-number*	This command is supported only on DSL NIM-VAB-A.
Enable bonding	Device(config)# **line-mode bonding**	This command is supported only on DSL NIM-VAB-A.
Load firmware on a device	Device# **configure terminal** Device(config)# **controller VDSL** *slot* / *subslot* / *port* Device(config-controller)# **firmware phy filename** *filename*	The Cisco Catalyst SD-WAN CLI template does not support specifying the file location. Prepend the file name with flash: or with bootflash:, depending on its location.
Enable or disable SRA	Device(config-controller)# **sra**	The Cisco Catalyst SD-WAN CLI template does not support the sra *line number* command. In line-mode bonding, sra enables sra on both lines and no sra disables sra on both lines.
Enable or disable bitswap	Device(config-controller)# **bitswap**	The Cisco Catalyst SD-WAN CLI template does not support the bitswap *line number* command. In line-mode bonding, bitswap enables bitswap on both lines and no bitswap disables bitswap on both lines.
Enable modem features	Device(config-controller)# **modem** *keyword*	-
Display a description of a controller	Device(config-controller)# **description** *string*	-
Enable dual ended line testing	Device(config-controller)# **diagnostics DELT**	-
Modify the file in which the training log is stored	Device(config-controller)# **training log filename flash:** *filename*	The Cisco Catalyst SD-WAN CLI template does not support specifying the file location. Prepend the file name with flash: or with bootflash:, depending where the file should be stored.
Enable sync mode	Device(config-controller)# **sync mode** *mode*	To switch from one sync mode to another, delete the existing sync mode, then configure the new one.

Function	Command	Guidelines
Enable sync interval	Device(config-controller)# **sync interval** *seconds*	-

GSHDSL configuration guidelines

This table provides usage information and guidelines that apply when you configure the Cisco G.SHDSL EFM/ATM in CPE or CO mode.

Function	Command	Guidelines
Configure a device with the dsl-group auto command	Device(config-controller)# **dsl-group auto**	Use customer premises equipment (CPE) mode when configuring a device with the dsl-group auto command. If you use this command in Central Office (CO) mode, the configuration does not take effect.
Add or delete a link	–	The efm-grp command is not supported. To add or delete a link to a dsl-group, delete the dsl-group, then create a new dsl-group.
Load firmware on a device	Device(config-controller)# **firmware phy filename** *location*	File name location options are not supported when using the firmware phy command. Prepend the file name with flash: or with bootflash:, depending on the location.
Create or delete an annex	Device(config-controller-dsl-group)# **no shdsl annex** Device(config-controller-dsl-group)# **no shdsl rate** *rate*	To avoid Cisco IOS and Cisco Catalyst SD-WAN configuration from going out of sync when you create or delete an annex, create or delete the rate in the same transaction.
Enable SHDSL to use enhanced mode	(config-controller-dsl-group)# **shdsl 4-wire mode enhanced**	To enable SHDSL to use the enhanced mode in a 2-pair digital subscriber line (DSL) group, use the shdsl 4-wire mode enhanced command in configuration controller DSL group mode.
Ignore CRC errors	(config-controller-dsl-group)# **ignore** *seconds*	To configure a device to ignore CRC errors, use the ignore command. Replace *timeout* with a value from 0 through 60, which indicates the number of seconds that the device ignores CRC errors that do not resolve before the device terminates an action.
Shutdown a DSL group	(config-controller-dsl-group)# **shutdown**	To shut down a DSL group, use the shutdown command.

GSHDSL EFM-ATM NIM

Describes the Cisco GSHDSL EFM-ATM NIM hardware, outlining its features, operational modes, and integration within network infrastructures.

G.SHDSL EFM/ATM NIM

The Cisco G.SHDSL EFM/ATM NIM connects Cisco 4000 Series Integrated Services Routers with central office Digital Subscriber Line Access Multiplexers (DSLAMs) and supports up to four DSL pairs. The DSL pairs are bundled in groups and configured in the Cisco IOS CLI by using the dsl-group command. Use the mode command to choose the mode (ATM or EFM).

The NIM supports the following configuration:

- You can configure up to four DSL groups.
- You can configure auto mode on only one DSL group. For example, DSL group 0.
- In ATM Mode, you can configure the lines to use 2-wire, 4-wire (standard or enhanced), or m-pair.
- In EFM mode, you can configure a DSL group with any one of the lines in 2-wire non-bonding mode or with multiple lines in bonding mode.
- Depending on the mode (ATM or EFM), the corresponding interface (ATM or EFM) is automatically created.

Cisco VDSL examples

Provides practical Cisco VDSL configuration examples, illustrating common deployment scenarios and settings for reference.

Configuration example for VDSL:

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config)# operating mode auto
```

```
Device# config-transaction
Device(config)# line-mode single-wire line 1
```

```
Device# config-transaction
Device(config)# line-mode bonding
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# firmware phy filename
```

```
flash:IDC_1.7.2.6_DFE_FW_BETA_120111A.pkg
```

```
Device# config-transaction
Device(config-controller)# sra
```

```
Device# config-transaction
Device(config-controller)# bitswap
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# modem customUKAnnexM
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# description to ISP 1
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# diagnostics DELT
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# training log filename bootflash:VDSLLOG.log
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# sync mode ansi previous
```

```
Device# configure terminal
Device(config)# ptp clock ordinary domain 0
Device(config-ptp-clk)# clock-port slave slaveport
Device(config-ptp-port)# sync interval -4
Device(config-ptp-port)# end
```

The following example show the VDSL configuration:

```
Device(config)# show controllers vdSL 0/2/0
Controller VDSL 0/2/0 is UP

Daemon Status:          UP

                        XTU-R (DS)              XTU-C (US)
Chip Vendor ID:         'BDCM'                   'BDCM'
Chip Vendor Specific:   0x0000                   0xA39A
Chip Vendor Country:    0xB500                   0xB500
```

```
Modem Vendor ID:         'CSCO'                    'BDCM'
Modem Vendor Specific:   0x4602                    0x0000
Modem Vendor Country:    0xB500                    0xB500
Serial Number Near:      FGL2149956Y C1117-4P 16.7.20180
Serial Number Far:
Modem Version Near:      16.7.20180709:09395
Modem Version Far:       0xa39a

Modem Status:            TC Sync (Showtime!)
DSL Config Mode:         AUTO
Trained Mode:            G.993.2 (VDSL2) Profile 17a

TC Mode:                 PTM
Selftest Result:         0x00
DELT configuration:      disabled
DELT state:              not running

Failed full inits:       0
Short inits:             0
Failed short inits:      0

Modem FW  Version:       4.14L.04
Modem PHY Version:       A2pv6F039t.d26d

Line 0:

                         XTU-R (DS)                XTU-C (US)
Trellis:                  ON                        ON
SRA:                      enabled                   enabled
SRA count:                0                         0
Bit swap:                 enabled                   enabled
Bit swap count:           1                         3
Line Attenuation:         18.4 dB                    0.0 dB
Signal Attenuation:        0.0 dB                    0.0 dB
Noise Margin:              5.2 dB                   6.0 dB
Attainable Rate:         46022 kbits/s              18866 kbits/s
Actual Power:            14.5 dBm                  10.4 dBm
Per Band Status:         D1      D2      D3       U0      U1      U2      U3
Line Attenuation(dB):    13.9    32.7    50.1     N/A     25.6    37.7    42.3
Signal Attenuation(dB):  13.5    32.4    N/A      N/A     25.0    36.9    41.9
Noise Margin(dB):        5.3     5.1     N/A      N/A     6.0     6.0     5.9
Total FECC:              446                       0
Total ES:                3                         0
Total SES:               0                         0
Total LOSS:              0                         0
Total UAS:               50                        50
Total LPRS:              0                         0
Total LOFS:              0                         0
Total LOLS:              0                         0


                    DS Channel1     DS Channel0   US Channel1       US Channel0
Speed (kbps):              NA           47610            NA             18859
SRA Previous Speed:        NA               0            NA                 0
Previous Speed:            NA               0            NA                 0
Reed-Solomon EC:           NA             446            NA                 0
CRC Errors:                NA              51            NA                 0
Header Errors:             NA            3935            NA                 0
Interleave (ms):           NA            1.00            NA              1.00
Actual INP:                NA            0.00            NA              0.00
```

```
Training Log :  Stopped
Training Log Filename : flash:vdsllog.bin
```

Cisco GSHDSL examples

Presents Cisco GSHDSL configuration examples, showcasing typical setups and solution patterns to support deployment and troubleshooting.

Configuration example for GSHDSL:

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# dsl-group auto
```

```
Device# config-transaction
Device(config)# controller VDSL 0/0/0
Device(config-controller)# firmware phy filename
bootflash:IDC_1.1.1.0_DFE_1.1-1.8.1__001.pkg
```

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# dsl-group 0 pairs 0
Device(config-controller-dsl-group)# no shdsl annex
Device(config-controller-dsl-group)# no shdsl rate 5696
```

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
(config-controller-dsl-group)# shdsl 4-wire mode enhanced
```

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
config-controller-dsl-group)# ignore 30
```

```
Device# config-transaction
Device(config)# controller SHDSL 0/0/0
Device(config-controller)# termination cpe
Device(config-controller)# dsl-group 0 pairs 0
config-controller-dsl-group)# shutdown
```

The following example show the GSHDSL configuration:

```
Device# show controllers shDSL 0/1/0
Controller SHDSL 0/1/0 is UP
  Hardware is NIM-SHDSL-EA, on slot 0,bay 0
  Capabilities: EFM: 2-wire, EFM-Bond, Annex A, B, F & G
```

```
              ATM: 2-wire, Mpair, Annex A, B, F & G
CPE termination
cdb=0x7F7EB723D8A8
Vendor: Intel, Chipset: SOCRATES-4e
PHY Source: System
IDC Firmware version: 0.0.0.0
DFE Firmware version:
Group 0 info:
      Type: EFM Auto status: Down
       Ethernet Interface: Ethernet0/1/0, hwidb: 0x7F7EB723B648
       ATM Interface: ATM0/1/0, hwidb: 0x7F7EB724CE08
      Configured/active num links: 4/0, bit map: 0xF/0x0
      Line termination: CPE, Annex: auto
      PMMS disabled,Line coding: AUTO-TCPAM
      Configured/actual rate: AUTO/0 kbps
      Dying Gasp: Present
      SHDSL wire-pair (0) is in DSL DOWN state
              LOSWS Defect alarm: none
              SNR Margin alarm: none
              Loop Attenuation alarm: none
              Termination: CPE, Line mode: EFM Auto, Annex: auto
              Line coding: AUTO-TCPAM
              Configured/actual rate: AUTO/0 kbps
              Modem status: DOWN_NOT_READY,Condition: NO_COND_
      DSL Stats:
              Power Back Off: 0dB
              LoopAttn: 0dB, SnrMargin: 0dB
              Current 15 minute statistics (Time elapased 1 seconds)
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Previous 15 minute statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Current 24 hr statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Previous 24 hr statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
      EFM Stats:
              EFM-TC Tx: data frames: 0
              EFM-TC Rx: data frames: 0
      SHDSL wire-pair (1) is in DSL DOWN state
              LOSWS Defect alarm: none
              SNR Margin alarm: none
              Loop Attenuation alarm: none
              Termination: CPE, Line mode: EFM Auto, Annex: auto
              Line coding: AUTO-TCPAM
              Configured/actual rate: AUTO/0 kbps
              Modem status: DOWN_NOT_READY,Condition: NO_COND_
      DSL Stats:
              Power Back Off: 0dB
              LoopAttn: 0dB, SnrMargin: 0dB
              Current 15 minute statistics (Time elapased 1 seconds)
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Previous 15 minute statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Current 24 hr statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
              Previous 24 hr statistics
                      ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
      EFM Stats:
              EFM-TC Tx: data frames: 0
              EFM-TC Rx: data frames: 0
      SHDSL wire-pair (2) is in DSL DOWN state
              LOSWS Defect alarm: none
```

```
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
        SHDSL wire-pair (3) is in DSL DOWN state
                LOSWS Defect alarm: none
                SNR Margin alarm: none
                Loop Attenuation alarm: none
                Termination: CPE, Line mode: EFM Auto, Annex: auto
                Line coding: AUTO-TCPAM
                Configured/actual rate: AUTO/0 kbps
                Modem status: DOWN_NOT_READY,Condition: NO_COND_
        DSL Stats:
                Power Back Off: 0dB
                LoopAttn: 0dB, SnrMargin: 0dB
                Current 15 minute statistics (Time elapased 1 seconds)
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 15 minute statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Current 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
                Previous 24 hr statistics
                        ES:0, SES:0, CRC:0, LOSWS:0, UAS:0
        EFM Stats:
                EFM-TC Tx: data frames: 0
                EFM-TC Rx: data frames: 0
Group 1 is not configured
Group 2 is not configured
Group 3 is not configured
```

19 VFR and Underlay Fragmentation

Topics:

- Feature history for VFR and underlay fragmentation
- Virtual Fragmentation Reassembly
- Configure VFR and underlay fragmentation
- Verify VFR and underlay fragments

Introduces the principles of VFR and underlay fragmentation, providing foundational knowledge for their integration into network environments.

Feature history for VFR and underlay fragmentation

Outlines the feature history for VFR and underlay fragmentation, detailing packet fragmentation and reassembly mechanisms, their benefits for SDWAN network efficiency, and release information for VFR-enabled platforms.

Table 51: Feature History

Feature Name	Release Information	Description
VFR and Underlay Fragmentation	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	In Cisco Catalyst SD-WAN networks, the VFR (Virtual Fragmentation Reassembly) actively fragments and reassembles packets. The packets undergo fragmentation to improve transportation efficiency while passing through a VFR-enabled Cisco IOS XE Catalyst SD-WAN device. The VFR reassembles the fragmented packets to match the original incoming packet. The reassembled packet contains critical Layer 4 or Layer 7 information necessary for proper reception by the destination device. Underlay fragmentation refers to the process of breaking down a large data packet into smaller fragments at the network layer. Underlay fragmentation allows the successful transmission of packets that exceed the MTU limitations by breaking them down into manageable fragments and ensuring their reliable delivery.

Virtual Fragmentation Reassembly

Introduces Virtual Fragmentation Reassembly, describing how it collects and reassembles IP fragments lacking Layer 4 information, supports dynamic ACL analysis for security, enables features like firewall and NAT inspection, and outlines default and reassembly modes for packet processing and interoperability.

A Virtual Fragmentation Reassembly (VFR) is a Cisco feature that

- allows a router to collect IP fragments even if they lack necessary Layer 4 information,
- performs dynamic access control list (ACL) analysis for security, and
- enables security features such as Cisco IOS Firewall and NAT to inspect traffic.

While transmitting data across a network, due to various network constraints, the original data packets fragment into smaller fragments to facilitate seamless transmission. While the packets travel through the Cisco IOS XE Catalyst SD-WAN device, they are fragmented. VFR allows fragmented packets to be reassembled efficiently before reaching their destination.

Packet reassembly modes

In Cisco Catalyst SD-WAN network, data packets undergo reassembly in two modes:

- Default mode: Packets are virtually reassembled by default. Upon the delivery of the first fragment, each feature in the network receives the entire payload of the virtually reassembled packet. When the last fragment is received, the remaining features reassemble the packet. The original packet is fragmented, and the internal fragment information structure is shared. The fragments are then queued for refragmentation based on the fragment-offset sequence. The VFR mechanism reconstructs the packets using information from the fragment headers, such as fragment identifiers, sequence numbers, and offsets.
- Reassembly mode: Packets undergo physical reassembly, and fragment header information isn't saved. Upon receiving the last fragment, the fragments reassemble via a metapacket, and the internal fragment information structure is released.

If the packets were originally fragmented using the default mode, they undergo reassembly as if they were the original incoming packets. On the other hand, when the reassembly mode is utilized to virtually fragment the packets, they experience fragmentation based on the MTU of the egress interface before reassembly.

Some features (such as NAT, Cisco IOS XE Firewall, IPSec) automatically enable VFR to obtain Layer 4 or Layer 7 information.

When a particular interface enables VFR, it overrides the existing firewall or NAT's VFR mode configuration by default, ensuring interoperability with the firewall or NAT.

Underlay fragmentation

An underlay fragmentation is a network layer process that

- breaks down large data packets that exceed the Maximum Transmission Unit (MTU) size supported by the Cisco Catalyst SD-WAN network infrastructure,
- enables the transmission of packets that exceed MTU limitations by fragmenting them into smaller parts, and
- ensures the successful delivery of these fragments across the network.

Benefits of VFR and underlay fragmentation

- VFR enables the Cisco IOS XE Firewall to create appropriate dynamic access control lists (ACLs) to protect the network from various fragmentation attacks.
- VFR is responsible for detecting and preventing various types of fragment attacks.
- VFR drops all fragments within a fragment chain if an overlap of a fragment is detected.

Prerequisites for configuring VFR and underlay fragmentation

Properly configure the Maximum Transmission Unit (MTU) size on all network devices. The MTU defines the maximum packet size that can be transmitted without fragmentation. Ensure the MTU is set appropriately on every device along the network path to avoid unintended underlay fragmentation.

Restrictions for configuring VFR and underlay fragmentation

Fragment handling requirements

- The VFR process requires all fragments within an IP datagram to be present. If load balancing causes fragments to be sent to different devices, VFR may fail and drop fragments.
- If any fragments in a series of fragmented packets are lost or arrive out of order, the reassembly process may fail, resulting in incomplete or corrupted packets.

Feature integration

- VFR is designed to operate with features that require fragment reassembly, such as Cisco Catalyst SD-WAN NAT and IPsec. By default, NAT, Crypto-based IPSec, and NAT64 internally enable or disable VFR on an interface when these features are activated. If multiple features enable VFR on the same interface, VFR uses a reference count to track the number of enabling features. VFR is automatically disabled when the reference count reaches zero.
- The VFR CLIs are unavailable under port-channel sub-interfaces.

Underlay fragmentation

- The underlay fragmentation mechanism operates only at the network layer and is limited to the underlying network infrastructure. It does not manage fragmentation and reassembly across multiple network segments or provide end-to-end fragmentation handling.

Use cases for VFR and underlay fragmentation

Networks such as long-distance connections such as a connection between an airplane and aiport signal towers, can experience interruptions, due to the time it takes for large packets to traverse these links. When VFR is enabled, the fragments reassemble into a complete datagram, and then are fragmented within the Cisco Catalyst SD-WAN tunnel interface. With this, the first fragment is sent out first and there is no interruption in receiving the packets.

Underlay fragmentation helps in fragmenting large packets into smaller sizes, and reconstruct the packet back into the original one. This improves the overall application performance.

Boost mode

The boost mode helps in resolving one of the identified bottlenecks related to the memory management of fragments within the data plane of the network.

The boost mode is disabled by default on Cisco IOS XE Catalyst SD-WAN devices.

Before Cisco IOS XE Catalyst SD-WAN Release 17.12.1a	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a and later
The memory allocation to reassembly of fragments occured from a global chunk, necessitating a lock in period for the memory until the reassembly is complete. This leads to potential competition among multiple threads for the same global chunk and results in waiting for the same memory.	The boost mode enhances performance by utilizing CVLA, an alternative data plane memory infrastructure. Unlike the chunk mechanism, CVLA is lock-free and is an efficient memory management mechanism within Cisco IOS XE devices.

Configure VFR and underlay fragmentation

Guides configuration of VFR and underlay fragmentation, including procedures for configuring underlay fragmentation via configuration groups and CLI, enabling boost mode with CLI commands, and configuring VFR with CLI.

Use one of these methods to configure VFR and underlay fragmentation:

- Configuration groups
- CLI commands

Configure underlay fragmentation using a configuration group

Before you begin

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a

Procedure

1. From the Cisco SD-WAN Manager, choose **Configuration** > **Configuration Groups**.
2. Click **Transport & Management Profile**.
3. Select the desired transport profile and click **Edit**.
4. Click **Edit Ethernet Interface** > **Tunnel**.
5. Enable **Allow Fragmentation** and **MTU To Max**.
6. Click **Save**.

Configure VFR using CLI commands

Enable virtual fragment reassembly (VFR) on interfaces to prevent fragmentation attacks and ensure correct packet delivery.

You can configure VFR using a CLI template. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

> **Note**
>
> By default, CLI templates execute commands in global config mode.

Procedure

Enable VFR.

- Enable VFR for IPv4 packets
- Enable VFR for IPv6 packets

To enable VFR for IPv4 packets on Inbound Interface Traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR on the interface and specify the maximum threshold values.

ip virtual-reassembly [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example

Here is the complete configuration example to enable VFR for IPv4 packets:

```
interface GigabitEthernet5
ip virtual-reassembly max-reassemblies 64 max-fragments 16 mode default timeout
 5
```

To enable VFR for IPv4 packets on outbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for outbound interface traffic on the interface and specify the maximum threshold values.

ip virtual-reassembly-out [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example

Here is the complete configuration example to enable VFR for IPv4 packets:

```
interface GigabitEthernet 5
ip virtual-reassembly-out mode default max-fragments 64
```

To enable VFR for IPv6 packets on inbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for IPv6 packets on inbound interface traffic.

ipv6 virtual-reassembly [**in**|**out**] [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example

Here is the complete configuration example to enable VFR for IPv6 packets:

```
interface GigabitEthernet 5
ipv6 virtual-reassembly in mode default max-fragments 25
max-reassemblies 1024
```

To enable VFR for IPv6 packets on outbound interface traffic perform the following steps:

a) Configure an interface type and enter interface configuration mode.

interface *interface-type interface-number*

b) Enable VFR for IPv6 packets on outbound interface traffic.

ipv6 virtual-reassembly [**in**|**out**] [**max-reassemblies** *number*] [**max-fragments** *number*] [**timeout** *seconds*] [**mode** *modes*][**drop-fragments**]

Example

Here is the complete configuration example to enable VFR for IPv6 packets:

```
interface GigabitEthernet 5
ipv6 virtual-reassembly out mode default max-fragments 25
```

Configure underlay fragmentation using CLI commands

You can configure underlay fragmentation using CLI templates. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

Note

By default, CLI templates execute commands in global config mode.

This section provides example CLI configurations to configure underlay fragmentation.

Procedure

1. Enter the config-sdwan mode

   Example

   ```
   sdwan
   ```

2. Configure an interface type and enter interface configuration mode.

   Example

   ```
   interface interface-name interface-number
   ```

3. Configure the tunnel interface.

   Example

   ```
   tunnel-interface
   ```

4. Skip Layer 3 fragmentation and clear overlay DF bit.

   Example

   ```
   inner-fragmentation-disable
   ```

5. Perform the encapsulation for the GRE interface of the TLOC.

   Example

   ```
   encapsulation gre
   ```

   Only GRE encapsulation is supported for underlay fragmentation in Cisco IOS XE Catalyst SD-WAN Release 17.12.1a.

Here is the complete configuration example to enable underlay fragmentation:

```
sdwan
interface GigabitEthernet1
tunnel-interface
inner-fragmentation-disable
encapsulation gre
```

Enable boost mode using CLI commands

You can enable boost mode using a CLI template. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

> **Note**
>
> By default, CLI templates execute commands in global config mode.

Procedure

Enable the boost mode.

Example

```
platform ipreass boost-mode
```

Here is the complete configuration example to enable the boost mode:

```
platform ipreass boost-mode
```

Verify VFR and underlay fragments

Provides instructions for validating and verifying VFR and underlay fragmentation configuration and packet handling using diagnostic commands.

Boost mode

The following is a sample output of the **show platform hardware qfp active infrastructure cvla client handles** command:

```
Device# show platform hardware qfp active infrastructure cvla client handles
Handles for cpp 0:

---------------------

Entity name: IPREASS_CVLA_0

Handle: 0xeea45000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF_AOR

Handle: 0xeea0d000
```

```
Number of allocations: 0

Memory allocated: 0


Entity name: NBAR_CVLA_ENTITY

Handle: 0xee946000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF Chunk 2

Handle: 0xef929000

Number of allocations: 0

Memory allocated: 0


Entity name: FNF Chunk 1

Handle: 0xef928000

Number of allocations: 0

Memory allocated: 0




----------------------
```

The boost mode is disabled if entity for **IPREASS_CVLA_*** is not displayed. Once the boost mode is disabled, the **IPREASS_CVLA_*** disappears after 64 seconds.

VFR for IPv4 packets

The following is a sample output from the **show ip virtual-reassembly** command:

```
Device# show ip virtual-reassembly GigabitEthernet 5
GigabitEthernet5:

   Virtual Fragment Reassembly (VFR) is ENABLED [out]

   Concurrent reassemblies (max-reassemblies): 16

   Fragments per reassembly (max-fragments): 32

   Reassembly timeout (timeout): 3 seconds
```

```
   Drop fragments: OFF


   Current reassembly count:0

   Current fragment count:0

   Total reassembly count:12

   Total reassembly timeout
```

The example shows if VFR for IPv4 is enabled or not. **Virtual Fragment Reassembly (VFR) is ENABLED [out]** signifies that VFR is enabled. The total packets that underwent reassembly are also displayed.

VFR for IPv6 packets

The following is a sample output from the **show ipv6 virtual-reassembly** command:

```
Device# show ipv6 virtual-reassembly GigabitEthernet 5
GigabitEthernet5:

   IPv6 Virtual Fragment Reassembly (IPV6VFR) is ENABLED [out]

   IPv6 configured concurrent reassemblies (max-reassemblies): 64

   IPv6 configured fragments per reassembly (max-fragments): 16

   IPv6 configured reassembly timeout (timeout): 3 seconds

   IPv6 configured drop fragments: OFF


   IPv6 current reassembly count:0

   IPv6 current fragment count:0

   IPv6 total reassembly count:12

   IPv6 total reassembly timeout count:0
```

The example shows if VFR for IPv6 is enabled or not. **Virtual Fragment Reassembly (VFR) is ENABLED [out]** signifies that VFR is enabled. The total packets that underwent reassembly are also displayed.

Underlay fragmentation

The following is a sample output from the **show ip traffic interface GigabitEthernet 1** command:

```
Device# show ip traffic interface GigabitEthernet 1
GigabitEthernet 1 statistics :

  Rcvd:  11048818 total, 749458331 total_bytes

         0 format errors, 0 hop count exceeded

         0 bad header, 0 no route

         0 bad destination, 0 not a router
```

```
         0 no protocol, 0 truncated

         0 forwarded

         0 fragments, 0 total reassembled

         0 reassembly timeouts, 0 reassembly failures

         0 discards, 0 delivers

  Sent:  0 total, 0 total_bytes 0 discards

         0 generated, 0 forwarded

         0 fragmented into, 0 fragments, 0 failed

  Mcast: 0 received, 0 received bytes

         0 sent, 0 sent bytes

  Bcast: 0 received, 1256 sent
```

The example shows the number of packets that were sent and received, including the total number of packets. A change from the previous number of packet tranfer indicates that underlay fragmentation is enabled.

The following is a sample output from **show sdwan ftm tloc-list** command:

```
Device# show sdwan ftm tloc-list

--- LOCAL  TLOC LIST ---




Id: 32775 (binosId=0xf808007f), Tenant Id: 0      LocalTLOC, num-nhops: 0
,hash: 0, ref: 1    SLA 0x0:0x0 Inner-fragmentation
-disable: No


[TOTAL-LOCAL-TLOC:1]




--- REMOTE TLOC LIST ---




Id: 32768 (binosId=0xf808000f), Tenant Id: 0       SLAClass, num-nhops: 0
,hash: 0, ref: 1    SLA 0x0:0x0
 num-active-nhops: 0


Id: 32774 (binosId=0xf808006f), Tenant Id: 0       SLAClass, num-nhops: 1
```

```
,hash: 0, ref: 1    SLA 0x1:0x0
 [nhop1] nhop-Id: 19   , Type: IPsec     , Encap: IPSEC SLA
0x1:0x0hw_record_index:   5 198.100.1.5/12366->198.100.1.6/12346 pr
oto 0x800 hash 0x13 wan-if 3 tloc 32774 R-color mpls local-tloc 32775 L-color
 mpls BFD UP tloc-capability 0 SLA 0x1:0x0 weight
1   pref 0

 num-active-nhops: 1


[TOTAL-REMOTE-TLOC:2]


--- PENDING TLOC LIST (is_pending_updates:FALSE)---




[TOTAL-PENDING-TLOC:0]



--- UNMATCHED TLOC LIST (is_pending_updates:FALSE)---



[TOTAL-UNMATCHED-TLOC:0]



--- TENANT LOCAL  TLOC LIST ---
```

The example displays all the local TLOCs in the network.

The following is a sample output from **show platform software sdwanR0 next-hop overlay all** command:

```
Device# show platform software sdwan R0 next-hop overlay all

Show sdwan next-hop oce all :


OCE ID: 0xf800013f, OCE Type: SDWAN_NH_OVERLAY

Overlay: client_handle (nil), ppe addr (nil)

  overlay encap: ipsec

  src-ip: 198.100.1.5, src-port: 12366

  dst-ip: 198.100.1.6, dst-port: 12346

  flags: 0x0, linktype: MCP_LINK_IP, ifhandle: 15, encap type: MCP_ET_NULL
```

```
 encap rewrite: 00

 mtu: 1446, fixup: 0x0, fixup_flags_2: 0x0, color: mpls, phy_oce_handle: 31,
nh_overlay_h: 0xf800013f
   Overlay_CFG:

   encap type: ipsec

   src-ip: 198.100.1.5, src-port: 12366

   dst-ip: 198.100.1.6, dst-port: 12346

   local_system_ip: 1.1.1.1

   remote_system_ip: 2.2.2.2

   local_color: 2 [mpls], remote_color: 2 [mpls]

   wan_ifindex: 8 [GigabitEthernet2], tun_ifindex: 15 [Tunnel0]

   tun_adj_id: 0, l2_adj_id: 0x1f, tunnel_qos_dpidx: 0x0

   bfd-ld: 20005, ipsec_flow_id: 603979786, session_id: 5

   Inner-fragmentation-disable: yes
```

The example demonstrates whether the inner fragmentation is disabled or enabled in a particular next-hop overlay.

The following is a sample output from **show platform software sdwan F0 next-hop overlay all** command:

```
Device# show platform software sdwan F0 next-hop overlay all

OCE ID: 0xf800013f, OCE Type: SDWAN_NH_OVERLAY

Overlay: client_handle 0x63d321350ba0, ppe addr db910710

  overlay encap: ipsec

  src-ip: 198.100.1.5, src-port: 12366

  dst-ip: 198.100.1.6, dst-port: 12346

  flags: 0x0, linktype: MCP_LINK_SDWAN, ifhandle: 15, encap type: MCP_ET_ARPA

  encap rewrite: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

 mtu: 1446, fixup: 0x0, fixup_flags_2: 0x800000, color: mpls, phy_oce_handle:
31, nh_overlay_h: 0xf800013f
   Overlay_CFG:

   encap type: ipsec

   src-ip: 198.100.1.5, src-port: 12366

   dst-ip: 198.100.1.6, dst-port: 12346

   local_system_ip: 1.1.1.1

   remote_system_ip: 2.2.2.2
```

```
    local_color: 2 [mpls], remote_color: 2 [mpls]
    wan_ifindex: 8 [GigabitEthernet2], tun_ifindex: 15 [Tunnel0]
    tun_adj_id: 0, l2_adj_id: 0x1f, tunnel_qos_dpidx: 0x0
    bfd-ld: 20005, ipsec_flow_id: 603979786, session_id: 5
    Inner-fragmentation-disable: yes
```

The example demonstrates whether the inner fragmentation is disabled or enabled in all the available overlays.

20 Packet-order preservation during tunnel underlay reassembly

Topics:

- Feature history for packet order preservation during tunnel underlay reassembly
- Fragmentation and reassembly packet order preservation processes
- Use cases for packet-order preservation during tunnel underlay reassembly
- Enable boost mode in SD-WAN Manager
- Verify tunnel underlay reassembly boost mode is active

Describes packet-order preservation mechanisms used during tunnel underlay reassembly, outlining methods, potential issues, and configuration guidelines to ensure optimal network performance.

Feature history for packet order preservation during tunnel underlay reassembly

Describes the feature history of device tagging, by release.

Feature history of device tagging, by release.

Table 52: Feature History

Feature Name	Release Information	Feature Description
Packet Order Preservation during Tunnel Underlay Reassembly	Cisco IOS XE Catalyst SD-WAN Release 26.2.1 Cisco Catalyst SD-WAN Manager Release 26.2.1	Packet-order preservation for reassembly ensures that a tunnel endpoint forwards completed, reassembled packets from the same traffic flow in correct order. The feature operates with packet reassembly boost mode on supported Cisco IOS XE Catalyst SD-WAN devices and SD-WAN Manager.

Fragmentation and reassembly packet order preservation processes

Explains how tunnel underlay fragmentation and packet reassembly preserve packet order in SD-WAN deployments.

A packet order preservation for tunnel underlay reassembly is a process that:

- ensures a tunnel endpoint forwards completed, reassembled packets from the same traffic flow in order
- operates with IP reassembly boost mode on supported Cisco IOS XE Catalyst SD-WAN device, and
- performs reassembly of fragmented outer packets at the receiving network device to maintain traffic flow integrity.

Packet order preservation in boost mode

SD-WAN tunnels support underlay network fragmentation and reassembly, so large packets that exceed the underlay MTU can be split for transmission and put back together by the network. This allows SD-WAN tunnels handle fragmentation and reassembly the same way as GRE and IPsec tunnels.

In underlay fragmentation scenarios, packets from the same flow could exit the tunnel endpoint out of sequence. Packet-order preservation, enabled by boost mode, prevents this outcome by maintaining strict per-flow sequencing so packets exit in their original order. This ensures applications receive packets in the correct sequence, improving performance and maintaining integrity in SD-WAN networks.

Benefits

- Prevents tunnel underlay reassembly from introducing packet reordering within a flow.
- Helps applications that are sensitive to packets arriving out of order.
- Supports deployments in which the receiving tunnel endpoint reassembles fragmented outer IP packets.

Use cases for packet-order preservation during tunnel underlay reassembly

Explains packet-order preservation methods for tunnel underlay reassembly and their significance in maintaining flow integrity.

Use packet order preservation during tunnel underlay reassembly when tunnel encapsulation can cause outer IP packets to exceed the underlay MTU and the receiving tunnel endpoint must reassemble the resulting fragments. The packet-order preservation is useful for applications that are sensitive to reassembly-induced packet reordering and for deployments that depend on network-side reassembly.

For example, a Layer 2 VPN can carry jumbo frames from an SR-MPLS over IS-IS network. SD-WAN tunnel encapsulation increases the packet size, which can cause the outer packet to exceed the underlay MTU. After the receiving SD-WAN edge device reassembles the outer fragments, packet-order preservation keeps completed packets from the same flow in order.

Enable boost mode in SD-WAN Manager

Use this procedure to enable packet-order preservation for tunnel underlay reassembly in SD-WAN Manager.

Use this procedure to enable boost mode for packet-order preservation during tunnel underlay reassembly in SD-WAN Manager.

Before you begin

- Ensure that the configuration group exists and contains the devices on which you want to enable the packet-order preservation. For more information, refer to *Configure underlay fragmentation using a configuration group*.
- Use the CLI add-on feature template in SD-WAN Manager to configure underlay fragmentation using a configuration group. For more infromation, refer to the *Configure VFR and underlay fragmentation* section in this chapter.

Follow this step to enable boost mode for packet-order preservation during tunnel underlay reassembly.

> **Note**
>
> Boost mode is disabled by default on all Cisco IOS XE Catalyst SD-WAN devices.

Procedure

Enable boost mode using in global configuration mode.

Example

```
Device(config)# platform ipreass boost-mode
```

For more information, refer to platform ipreass boost-mode

When the boost mode is enabled, tunnel underlay reassembly preserves the order of completed packets within each flow.

Verify tunnel underlay reassembly boost mode is active

Provides the commandto verify if tunnel underlay reassembly boost mode is active on a device.

To determine whether tunnel underlay reassembly boost mode is active, use the following command:

```
Device# show platform hardware qfp active infrastructure cvla client handles
```

In the command output, the presence of an entity named `IPREASS_CVLA_*` indicates that boost mode is active.

This command verifies boost mode only. It does not verify packet order preservation during reassembly.

21 Cisco SD-WAN Controller Route Filtering by TLOC Color

Topics:

- Feature history for route filtering by TLOC color
- Route filtering mechanism by TLOC color
- Configure Cisco SD-WAN Controller route filtering by TLOC color using a CLI template
- Monitor Cisco SD-WAN Controller route filtering by TLOC color

Feature history for route filtering by TLOC color

This section lists the development milestones and release information for route filtering by TLOC color, including feature description.

This table describes the developments of this feature, by release.

Table 53: Feature History Table

Feature Name	Release Information	Description
Cisco SD-WAN Controller Route Filtering by TLOC Color	Cisco IOS XE Catalyst SD-WAN Release 17.11.1a Cisco Catalyst SD-WAN Control Components Release 20.11.1	Cisco SD-WAN Controller can reduce the number of routes advertised to routers in the network and exclude routes that are not relevant to a particular device. The controller uses the colors of TLOCs on each device to filter and reduce the number of routes. A router that only has private TLOCs does not need routes to public TLOCs. Advertising fewer routes helps to avoid reaching the send path limit for routers in the network.

Route filtering mechanism by TLOC color

Explains how the filtering mechanism identifies and excludes irrelevant routes from advertisements to individual routers.

A route filtering by TLOC color is a Cisco SD-WAN Controller feature that

- reduces the number of advertised routes to routers by excluding irrelevant routes,
- bases filtering on the colors of TLOCs associated with each device, and
- advertises only routes compatible with one or more of the router's TLOCs.

Route filtering by TLOC color allows Cisco SD-WAN Controllers to selectively advertise routes to individual routers. For each router, only the routes that match the color of one or more of its TLOCs are advertised, ensuring that only relevant routes are sent to each device.

Using route filtering, Cisco SD-WAN Controllers can reduce the number of routes they advertise to routers in the network by excluding routes that are not relevant to a particular device. The filtering is based on the colors of TLOCs on each device: for each individual router, the Cisco SD-WAN Controller advertises only routes that are compatible with one or more of the router's TLOCs.

How Cisco SD-WAN Controller route filtering by TLOC color works

Default behaviour: Cisco SD-WAN Controller route filtering by TLOC color is disabled by default.

Summary

Cisco SD-WAN Controllers apply the following logic when determining whether routes are compatible:

- A TLOC with a public color can resolve a path to a route for a TLOC with a public color on a peer device.

- A TLOC of a particular color can resolve a path to a route for a TLOC of the same color on a peer device.
- A TLOC with a public color cannot resolve a path with a TLOC in a private color set.

Public colors include default, biz-internet, public-internet, lte, 3g, red, green, blue, gold, silver, bronze, custom1, custom2, and so on. Private colors include mpls, metro-ethernet, private1, private2, and so on.

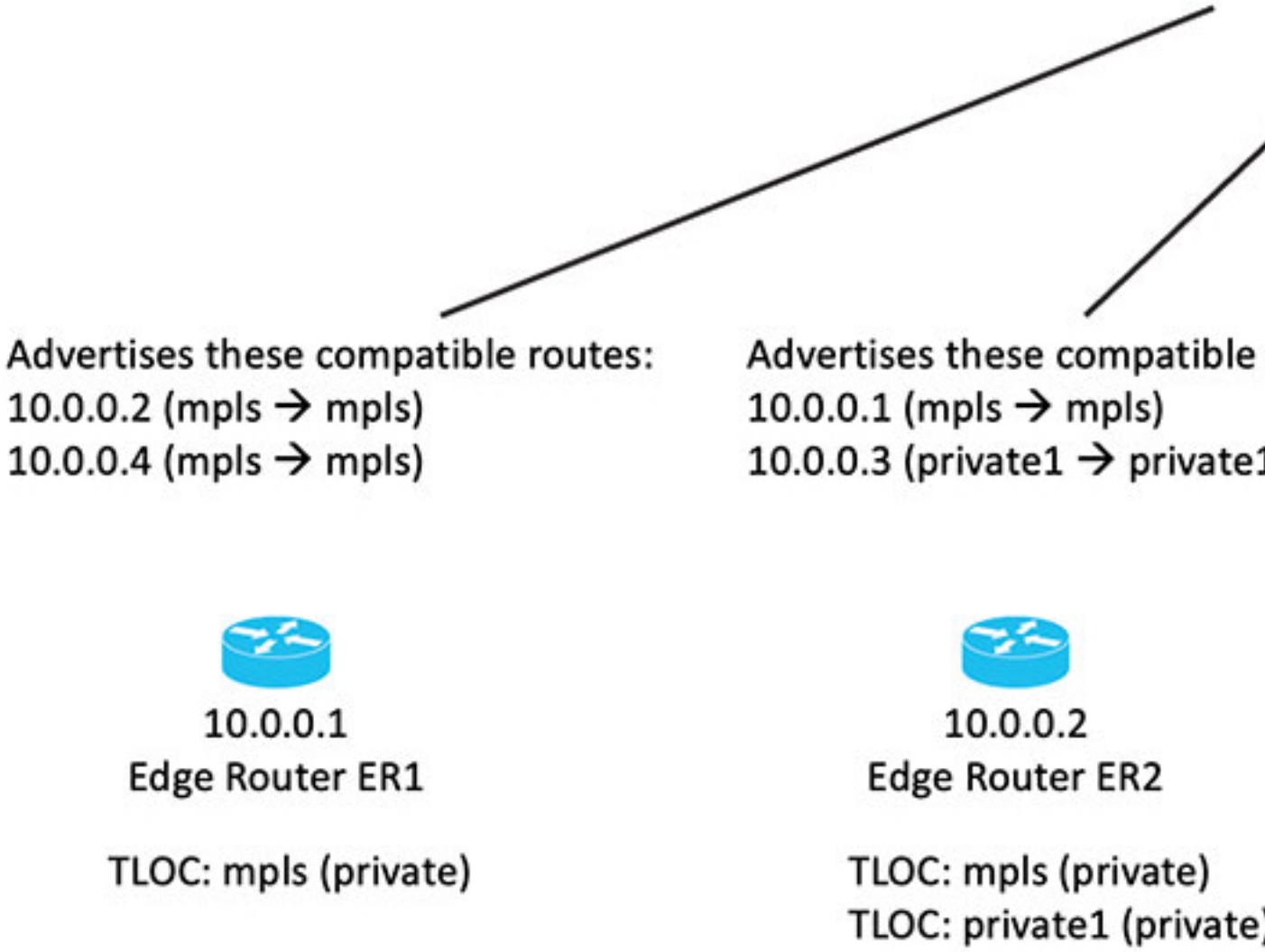
For information about private and public TLOC colors, see Unicast Overlay Routing in the *Cisco SD-WAN Routing Configuration Guide, Cisco IOS XE Release 17.x.*.

For example, if a router only has TLOCs with private colours, Cisco SD-WAN Controllers do not advertise public routes to the device. Similarly, if a router only has TLOCs with public colors, Cisco SD-WAN Controllers do not advertise private routes to the device.

The following illustration provides further detail:

Workflow

Figure 7: Cisco SD-WAN Controller Route Filtering by TLOC Color, With the Feature Enabled



If you change the color assignment of a TLOC, the device updates the Cisco SD-WAN Controllers, enabling them to adjust the Cisco SD-WAN Controller route filtering by TLOC color accordingly.

What's next

You can override the default logic if necessary and do one of the following:

- Configure two TLOC colors to be compatible even if they are incompatible by default.
- Configure two TLOC colors to be incompatible even if they are compatible by default.

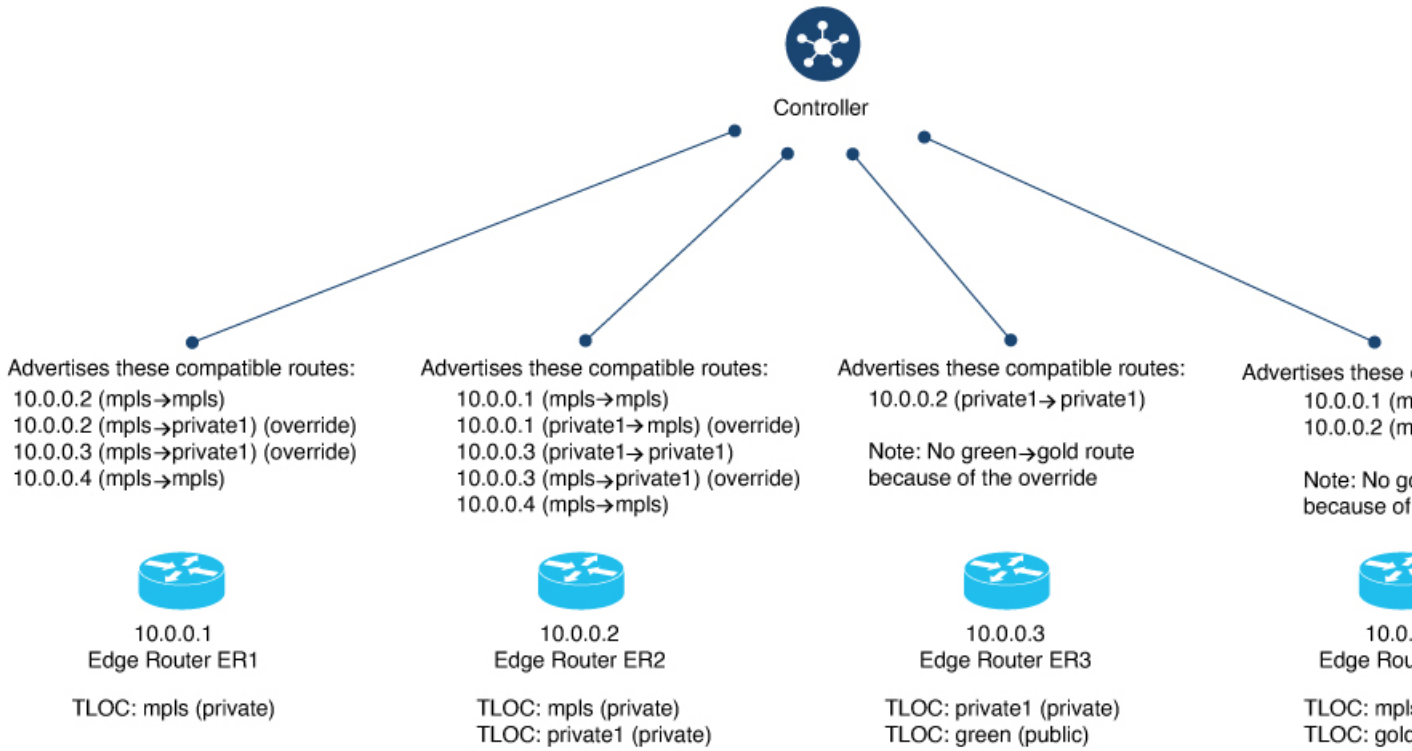
This may be helpful in specific unconventional scenarios. See the tloc-color-compatibility command in Override Default TLOC Color Compatibility for Cisco SD-WAN Controller Route Filtering by TLOC Color Using a CLI Template.

The following illustration shows an example of route filtering by TLOC color, with two overrides:

- Configure green and gold to be incompatible.

- Configure mpls and private1 to be compatible.

Figure 8: Cisco SD-WAN Controller Route Filtering by TLOC Color, With the Feature Enabled and Overrides



Routers in the network update Cisco SD-WAN Controllers when the status of their TLOCs changes. This may include reconfiguring a TLOC to a different color.

To account for temporary unavailability of a TLOC due to flapping, there is a dampening interval to delay reporting changes of TLOC status. By default, it is 60 seconds, but it can be configured to a value from 60 to 1200 seconds. For information, see Configure the Update Interval for Route Filtering by TLOC Color Using a CLI Template.

Benefits of Cisco SD-WAN Controller route filtering by TLOC color

- **Avoiding the send path limit**: Cisco SD-WAN Controller route filtering by TLOC color helps prevent routers from reaching their send path limit (for example, a limit of 32 routes), even if there are more routes available for a particular prefix.
- **Prioritizing relevant routes**: If the send path limit is set to a low value and many routes are available, filtering ensures that only relevant routes are advertised to the device. This prevents the send path limit from being reached with irrelevant routes and helps avoid routing failures.

Supported devices for Cisco SD-WAN Controller route filtering by TLOC color

Cisco IOS XE Catalyst SD-WAN devices supported for Cisco SD-WAN Controller route filtering by TLOC color.

Prerequisites for Cisco SD-WAN Controller route filtering by TLOC color

For Cisco SD-WAN Controllers to determine the compatibility of paths, the colors of TLOCs must be configured according to convention.

For example, a TLOC handling an MPLS connection must have the color mpls.

Restrictions for Cisco SD-WAN Controller route filtering by TLOC color

- When you enable Cisco SD-WAN Controller route filtering by TLOC color in a network, ensure that all you enable it on all Cisco SD-WAN Controllers in the network.
- Scenarios in which route filtering by TLOC color is enabled on some Cisco SD-WAN Controllers and disabled on others within the same network are not supported.

Configure Cisco SD-WAN Controller route filtering by TLOC color using a CLI template

Describes how you can outbound transport filter, adjust dampening intervals for flapping links, and define custom color compatibility. Use this procedure to implement control plane optimizations through CLI templates on both controllers and edge devices.

- Enable route filtering using a CLI template
- Configure the update interval for route filtering by TLOC color using a CLI template
- Override default TLOC color compatibility for Cisco SD-WAN Controller route filtering by TLOC color using a CLI template

Enable route filtering using a CLI template

Before you begin

By default, CLI templates execute commands in global configuration mode. For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

This configuration applies to a Cisco SD-WAN Controller.

Procedure

1. Enter OMP mode.

   omp

2. Enter filter-route configuration mode.

   filter-route

3. Enable route filtering.

   outbound tloc-color

```
omp
 filter-route
  outbound tloc-color
!
```

Configure the update interval for route filtering by TLOC color using a CLI template

Before you begin

By default, CLI templates execute commands in global configuration mode.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

This configuration applies to a Cisco IOS XE Catalyst SD-WAN device.

Procedure

1. Enter OMP configuration mode.

   omp

2. Configure the update interval, in seconds, in the range 60 to 1200.

   timers
   tloc-color-cap-update-interval *interval*

   Example

```
omp
  no shutdown
  ecmp-limit       6
  graceful-restart
  no as-dot-notation
  timers
   holdtime                         15
   tloc-color-cap-update-interval 120
   graceful-restart-timer         120
  exit
  address-family ipv4
   advertise ospf external
   advertise connected
   advertise static
  !
  address-family ipv6
   advertise ospf external
   advertise connected
   advertise static
  !
 !
!
```

Override default TLOC color compatibility for Cisco SD-WAN Controller route filtering by TLOC color using a CLI template

Before you begin

By default, CLI templates execute commands in global configuration mode.

For more information about using CLI templates, see *CLI Add-On Feature Templates* and *CLI Templates*.

You can override the default logic if necessary and do one of the following:

- Configure two TLOC colors to be compatible even if they are incompatible by default.

- Configure two TLOC colors to be incompatible even if they are compatible by default.

This may be helpful in specific unconventional scenarios.

This configuration applies to a Cisco SD-WAN Controller.

Procedure

1. Enter system mode.

   system

2. Enter TLOC color compatibility mode.

   tloc-color-compatibility

3. Enter one or more of the following.
   - To configure two TLOC colors to be compatible, do the following:

     compatible *first-color second-color*

   - To configure two TLOC colors to be incompatible, do the following:

     incompatible *first-color second-color*

This example does the following:

- Configures the lte and private1 TLOC colors to be compatible
- Configures the private1 and private2 TLOC colors to be compatible
- Configures the lte and default TLOC colors to be incompatible
- Configures the lte and 3g TLOC colors to be incompatible

```
system
 host-name vm1
 system-ip 10.0.10.1
 site-id               100
 tloc-color-compatibility
  compatible lte private1
  !
  compatible private1 private2
  !
  incompatible lte default
  !
  incompatible lte 3g
  !
 !
```

Monitor Cisco SD-WAN Controller route filtering by TLOC color

Describes how to verify the transport attributes advertised by peer devices and validate path compatibility logic.

- View TLOC Colors for a Device
- Check TLOC Color Compatibility

View TLOC colors for a device

When applying route filtering, the controllers use this TLOC color information to determine which routes are relevant to a device.

Procedure

To view the list of the TLOC colors that a device advertises to Cisco SD-WAN Controllers, use the **show support omp peer peer-ip** command on a Cisco SD-WAN Controller.

Example

The following example shows the TLOC colors that the peer device 10.0.0.15 is advertising–in this case, lte and 3g.

```
device#show support omp peer peer-ip 10.0.0.15 | inc color
ed bitmap: 0xc0, TLOC color supported list: lte 3g
```

Check TLOC color compatibility

Procedure

To check the compatibility of TLOC colors, use the **request support omp tloc-color-compat** command.

Example

The following example requests information about whether the 3g and lte colors are compatible. These are both public TLOC colors, so they are compatible:

```
vsmart# request support omp tloc-color-compat 3g lte
Checking compatibility for colors:3g and lte
TLOC colors: 3g and lte are compatible
```

The following example requests information about whether the 3g and mpls TLOC colors are compatible. They are incompatible:

```
vsmart# request support omp tloc-color-compat 3g mpls
Checking compatibility for colors:3g and mpls
TLOC colors: 3g and mpls are incompatible
```

22 Network Slicing

Topics:

- Feature history of network slicing
- Network slicing
- Configure network slicing
- Verify network slicing using CLI

Outlines the fundamental concepts and technical details of network slicing, enabling users to understand its role in segmenting network resources for tailored services.

Feature history of network slicing

Details the evolution and key milestones of network slicing features, providing historical context for current capabilities and enhancements.

Table 54: Feature History

Feature Name	Release Information	Description
Cellular Network Slicing	Cisco IOS XE Catalyst SD-WAN Release 17.16.1a Cisco Catalyst SD-WAN Manager Release 20.16.1	This feature allows multiple networks to exist on the same physical network to optimize the network for different traffic types.

Network slicing

Explains core principles, architecture, and use cases of network slicing to illustrate its practical applications and advantages in modern network environments.

Network slicing is a network architecture solution that:

- partitions a single physical network into multiple virtual slices
- allows each slice to operate with customized network functions and resources
- allocates dedicated resources for diverse service types or customer requirements.

5G network slicing is supported only on 5G standalone networks. It is not available on 5G non-standalone (NSA) networks. The network provider defines a specific set of network functions and resources for each slice or service type. Service type defines the expected behavior of a network slice, specifying its features and services.

The following service types are supported in network slicing:

- Enhanced Mobile Broadband (eMBB): used for high-bandwidth and low-latency applications.
- Ultra-reliable low latency communications (URLLC): used for highly reliable and low-latency transmission requirements.
- Massive Internet of Things (MIoT): used for sending small quantities of data.

Configure network slicing

Provides step-by-step instructions for configuring network slicing, guiding users through the setup to enable segment-based network management.

Before you begin

Create a configuration group for Cisco Catalyst Cellular Gateways using **Workflows** > **Create Cellular Gateway Group**. On the **Configuration Groups** page, the resulting configuration group is labelled cellular gateway in the **Device Solution** column.

For information about creating configuration groups and applying them to devices, see the Using Configuration Groups section of *Cisco Catalyst SD-WAN Configuration Groups, Cisco IOS XE Catalyst SD-WAN Release 17.x*.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Create a Configuration Group. A new configuration group is created.
3. In the configuration group, click **Add Profile**.
4. Choose **Transport & Management Profile** from the drop-down list.
5. Click **Edit** to add features to the Transport and Management profile and click **Add New Feature**.
6. Choose **Cellular Controller** from the **Add Feature** drop-down list.
7. Choose **Cellular Profile** from the **Add Fetaure** drop-down list.

Table 55: Cellular Settings

Field	Description
Primary Slot	Choose a SIM slot to designate it as primary. Range: 0, 1 Default: 0
SIM SLOT 0 Cellular Profile	
Profile Id	Profile ID. You can click **Add** to add multiple profiles.
Access Point Name	Access point name, from your service provider.
Authentication Method	Authentication method (**none**, **pap**, **chap**, **pap_or_chap**) indicated by your service provider.
Username	Username for authentication, as indicated by your service provider.
Password	Password for authentication, as indicated by your service provider.
Packet Data Network Type	Packet data network type (**IPv4**, **IPv6**, **IPv4v6**), as indicated by your service provider.
Attach Profile	Choose a attach profile from the defined profiles.
Data Profile	Choose a data profile from the defined profiles. You can use the same profile for the attach profile and data profile.
SIM SLOT 1 Cellular Profile	
See the fields described for SIM slot 0.	

8. For the configured **Cellular Controller**, choose the **Cellular Profile** or click **Add New** to add a new cellular profile.
9. Configure **Cellular Profile** using the following table.

Table 56: Cellular Profile Settings

Field	Description
Type	Choose a feature from the drop-down list.

Field	Description
Feature Name	Enter a name for the feature. The name can be up to 128 characters and can contain only alphanumeric characters.
Description	Enter a description of the feature. The description can be up to 2048 characters and can contain only alphanumeric characters.
Profile ID	Enter the identification number of the profile to use on the router. Range: 1 through 15
Access Point Name	Enter the name of the gateway between the service provider network and the public internet. It can be up to 32 characters long.
Authentication	Choose the authentication method used for the connection to the cellular network. It can be **none**, **pap**, **chap**, or **pap_chap**.
Profile Username	Enter the user name to use when making cellular connections for web services. It can be 1 to 32 characters. It can contain any alphanumeric characters, including spaces.
Profile Password	Enter the user password to use when making cellular connections for web services. The password is case-sensitive and can be clear text, or an AES-encrypted key. From Cisco Catalyst SD-WAN Manager Release 20.15.1, when you enter the password as clear text, Cisco SD-WAN Manager encrypts the password. When you view the configuration preview, the password appears in its encrypted form.
Packet Data Network Type	Choose the packet data network (PDN) type of the cellular network. It can be IPv4, IPv6, or IPv4v6.
No Overwrite	Enable this option to overwrite the profile on the cellular modem. By default, this option is disabled.
Slice Type	Choose the network slice type (SST) for the profile. The options are: • eMBB: Enhanced Mobile Broadband, used for high data throughput. • URLLC: Ultra-Reliable Low Latency Communication, used for high reliability and low latency transmission. • MIoT: Massive IoT, used for many devices transmitting small quantities of data.
Slice Differentiator	Enter the slice differentiator (SD) for the profile. This is an optional value that enables the user equipment (UE) to use multiple slice instances of the same SST. Range: 0 through 16777214
Slot	Enter the associated SIM slot for the profile.

Configure network slicing using CLI

The following example shows how to configure a cellular network slicing using a CLI:

```
Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
 ipv4
   no-overwrite
   slot            Associated sim slot
   slice-type Associated network slice type(SST)

Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
 ipv4 slice- type
<1-3>  Slice type number: 1(eMBB), 2(URLLC), 3(MioT)
Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
 ipv4 slice- type embb
   slice-differentiator Associated Slice Differentiator(SD)  <--- if SD is
bypassed here,"FF FF FF" wll be used internally as the SD to indicate that
there is no SD associated with this SST.
   slot                            Associated sim slot

Device(config-controller)# profile id 3 apn apn-ns authentication none pdn-type
 ipv4 slice type
embb slice-differentiator
  <0-16777214>  Slice Differentiator
Device(config-controller)#
Device(config-controller)#$ pdn ipv6  slice-type miot slice-differentiator 6

  slot  Associated sim slot
```

Verify network slicing using CLI

Guides users in verifying network slicing configurations using CLI commands to ensure proper deployment and operational status.

The following is a sample output from the **show cellular** command that displays network slicing configuration and slice type details:

```
Device# show cellular profile 6

Profile 6 = INACTIVE
--------
PDP Type = IPv6
Access Point Name (APN) = TestNSSAI6_URLLC1
Authentication = None
S-NSSAI Slice Type is URLLC
S-NSSAI Slice Differentiator = 1
```

23 Transport Gateway

Topics:

- Feature history for transport gateway
- Transport gateways for connecting networks in Cisco SD-WAN
- Configure a router as a transport gateway
- Configure the transport gateway path preference
- Configure the site type for a router
- Verify the site type of a router using the CLI
- Verify a transport gateway configuration using the CLI

Feature history for transport gateway

This section lists the development milestones and release information for transport gateways, including feature description.

This table describes the developments of this feature, by release.

Table 57: Feature History Table

Feature Name	Release Information	Description
Transport Gateway	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Manager Release 20.12.1	A transport gateway operates as the hub in a hub-and-spoke routing topology. It offers the advantage of achieving this topology without requiring complex routing policy configuration. The following are some uses of a transport gateway: • Providing connectivity to routers in disjoint underlay networks • Serving as a gateway (hub) for all traffic in one discrete network to reach another discrete network, such as directing all local network traffic to a cloud gateway

Transport gateways for connecting networks in Cisco SD-WAN

Explains how the transport gateways facilitate communication between routers that lack direct connectivity, such as bridging physical LANs to cloud networks.

A transport gateway is a network device that:

- connects routers that may or may not have direct connectivity,
- simplifies the process of providing connectivity between disjoint networks (such as between a physical LAN and a cloud-based network), and
- enables indirect connectivity without the complexity and limitations of manual control policy configuration.

A transport gateway facilitates communication between routers that are not directly connected, often bridging networks that are physically or logically separate (for example, connecting a traditional LAN to a cloud-based network).

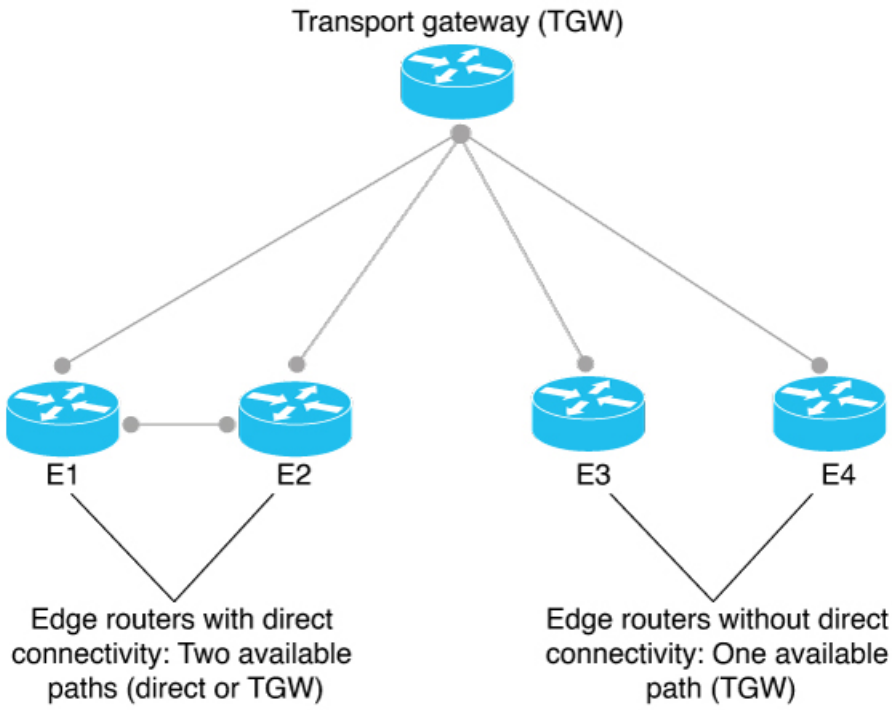
A transport gateway connects routers that may or may not have direct connectivity. A common use case for transport gateways is to provide connectivity between routers in disjoint networks, such as between a physical LAN and a cloud-based network.

Without a transport gateway, one method of configuring indirect connectivity for these routers is to create a control policy that configures routes through an intermediate device with connectivity to both networks. This provides indirect connectivity between the disjoint routers. This approach has the following problems:

- Complexity: Configuring a control policy to advertise prefixes is complicated.
- Potential unavailable traffic endpoint: The control policy cannot detect whether a device or a configured route is unavailable. This can lead to packet loss if a route becomes unavailable.

Configuring a router to operate as a transport gateway solves the same issue, but with a simpler configuration process.

Figure 9: Transport Gateway



In the context of Cisco Catalyst SD-WAN, you can efficiently configure a hub-and-spoke routing topology by using transport gateways as hubs. This enables you to create the hub-and-spoke topology without requiring complex routing policy configuration. For information, see *Hub-and-Spoke*.

How a router functions as a transport gateway

Summary

Starting from Cisco Catalyst SD-WAN Manager Release 20.16.1, routes that are reoriginated from a site through the transport gateway are filtered out by the Cisco SD-WAN Controller. These reoriginated routes are not sent back to the originating site or the sites which share same site ID as the originating site. The reoriginated routes are only distributed to different sites within the Cisco Catalyst SD-WAN network.

This change in routing mechanism is also backported to Cisco Catalyst SD-WAN Manager Release 20.15.2, and Cisco Catalyst SD-WAN Manager Release 20.12.5.

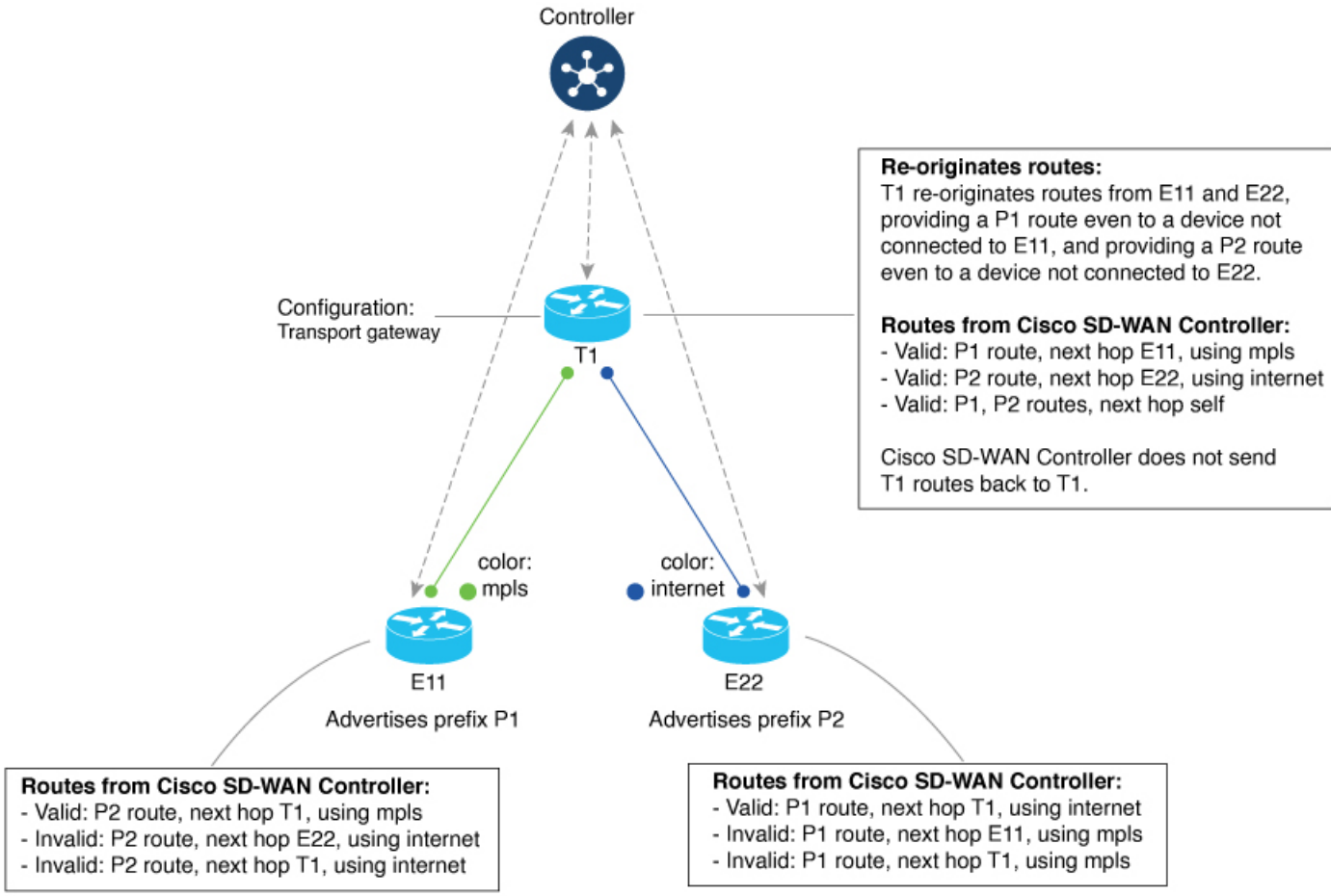
When a router is configured to function as a transport gateway, it does the following for each route that it learns from the Cisco SD-WAN Controllers:

1. The transport gateway re-originates each route, substituting its own TLOCs as the next hop for the routes. This means that it substitutes its TLOCs as the next hop for each route.
2. The transport gateway advertises the re-originated routes to the Cisco SD-WAN Controllers.
3. The transport gateway attaches its own affinity attribute to routes that it re-originates. In scenarios in which routers in the network have re-originated routes available from more than one transport gateway, the routers apply affinity group preference logic to choose a route.

In the following illustration, E11 advertises prefix P1 and E22 advertises prefix P2. E11 and E22 are disjoint–they do not have direct connectivity. The transport gateway re-originates routes from E11 and E22, providing a P1 route to E22 and a P2 route to E11.

Workflow

Figure 10: Transport Gateway Re-Originating Routes



Site type

One part of configuring networks to use transport gateways is assigning a site type parameter to routers in the network. Site type helps to classify the intended function of a router, helping to define its position within the topology. Site type values include br, branch, cloud, spoke, type-1, type-2, and type-3.

After assigning site types, you can configure routers to prefer a transport gateway path only for traffic destined to a specific site type. This provides greater granularity when configuring a preference for transport gateway paths.

Site types are arbitrary, with no specific meaning, except br (border router) and spoke, which have specific uses for Multi-Region Fabric or intent-based hub-and-spoke topology, respectively.

Site Type Inheritance

Every OMP vRoute and TLOC originated from a router inherits the site type attributes of the router.

For information about configuring a site type for a router, see Configure the Site Type for a Router Using Cisco SD-WAN Manager.

OMP best path logic and transport gateway path preference

In general, when multiple paths are available between two routers, the overlay management protocol (OMP) applies best path selection logic to choose the best path. The best path selection logic is biased toward paths with fewer hops.

Summary

When you have configured a transport gateway, you can configure routers to apply a specific preference for transport-gateway-re-originated paths, if available. This alters the OMP best path calculation to include the transport gateway, according to the details of the configuration, as described below.

For information about configuring the preference for transport-gateway-re-originated paths, see Configure the Transport Gateway Path Preference.

Workflow

This table describes the best path logic.

	Router Configuration		Resulting Best Path Behavior
	Transport Gateway Path Behavior	**Specify Site Type(s)**	
1.	Not configured	Not applicable	(This is the default behavior.) Prefer a direct path.
	Prefer Transport Gateway Path	No	Prefer a transport-gateway path over a direct path.
	Prefer Transport Gateway Path	Yes	For a transport-gateway path that matches a specified site type, prefer a transport-gateway path over a direct path. For a transport-gateway path that does not match a specified site type, prefer a direct path over a transport-gateway path.
	Do ECMP Between Direct and Transport Gateway Paths	No	Treat a direct path and a transport-gateway path as equal.
	Do ECMP Between Direct and Transport Gateway Paths	Yes	For a transport-gateway path that matches a specified site type, treat a direct path and a transport-gateway path as equal. For a transport-gateway path that does not match a specified site type, prefer a direct path over a transport-gateway path.

As described earlier, a transport gateway attaches its own affinity attribute to paths that it re-originates. In scenarios in which routers in the network have re-originated paths available from more than one transport gateway, the routers apply affinity group preference logic to choose a path.

How the transport gateway configuration works

Summary

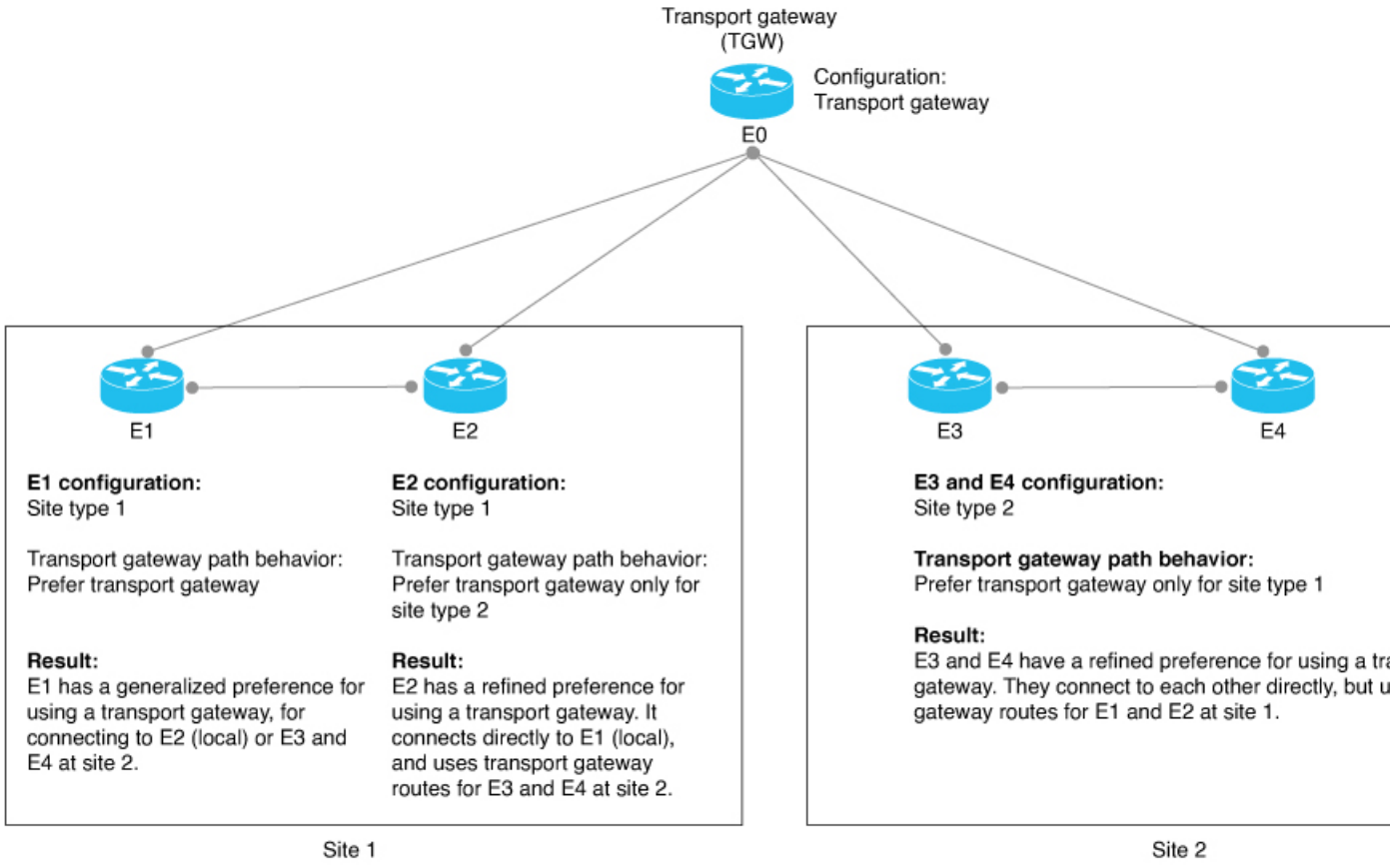
1. To configure a router to function as a transport gateway, use a System feature template or CLI add-on template. See Configure a Router as a Transport Gateway Using Cisco SD-WAN Manager.
2. To configure routers to use the transport gateway path, use an OMP feature template or CLI add-on template. See Configure the Transport Gateway Path Preference Using Cisco SD-WAN Manager. You can configure the OMP logic as follows:

- Prefer a transport gateway path over a direct path.
- Prefer a transport gateway path only for specific traffic, according to the site type attribute. See Configure the Site Type for a Router Using Cisco SD-WAN Manager.
- Consider direct paths and transport gateway paths as equal.

The following figure shows how routers in a network can operate with a transport gateway, preferentially directing all traffic or specific traffic through transport gateway routes.

Workflow

Figure 11: Edge Routers and Transport Gateway Path Preference



The devices in the illustration are configured as follows:

	Device	Configuration
1.	E0	a. Configure as a transport gateway. • By feature template: In a Cisco System template, use the **Transport Gateway** field. • By CLI add-on template: `system transport-gateway enable`

Device	Configuration

E1	**a.** Configure the site type as type-1.
	• By feature template:
	In a Cisco System template, use the **Site Type** field.
	• By CLI add-on template:
	`system site-type type-1`
	b. For best path, configure a preference for transport gateway routes.
	• By feature template:
	In an OMP template, use the **Transport Gateway Path Behavior** field. Choose the **Prefer Transport Gateway Path** option.
	• By CLI add-on template:
	`omp best-path transport-gateway prefer`

E2	**a.** Configure the site type as type-1.
	• By feature template:
	In a Cisco System template, use the **Site Type** field.
	• By CLI add-on template:
	`system site-type type-1`
	b. For best path, configure a preference for transport gateway routes for traffic to type-2 devices.
	• By feature template:
	In an OMP template, use the **Transport Gateway Path Behavior** field. Choose the **Prefer Transport Gateway Path** option. In the **Site Types** field, choose **type-2**.
	• By CLI add-on template:
	`omp best-path transport-gateway prefer transport-gateway-settings type-2`

Device	Configuration

E3 and E4	**a.** Configure the site type as type-2.
	• By feature template:
	In a Cisco System template, use the **Site Type** field.
	• By CLI add-on template:
	`system site-type type-2`
	b. For best path, configure a preference for transport gateway routes for traffic to type-1 devices.
	• By feature template:
	In an OMP template, use the **Transport Gateway Path Behavior** field. Choose the **Prefer Transport Gateway Path** option. In the **Site Types** field, choose **type-1**.
	• By CLI add-on template:
	`omp best-path transport-gateway prefer transport-gateway-settings type-1`

Use cases for transport gateways

In this use case, an organization needs to bridge a local network with a cloud services network, such as Azure or AWS. Edge routers in the local and cloud networks lack direct connectivity.

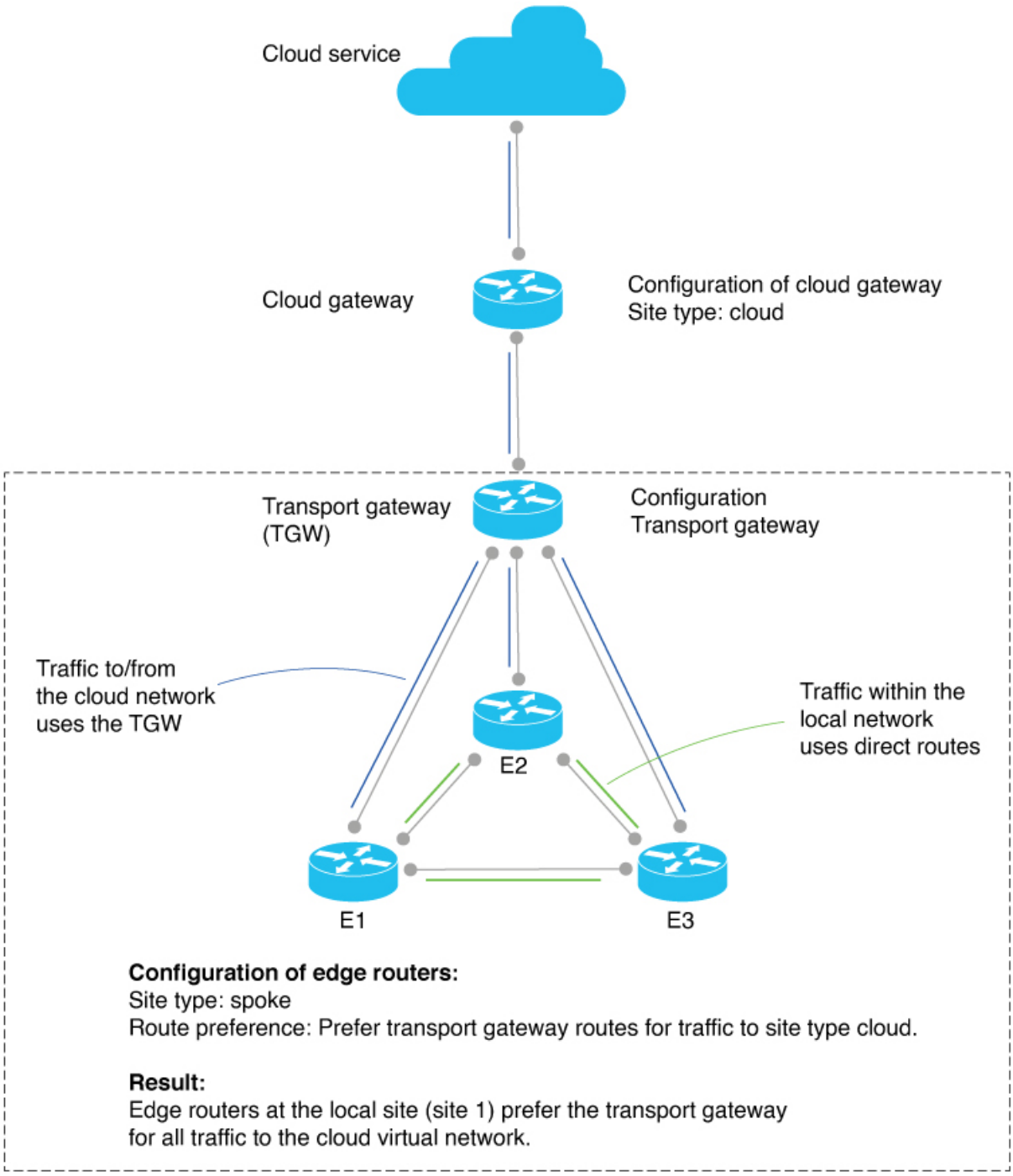
To create a transport gateway to bridge the local and cloud networks, network administrators configure the devices as follows:

Intent	Devices to Configure	Configuration
Configure the cloud gateway router with site type cloud.	Cloud gateway router	**1.** Configure the site type as cloud. • By feature template: In a Cisco System template, use the **Site Type** field. • By CLI template: **system site-type cloud**
Deploy a transport gateway to operate as a hub for cloud-destined traffic from devices in a local network. The transport gateway attracts the cloud-destined traffic and routes it to the cloud gateway for the cloud-based network.	Transport gateway router	**1.** Enable as a transport gateway. • By feature template: In a Cisco System template, use the **Transport Gateway** field. • By CLI template: **system transport-gateway enable**

Intent	Devices to Configure	Configuration
Traffic within the local network uses direct routes, not transport gateway routes. Traffic from the local network to the cloud uses a transport gateway route.	Edge routers in the local network	1. Use a transport gateway route for all cloud-destined traffic. • By feature template: In an OMP template, use the **Transport Gateway Path Behavior** field. Choose the **Prefer Transport Gateway Path** option. • By CLI template: **omp best-path transport-gateway prefer transport-gateway-settings cloud** 2. Configure the site type as spoke. • By feature template: In a Cisco System template, use the **Site Type** field. • By CLI template: **system site-type spoke**

The following illustration shows the topology and configuration:

Figure 12: Transport Gateway Topology and Configuration



Restrictions for transport gateways

Restriction	Description
Resource demands of transport gateway functionality	Because of the resource demands of transport gateway functionality, we recommend enabling this only on a high-performance device with CPU and memory resources to handle the additional load. The specific resource requirements depend on your networking environment.

Restriction	Description
Multiple transport gateways: best path	If you enable transport gateway functionality on multiple devices, edge routers apply best path selection logic to determine the best path. This may include multiple transport gateway paths.
Multiple transport gateways: preventing routing loops	If you enable transport gateway functionality on multiple devices within network, the Cisco SD-WAN Controllers for the network do the following to avoid creating routing loops: When a Cisco SD-WAN Controller receives a route re-originated by one transport gateway, it does not advertise the route to another transport gateway. Avoiding advertising a transport gateway route to another transport gateway prevents routing loops.
On-demand tunnels	You cannot configure dynamic on-demand tunnels for a device configured as a transport gateway. However, edge routers that are not operating as transport gateways can use on-demand tunnels. For information about dynamic on-demand tunnels, see Dynamic On-Demand Tunnels in the *Cisco SD-WAN Systems and Interfaces Configuration Guide, Cisco IOS XE Release 17.x*.

Configure a router as a transport gateway

Describes how to configure a router as a transport gateway using the CLI template or Cisco SD-WAN Manager.

- Configure a router as a transport gateway using the CLI
- Configure a router as a transport gateway using SD-WAN Manager

Configure a router as a transport gateway using a CLI template

Before you begin

By default, CLI templates execute commands in global configuration mode.

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Procedure

1. Enter system configuration mode.

```
system
```

2. Enable transport gateway functionality.

```
transport-gateway enable
```

> **Note**
>
> To disable transport gateway functionality, use the **no** form of the command.

```
system
  transport-gateway enable
```

Configure a router as a transport gateway using Cisco SD-WAN Manager

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Do one of the following:
   - To create a new System template, click **Add Template**, choose a device type, and click **Cisco System**.
   - To edit an existing System template, locate a System template in the table of existing feature templates, click **...** adjacent to the template, and choose **Edit**.
4. In **Basic Configuration**, in the **Transport Gateway** field, choose **On**.
5. Click **Save** if creating a new template, or **Update** if editing an existing template.

Configure the transport gateway path preference

Outlines how to assign router site types and configure path preferences for transport gateways.

The following sections describe methods for configuring a router's best path decision to handle transport-gateway-re-originated paths.

- Configure the transport gateway path preference using a CLI Template
- Configure the transport gateway path reference using Cisco SD-WAN Manager

Configure the transport gateway path preference using a CLI Template

Procedure

1. Enter sdwan configuration mode.

   sdwan

2. Enter system OMP configuration mode.

   omp

3. Configure the transport gateway path preference, using one of the following options:

   best-path transport-gateway {prefer | ecmp-with-direct-path}

Option	Description
ecmp-with-direct path	For devices that can connect through a transport gateway and through direct paths, apply equal-cost multi-path (ECMP) to all available paths.
prefer	For devices that can connect through a transport gateway, use only the transport gateway paths, even if other paths are available.

4. (Optional) Specify one or more site types to which to apply the transport gateway behavior. For information about how the Site Types parameter operates together with the Transport Gateway Path Behavior parameter, see OMP Best Path Logic and Transport Gateway Path Preference.

   omp best-path transport-gateway-settings site-types *site-types*

Option	Description
site-types	Include one or more of the following site types, separated by spaces: **cloud**, **branch**, **br**, **type-1**, **type-2, type-3**

> **Note**
>
> To use this command, ensure that you use the **omp best-path transport-gateway prefer** command in the previous step.

The following example configures a device to prefer transport gateway routes.

```
sdwan
omp
  omp best-path transport-gateway prefer
```

The following example configures a device to prefer transport gateway routes only for traffic destined to sites with site type cloud.

```
sdwan
omp
  omp best-path transport-gateway prefer
  omp best-path transport-gateway-settings site-types cloud
```

Configure the transport gateway path preference using Cisco SD-WAN Manager

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Do one of the following:

- To create a new OMP template, click **Add Template**, choose a device type, and click **Cisco OMP**.
- To edit an existing OMP template, locate a OMP template in the table of existing feature templates, click **...** adjacent to the template, and choose **Edit**.

4. In the **Best Path** section, in the **Transport Gateway Path Behavior** field, choose **Global** mode and choose one of the following options:

Option	Description
Do ECMP Between Direct and Transport Gateway Paths	For devices that can connect through a transport gateway and through direct paths, apply equal-cost multi-path (ECMP) to all available paths.
Prefer Transport Gateway Path	For devices that can connect through a transport gateway, use only the transport gateway paths, even if other paths are available.

> **Note**
>
> If you do not configure this field, by default, routers favor a direct path as the best path.

5. (Optional) Click the **Site Types** field and choose one or more site types to which to apply the transport gateway behavior. For information about how the Site Types parameter operates together with the Transport Gateway Path Behavior parameter, see OMP Best Path Logic and Transport Gateway Path Preference.
6. Click **Save** if creating a new template, or **Update** if editing an existing template.

Configure the site type for a router

Describes how to configure router site types using a CLI template or Cisco SD-WAN Manager.

- Configure the site type for a router using a CLI template
- Configure the site type for a router using Cisco SD-WAN Manager

Configure the site type for a router using a CLI template

Before you begin

By default, CLI templates execute commands in global configuration mode.

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Procedure

1. Enter system configuration mode.

```
system
```

2. Configure up to four site types for the router. Possible values are br, branch, cloud, spoke, type-1, type-2, and type-3.

```
site-type site-type
```

> **Note**
>
> To disable transport gateway functionality, use the **no** form of the command.

Example

The following example configures a router site type as cloud:

```
system
  site-type cloud
```

The following example configure a router with site types cloud and branch:

```
system
  site-type cloud branch
```

Configure the site type for a router using Cisco SD-WAN Manager

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Do one of the following:
   - To create a new System template, click **Add Template**, choose a device type, and click **Cisco System**.
   - To edit an existing System template, locate a System template in the table of existing feature templates, click **...** adjacent to the template, and choose **Edit**.
4. In **Basic Configuration**, click **Site Type** and choose a type from the drop-down list .
5. Click **Save** if creating a new template, or **Update** if editing an existing template.

Verify the site type of a router using the CLI

Confirms the assigned functional role of a device by executing operational commands on the router.

Procedure

Use the **show sdwan omp summary** command on a device to verify the site type configuration of a router. The output includes a site-type field and the configured value.

Example

In this example, the router is configured with a site type, spoke:

```
Device#show sdwan omp summary
...
site-type    SPOKE
...
```

Verify a transport gateway configuration using the CLI

Validates that a device is actively functioning as a transit hub by checking the running configuration and OMP status.

Procedure

Use the **show sdwan running-config system** command on a device to check whether it is configured as a transport gateway. In the output, **transport-gateway enable** indicates that it is configured.

```
Device#show sdwan running-config system
system
system-ip            192.168.1.1
domain-id            1
site-id              11100
region 1
!
role                 border-router
transport-gateway enable
...
```

You can also use the **show sdwan omp summary** command on a device to check whether it is configured as a transport gateway. In the output, **transport-gateway enabled** indicates that transport gateway functionality is enabled.

24 Symmetric Routing

Topics:

- Feature history for symmetric routing
- Symmetric routing mechanisms for Cisco SD-WAN
- Translating OMP metrics for devices outside of the overlay network
- Symmetric routing scenarios
- Configure symmetric routing
- Configure a router to translate OMP metrics to BGP or OSPF using CLI commands
- Verify symmetric routing
- Monitor RIB metric translation

Feature history for symmetric routing

Lists the development milestones and release information for symmetric routing, including feature description.

This table describes the developments of this feature, by release.

Table 58: Feature history

Feature name	Release information	Description
Symmetric Routing	Cisco IOS XE Catalyst SD-WAN Release 17.12.1a Cisco Catalyst SD-WAN Control Components Release 20.12.1	You can use affinity groups, affinity group preferences, and RIB metric translation to ensure devices route traffic flows symmetrically across the network. Symmetric routing supports a wide range of network topologies, including Multi-Region Fabric. Transport gateways translate RIB metrics into control plane protocols such as BGP and OSPF to extend symmetric routing beyond the overlay network. This translation applies the path-preference configuration to routers outside the overlay network, including routers in a data center LAN.

Symmetric routing mechanisms for Cisco SD-WAN

Explains symmetric routing and its importance for bidirectional services like NAT and Firewalls.

A symmetric routing is a traffic flow property that

- uses the same path for traffic in both directions
- maintains a consistent return path between endpoints, and
- supports features that require bidirectional path symmetry.

Some networking functionality requires symmetric routing to operate correctly, including Cisco NBAR2, Cisco Zone-Based Firewall (ZBF), Cisco Unified Threat Defense (UTD), Cisco Application Quality of Experience (AppQoE), and network address translation (NAT).

Within a Cisco Catalyst SD-WAN network, you can use affinity groups, affinity group preference, control policy, and other mechanisms to configure the network so that the preferred route between two endpoints remains consistent for traffic in both directions. This configuration ensures symmetric routing for traffic flows between those endpoints. In some scenarios, you can also ensure symmetric routing for traffic flows that extend to a device outside the Cisco Catalyst SD-WAN overlay network.

TLOC behavior when moving away from symmetric NAT

When a TLOC starts behind symmetric NAT and then moves to any other NAT type such as full cone, port-restricted cone, or restricted cone, the TLOC does not update its public IP or port. The learned NAT type of the TLOC also does not update. As a result, some or all BFD sessions for this TLOC go down. To recover from this state, you can run **clear sdwan control connections** from the edge router.

Assumption about router operation

All of this applies only when routers stay operational during a traffic flow. If a router in the path becomes inoperable, traffic must take a new route. This change can cause temporary asymmetric routing.

Benefits of symmetric routing configuration

Before Cisco IOS XE Catalyst SD-WAN Release 17.12.1a configuring symmetric routing required complex and error-prone control policies in the overlay network. These policies set up hop-by-hop routing in both directions.

In service-side routing, it required complex route-maps to maintain path symmetry in both directions.

From Cisco IOS XE Catalyst SD-WAN Release 17.12.1a onward, you can use affinity groups, affinity group preferences, and OMP metric redistribution to achieve symmetric routing. The following sections describe the details and supported scenarios.

Restrictions for symmetric routing

You cannot use both the **redistribute omp translate-rib-metric** command and the **redistribute omp metric** command together on the same device.

The **translate-rib-metric** option generates BGP attributes and OSPF metrics from OMP metrics, whereas the **metric** option configures the metrics explicitly. For information, see Translating OMP Metrics for Devices Outside of the Overlay Network.

Translating OMP metrics for devices outside of the overlay network

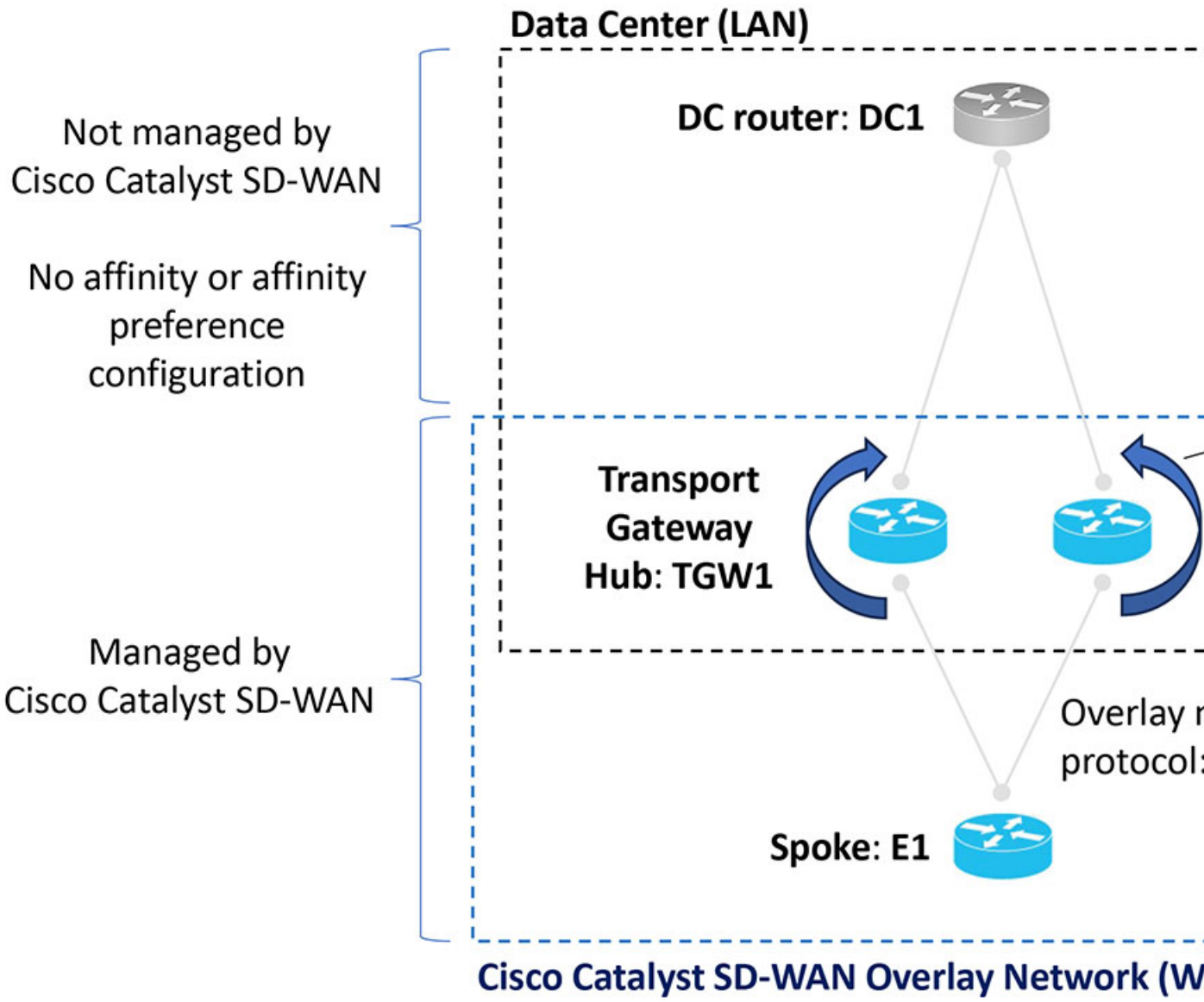
Describes how transport gateways extend symmetric routing to external LAN devices by translating RIB metrics into BGP or OSPF parameters.

A router configured as a transport gateway and operating as a hub (TGW1 in the illustration below) may conduct traffic between devices within the Cisco Catalyst SD-WAN overlay network (WAN) and a device outside of the overlay network (LAN), such as DC1 in the following illustration. This is WAN-to-LAN traffic. Note that devices outside of the overlay network are not managed by Cisco Catalyst SD-WAN.

A transport gateway translates RIB metric information into parameters used by the BGP or OSPF protocols. It uses those parameters in its BGP or OSPF routing tables, and when the transport gateway advertises routes to its BGP or OSPF neighbors, it includes the RIB-derived parameters with the routes.

These RIB-derived parameters influence path selection by devices in the LAN, helping to ensure that the LAN chooses the same path for LAN-to-WAN traffic as the overlay network uses for WAN-to-LAN traffic.

Figure 13: Translating OMP metrics



Translating OMP metrics to BGP attributes

When you enable a router to translate RIB metrics from OMP to BGP, the router uses the following OMP metric and attribute:

- OMP route metric

  Among the OMP metrics, one specific metric is named OMP.
- OMP AS-PATH

The router uses these to derive three BGP attributes:

- BGP MED
- BGP LOCAL_PREF
- BGP AS_PATH

For information about how to view OMP metrics for a route and the resulting BGP attributes, see Monitor RIB metric translation.

Translation from OMP to BGP

Table 59: Translation of OMP metrics to BGP attributes

BGP Attribute	How it is derived
BGP MED	Equal to the OMP route metric.
BGP LOCAL_PREF	255 – (OMP route metric)
BGP AS_PATH	Two possibilities: • If the **propagate-aspath** command is used: 1. If OMP AS-PATH is empty, then the router uses its own local AS value and repeats it (OMP route metric) times, with a maximum of 13 repetitions. 2. If OMP AS-PATH is not empty, then the router uses the OMP AS-PATH and prepends it with the first AS in the OMP AS-PATH (OMP route metric) times, with a maximum of 13 times. • If the **propagate-aspath** command is not used: The router uses a list of its own local AS value, repeated (OMP route metric) times, prepending the value up to a maximum of 13 repetitions.

AS-PATH propagation recommendation

In most scenarios, when you enable translation of RIB metrics using the **redistribute omp translate-rib-metric**command, also enable propagating the AS-PATH metric using the **propagate-aspath** command. If you omit this, the router treats the AS-PATH metric as empty.

The router includes these BGP attributes with the routes that it re-originates to a device in a LAN outside the overlay network that uses BGP.

BGP attributes without RIB metric translation

The table below shows combinations of OMP metrics and the BGP attributes that the router derives when RIB metric translation is not enabled.

Table 60: Translation from OMP to BGP without RIB metric translation enabled

	OMP metrics: Example combinations		Translation to BGP attributes: propagate-aspath enabled translate-rib-metric not enabled		
Example	OMP route metric	OMP AS-PATH	BGP MED	BGP LOCAL_PREF	BGP AS_PATH
1	0	100 101	1000	50	100 101
2	1	100 101	1	50	100 101

	OMP metrics: Example combinations		Translation to BGP attributes: propagate-aspath enabled translate-rib-metric not enabled		
Example	**OMP route metric**	**OMP AS-PATH**	**BGP MED**	**BGP LOCAL_PREF**	**BGP AS_PATH**
3	2	100 101	2	50	100 101
4	10	(empty)	10	50	(empty)
5	14	100 101	14	50	100 101

BGP attributes without RIB metric translation

The table below shows combinations of OMP metrics and the BGP attributes that the router derives when RIB metric translation is not enabled.

Table 61: Translation from OMP to BGP without RIB metric translation enabled

	OMP metrics: Example combinations		Translation to BGP attributes: propagate-aspath enabled translate-rib-metric not enabled		
Example	**OMP route metric**	**OMP AS-PATH**	**BGP MED**	**BGP LOCAL_PREF**	**BGP AS_PATH**
1	0	100 101	1000	50	100 101
2	1	100 101	1	50	100 101
3	2	100 101	2	50	100 101
4	10	(empty)	10	50	(empty)
5	14	100 101	14	50	100 101

Translating OMP metrics to an OSPF metric

If you do not configure the router to translate RIB metrics, it uses a default OSPF metric when it redistributes routes to a device outside the Cisco Catalyst SD-WAN overlay network. The default OSPF metric is 16777214 (hexadecimal FFFFFE).

When you enable the router to translate RIB metrics, it assigns the OMP route metric value as the OSPF metric. For example, if the OMP route metric is 10, the OSPF metric is also 10.

For information about viewing the OMP metrics for a route and the resulting BGP metrics, see Monitor RIB Metric Translation.

Symmetric routing scenarios

Use these scenarios to understand how transport gateway and Multi-Region Fabric deployments maintain symmetric routing through route re-origination and path-preference attributes across complex, multi-hop topologies and data centers.

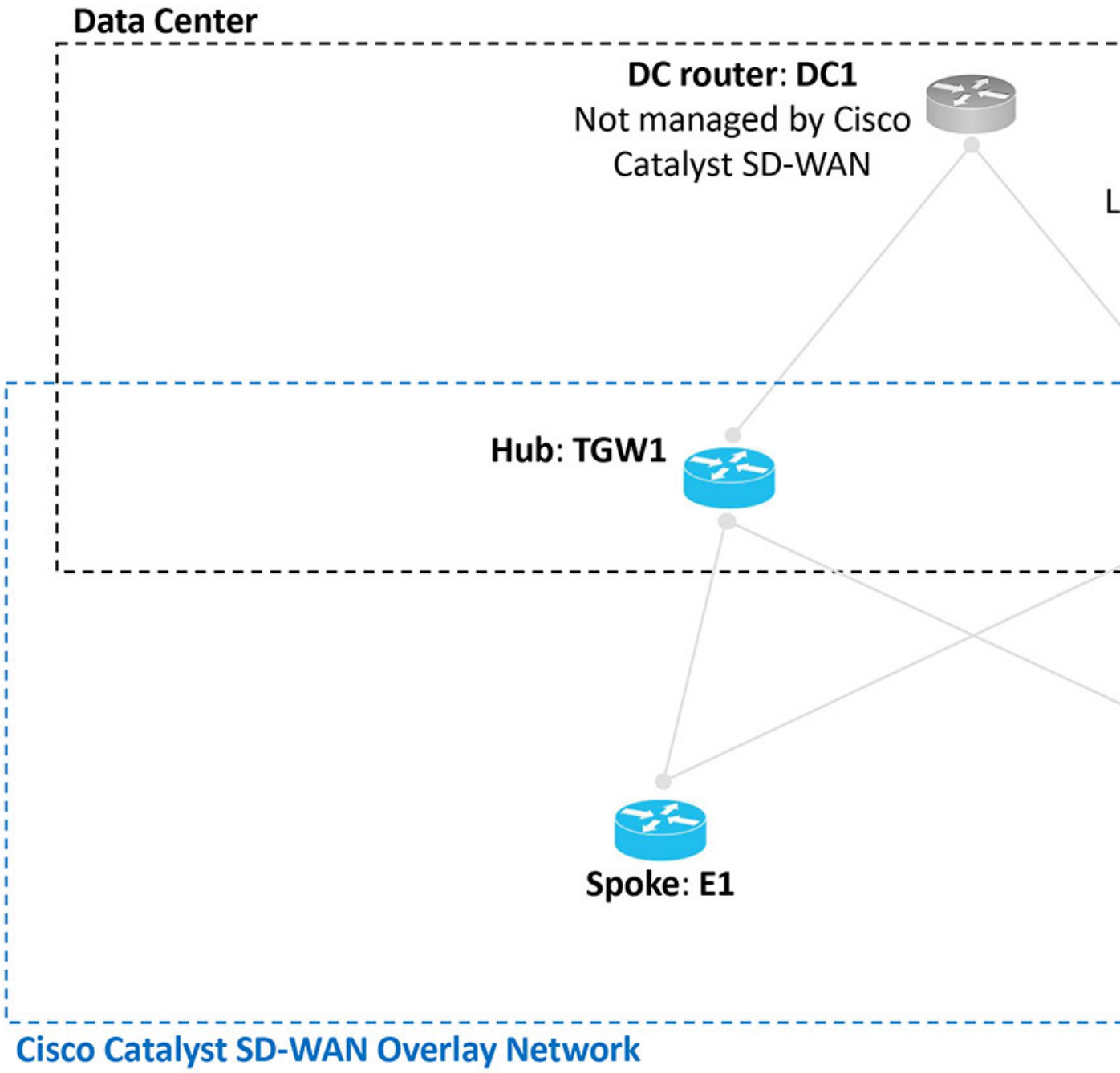
An overview of the configuration workflow helps you understand the scenarios in which Cisco Catalyst SD-WAN supports symmetric routing. The figures below shows

- a transport gateway scenario, and
- a Multi-Region Fabric scenario.

Transport gateway scenario

In the transport gateway scenario, the goal is to ensure symmetric routing between the spoke devices (E1 and E2 in the illustration) and the data center router (DC1).

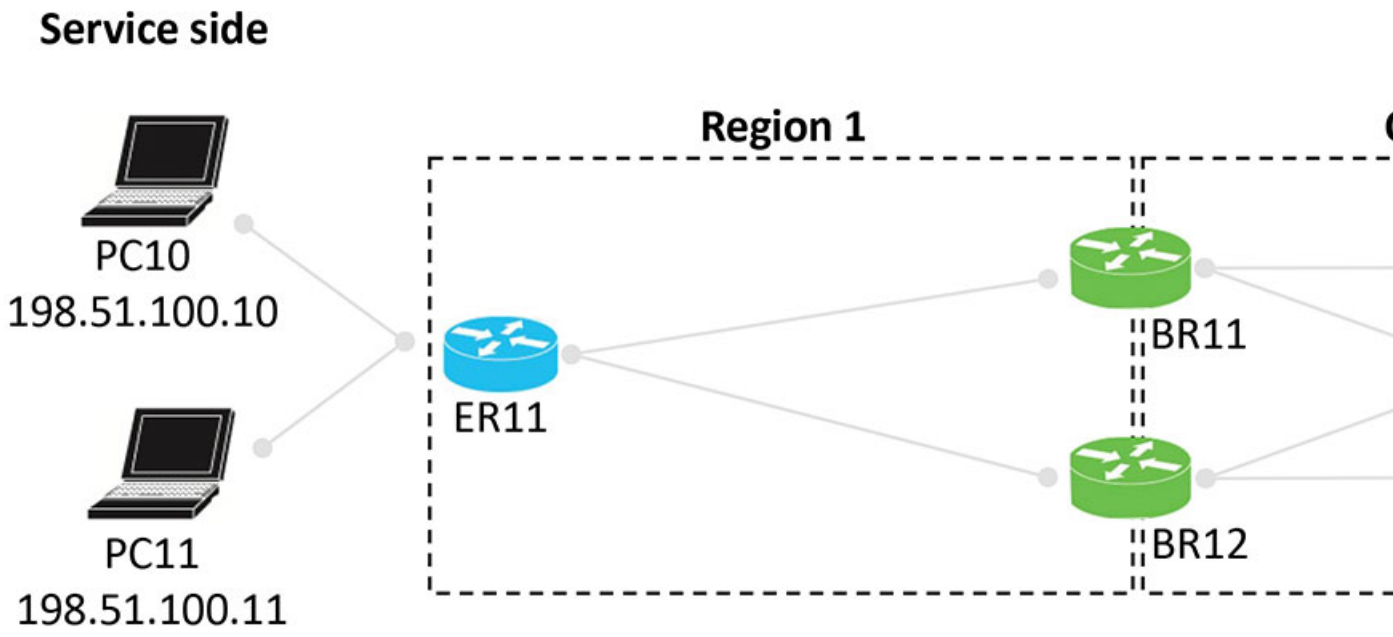
Figure 14: Transport gateway scenario with a data center LAN



Multi-region fabric scenario

In the Multi-region fabric scenario, the goal is to ensure symmetric routing between the PC devices served by edge router ER11 in Region 1, and the PC devices served by ER21 in Region 2.

Figure 15: Multi-region fabric scenario



Configuration overview

The steps below provide an overview of the configuration required for symmetric routing.

Configuration step	Devices	Description
1. Configure affinity group preference	Spoke routers Edge routers in a multi-region fabric scenario	To ensure traffic symmetry within the overlay network, configure spoke routers (or edge routers in a multi-region fabric scenario) in the network with an affinity group preference. This can be a manually configured order of preference or automatic preference. With automatic affinity preference order, a spoke device or edge router prefers paths tagged with a lower affinity group number. For configuration instructions, see Configure a Router Affinity Group or Affinity Group Preference.

Configuration step	Devices	Description
2. Configure affinity groups	Transport gateways Border routers in a multi-region fabric scenario	To ensure traffic symmetry within the overlay network, configure border routers and transport gateways with (a) an affinity group number, or (b) affinity groups per VRF for some or all VRFs that the devices handle. You can configure both (a) and (b) together. For example, if a device has a VRF range of 1 to 10, you can configure a device as follows: • System-level affinity group 10 • Affinity groups per VRF: Affinity group 20 for VRF6 through VRF 10 The result is that vRoutes in the range 1 to 5 are tagged with affinity group 10 (from the system-level affinity group), and vRoutes in the range of 6 to 10 are tagged with affinity group 20. For configuration instructions, see Configure a Router Affinity Group or Affinity Group Preference.
3. Enable translation of RIB metrics	Transport gateways Border routers in a multi-region fabric scenario	To enable symmetric routing between the overlay network and a LAN, on the border routers or transport gateways that conduct traffic with a LAN, enable translation of RIB metrics for redistribution of OMP routes to LAN routing protocols. For a full explanation, see Translating OMP Metrics for Devices Outside of the Overlay Network. For configuration instructions, see Configure a Router to Translate OMP Metrics to BGP or OSPF Using a CLI Template.

Example configurations for symmetric routing in transport gateway and multi-region fabric deployments

The illustrations below show the two scenarios described earlier, with an example configuration for each router, in accordance with the steps described here to ensure symmetric routing.

Figure 16: Transport gateway scenario with a data center LAN, showing a configuration for symmetric routing

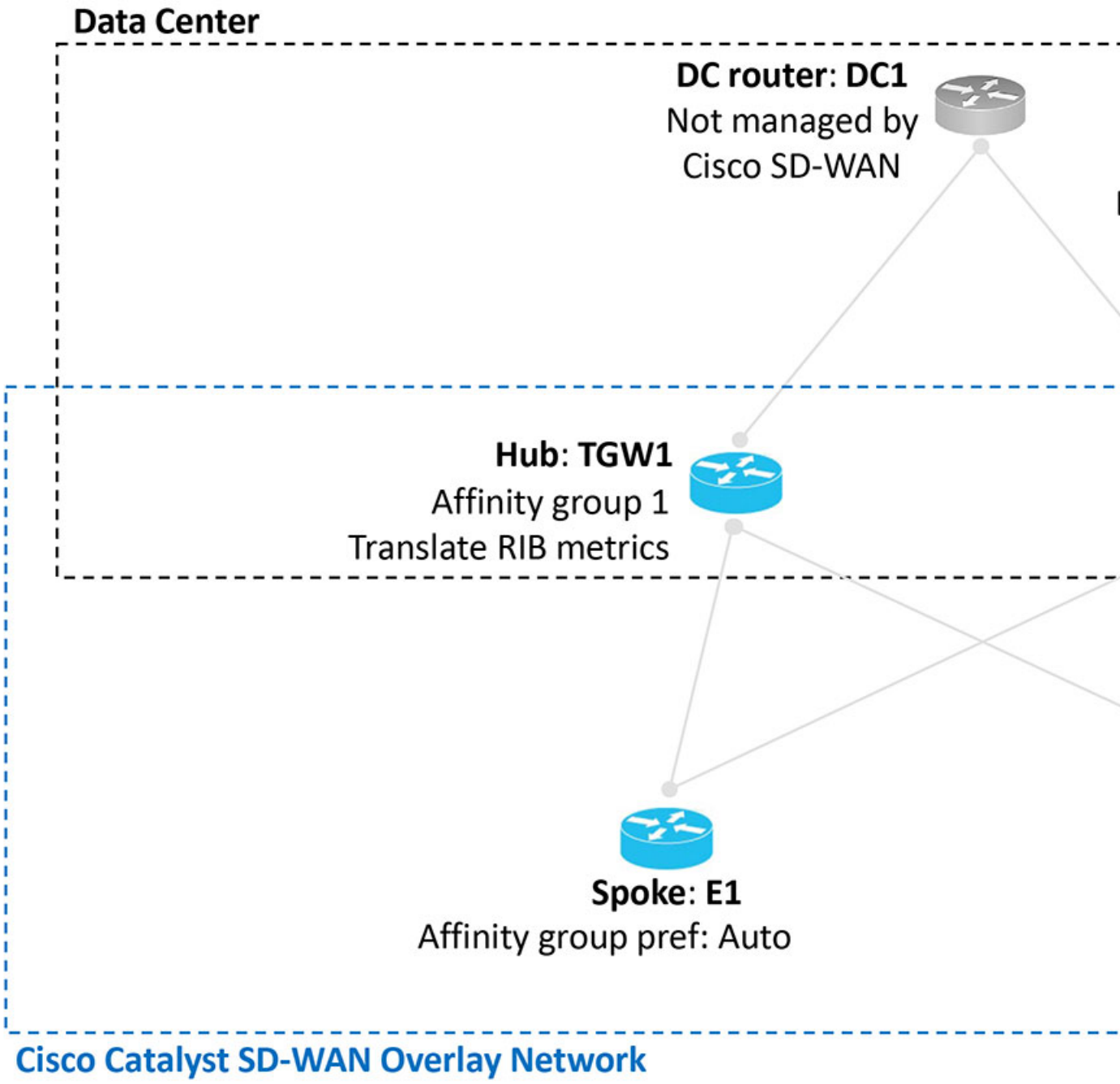
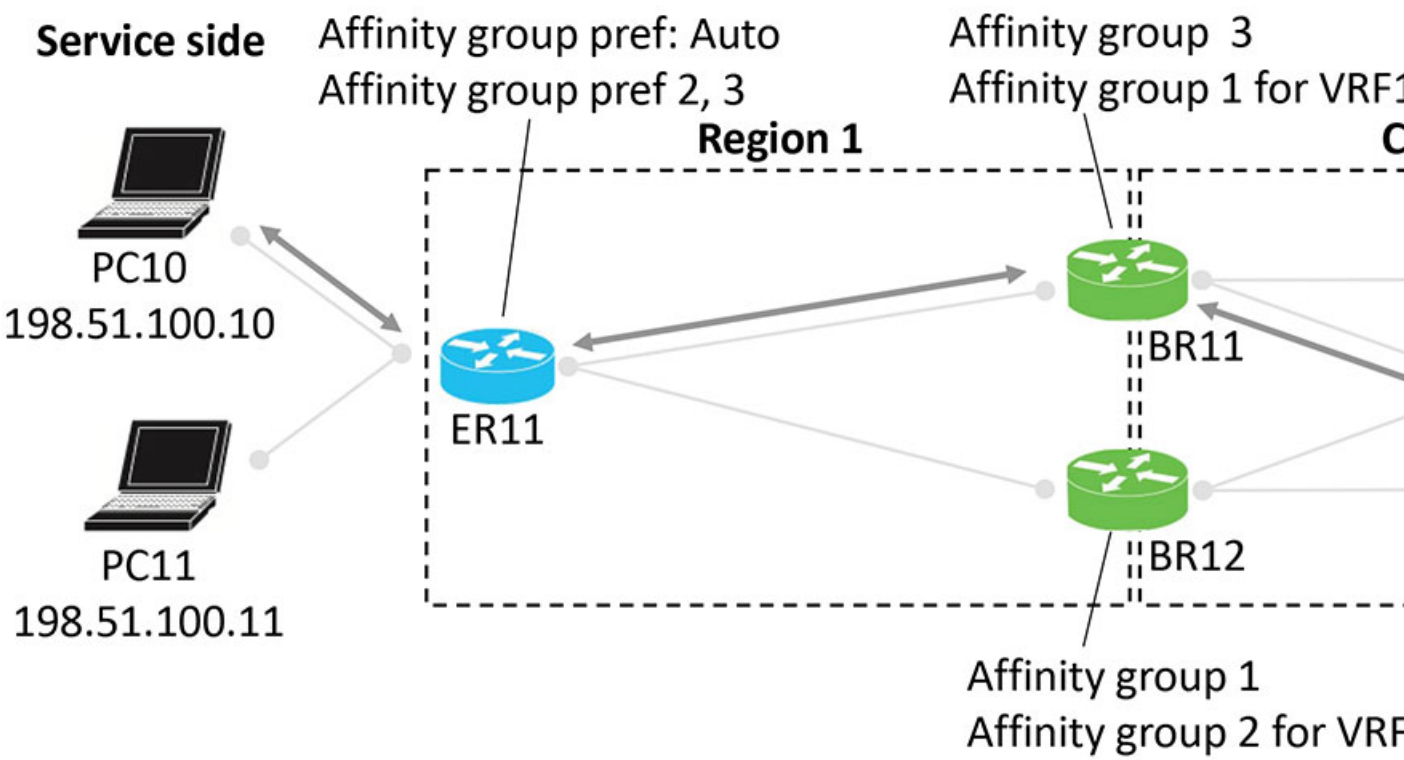


Figure 17: Multi-region fabric scenario, showing a configuration for symmetric routing



Example of configuration for symmetric routing and the mechanism

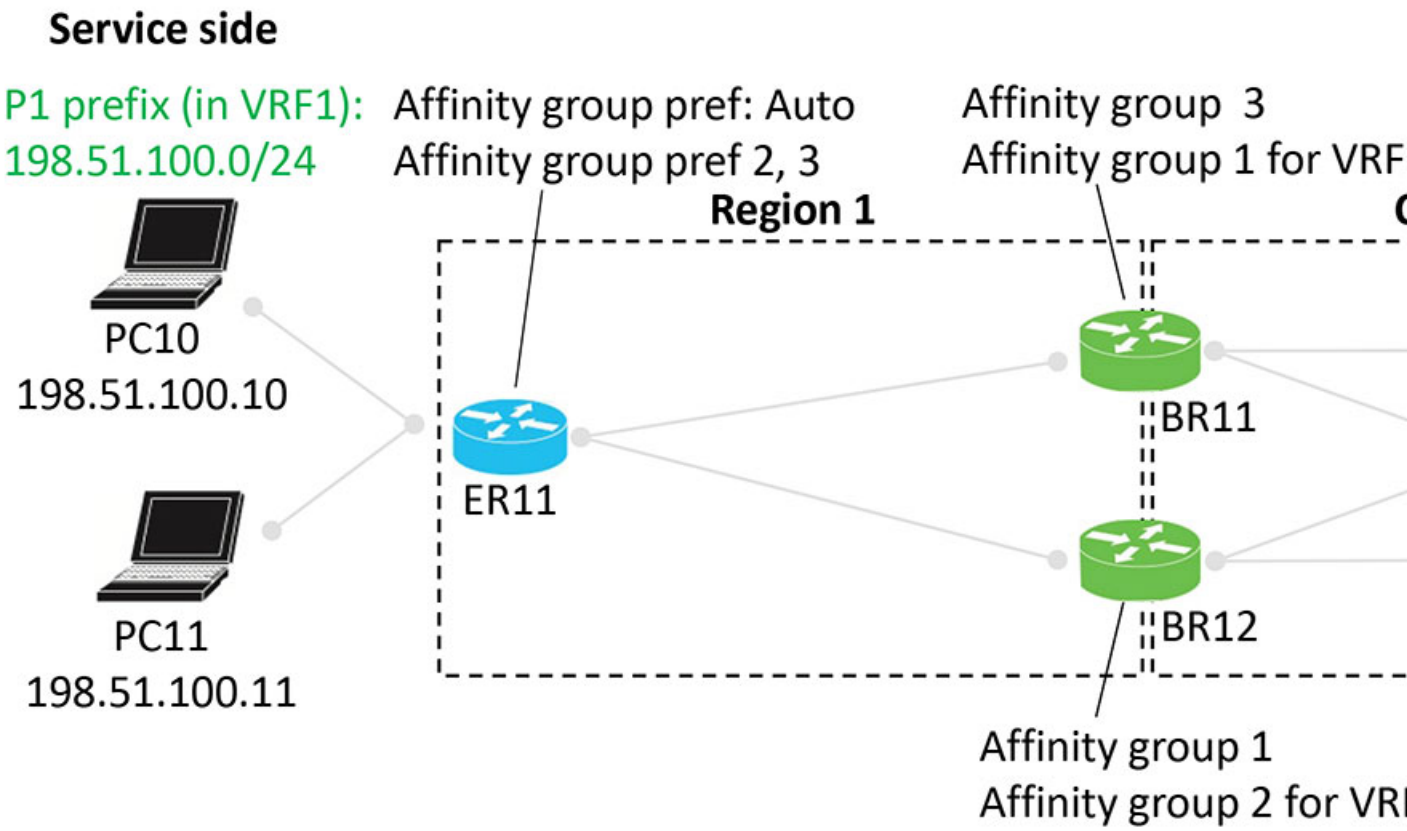
This comprehensive example describes how to configure border routers and edge routers in a Multi-Region Fabric (MRF) environment to achieve symmetric routing between PC devices behind ER11 (Region 1) and ER21 (Region 2).

The example focuses specifically on traffic between PC10 and PC20.

The step-by-step illustrations show how route re-origination and path preference ensure that traffic in both directions follows the same path across multiple hops.

Multi-region fabric scenario: configuration for symmetric routing

Figure 18: Multi-region fabric scenario, configuration for symmetric routing



Advertising P1 routes

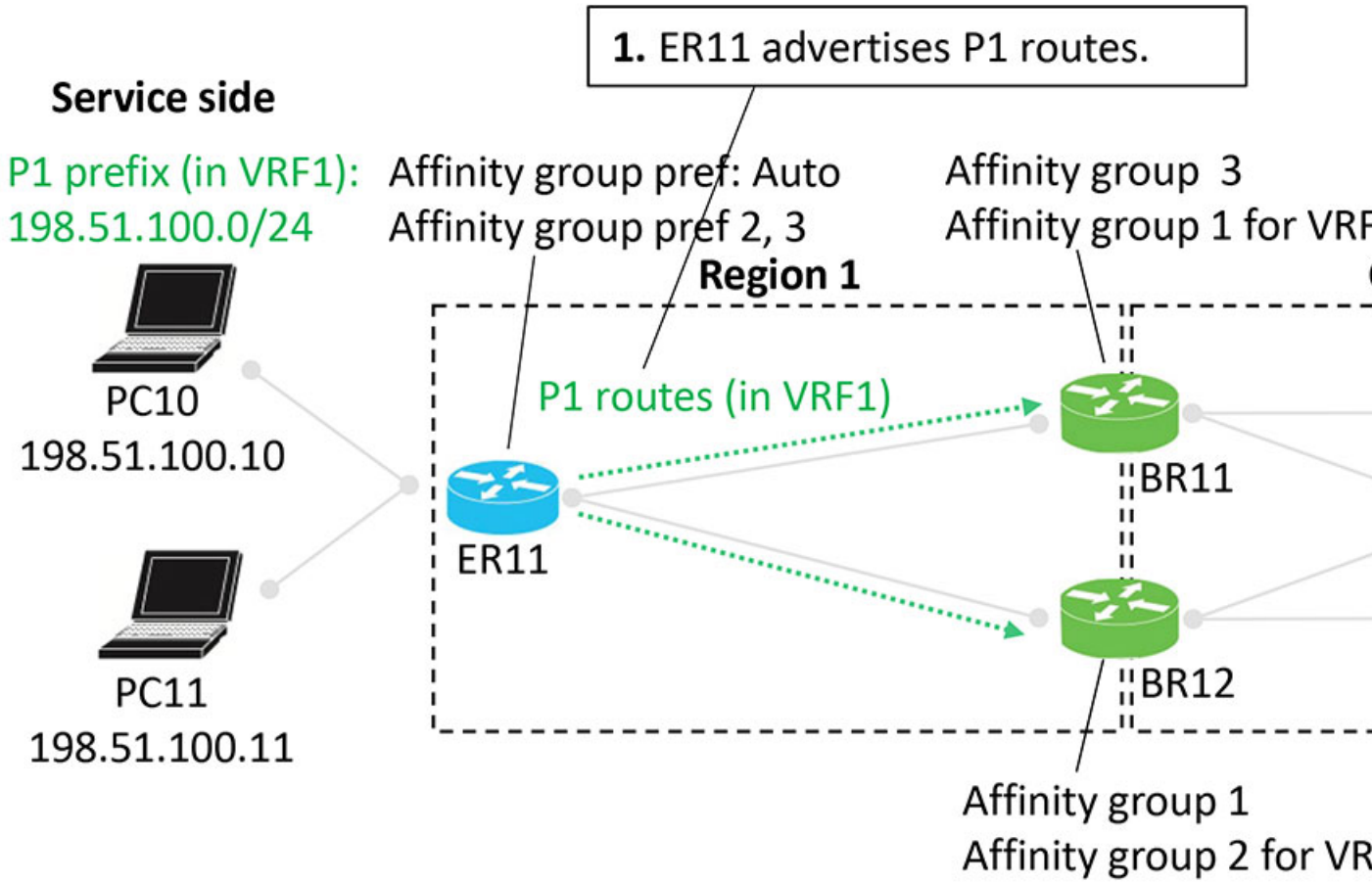
Edge router ER11 advertises P1 routes. These routes are re-originated as they move from Region 1 toward Region 2, passing through border routers that assign affinity groups and derived affinity groups (DAG).

Routers select preferred routes based on:

- Outside the core region: Affinity group preference
- Inside the core region: Lowest derived affinity group (dag) value

Figures:

Figure 19: Edge router ER11 advertises P1 routes



····▶	ER11 advertises P1 routes

Figure 20: Border routers BR11 and BR12 re-originate the P1 routes

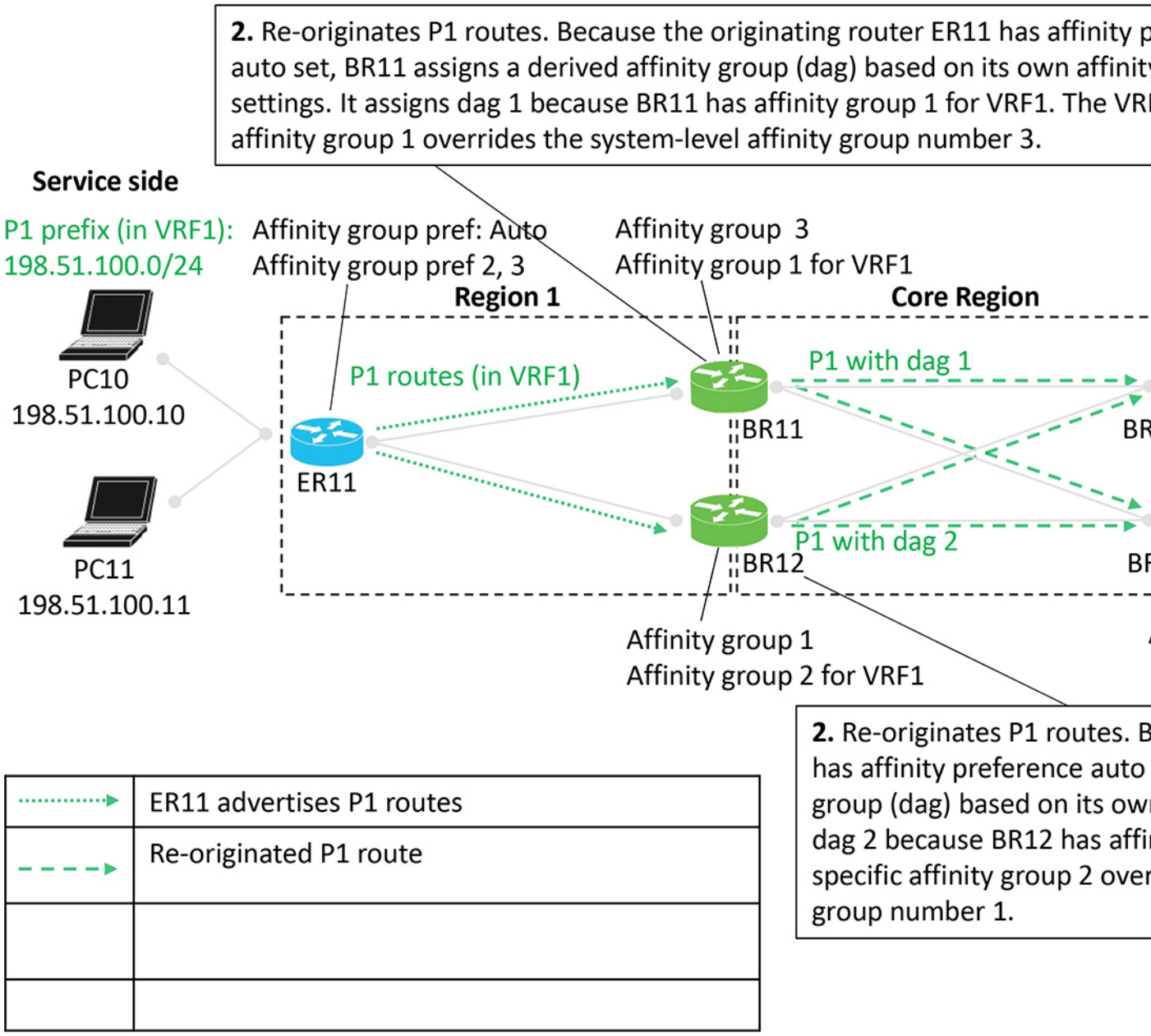
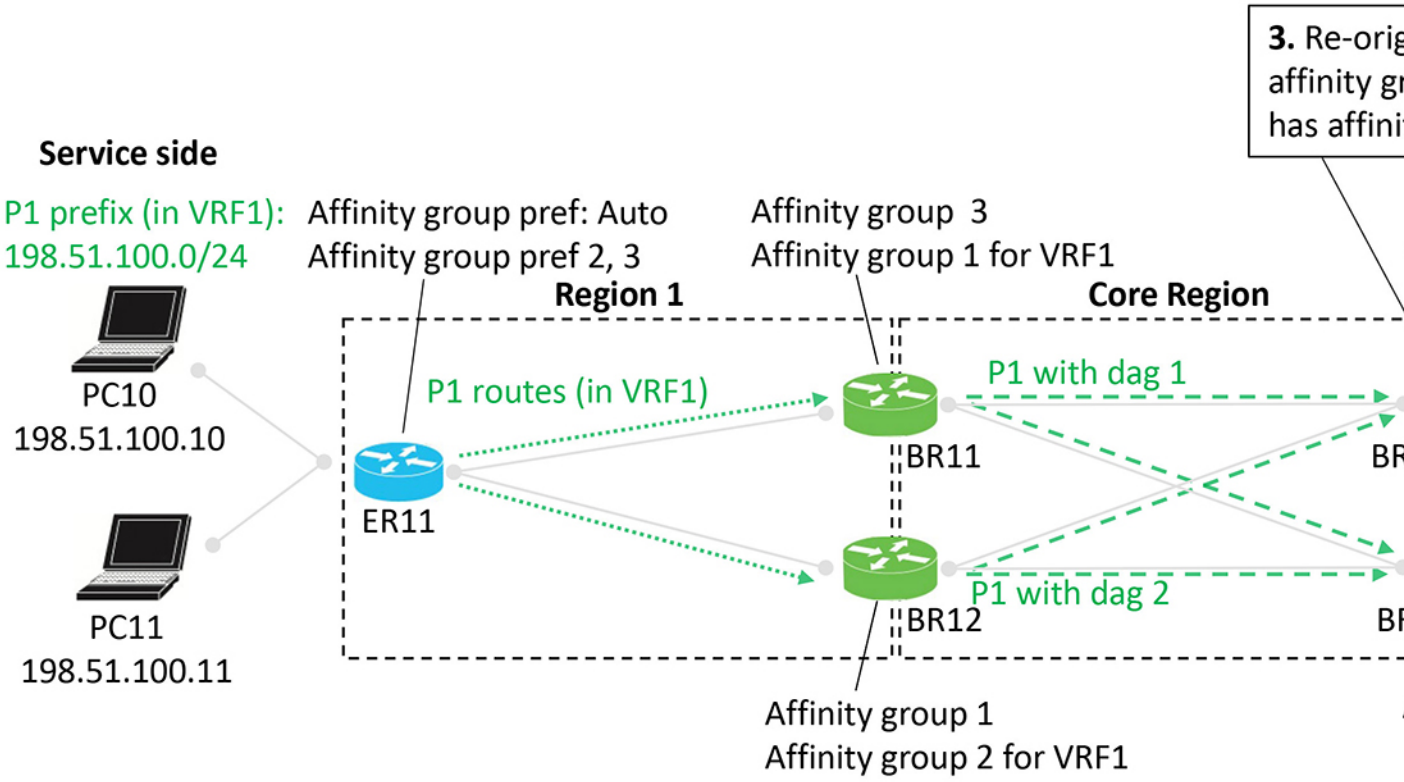
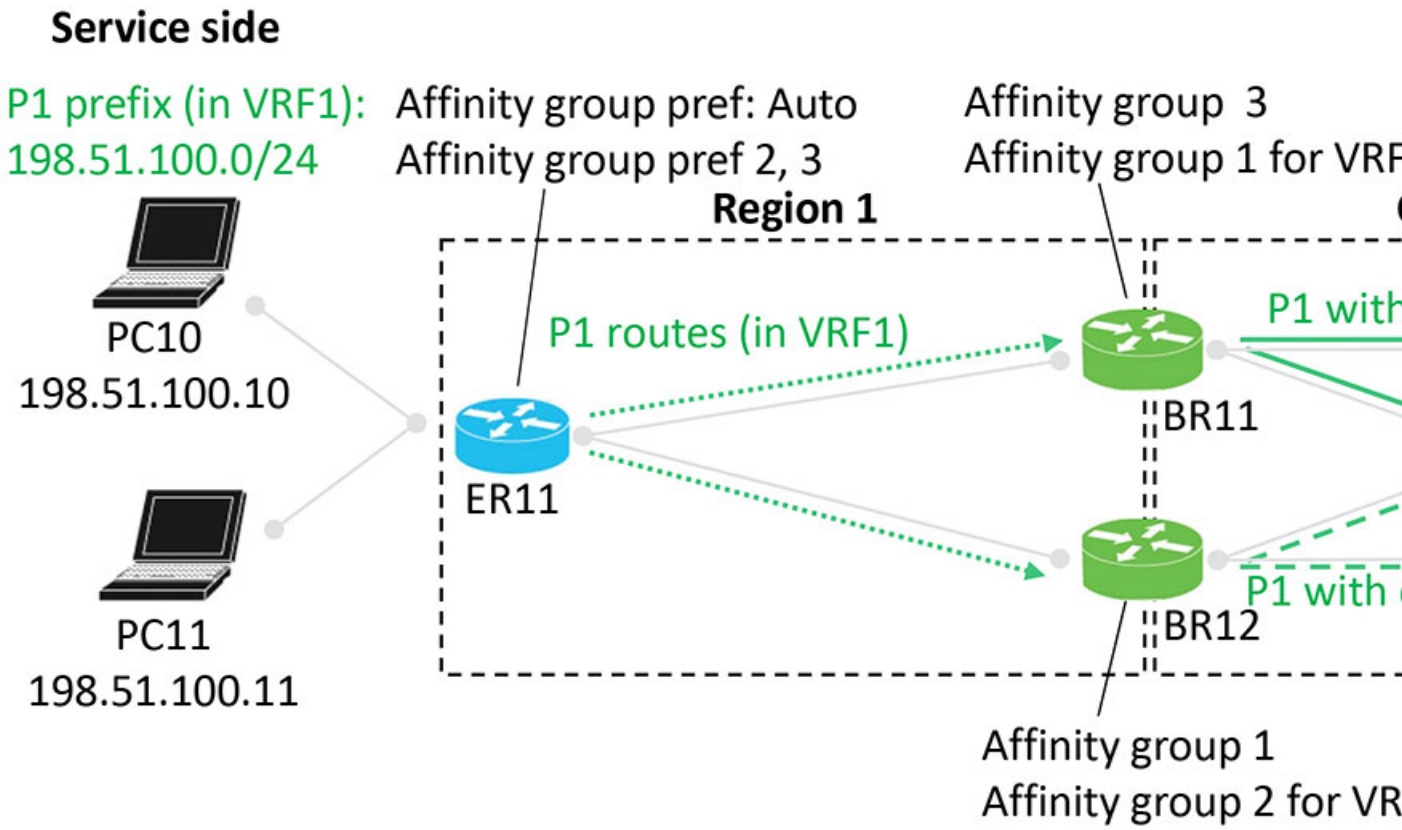


Figure 21: Border routers BR21 and BR22 re-originate the P1 routes



····▸	ER11 advertises P1 routes
- - -▸	Re-originated P1 route

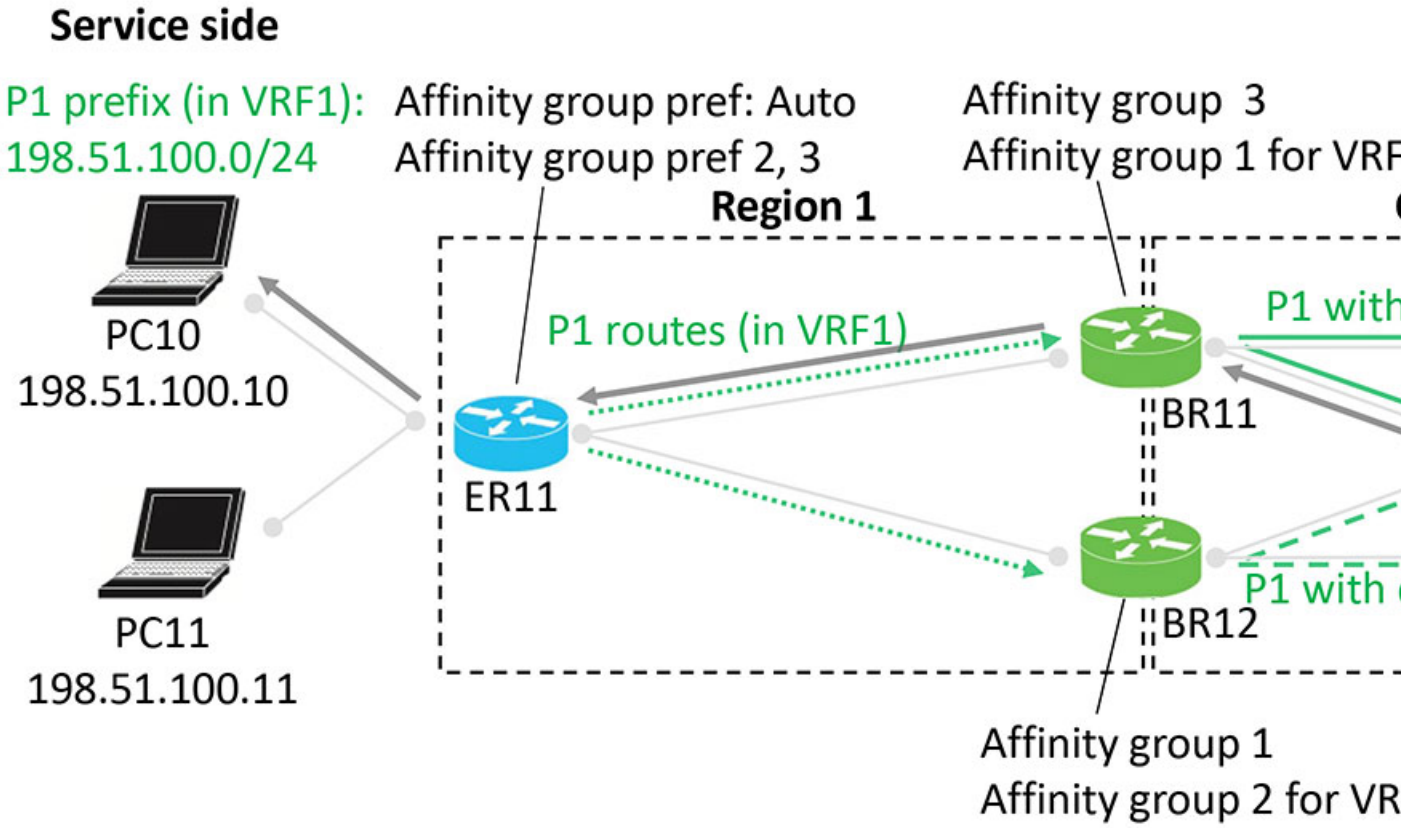
Figure 22: Route preference according to affinity group and derived affinity group



····▸	ER11 advertises P1 routes
- - - -▸	Re-originated P1 route **not preferred** by the receiving device due to higher ag or dag
——▸	Re-originated P1 route **preferred** by the receiving device receiving due to lower ag or dag

4. Rou
- Outsi
- In the
Solid a

Figure 23: Resulting path of traffic to P1



····▸	ER11 advertises P1 routes
- - - ▸	Re-originated P1 route **not preferred** by the receiving device due to higher ag or dag
——▸	Re-originated P1 route **preferred** by the receiving device receiving due to lower ag or dag
——▸	Resulting path between PC20 and PC10

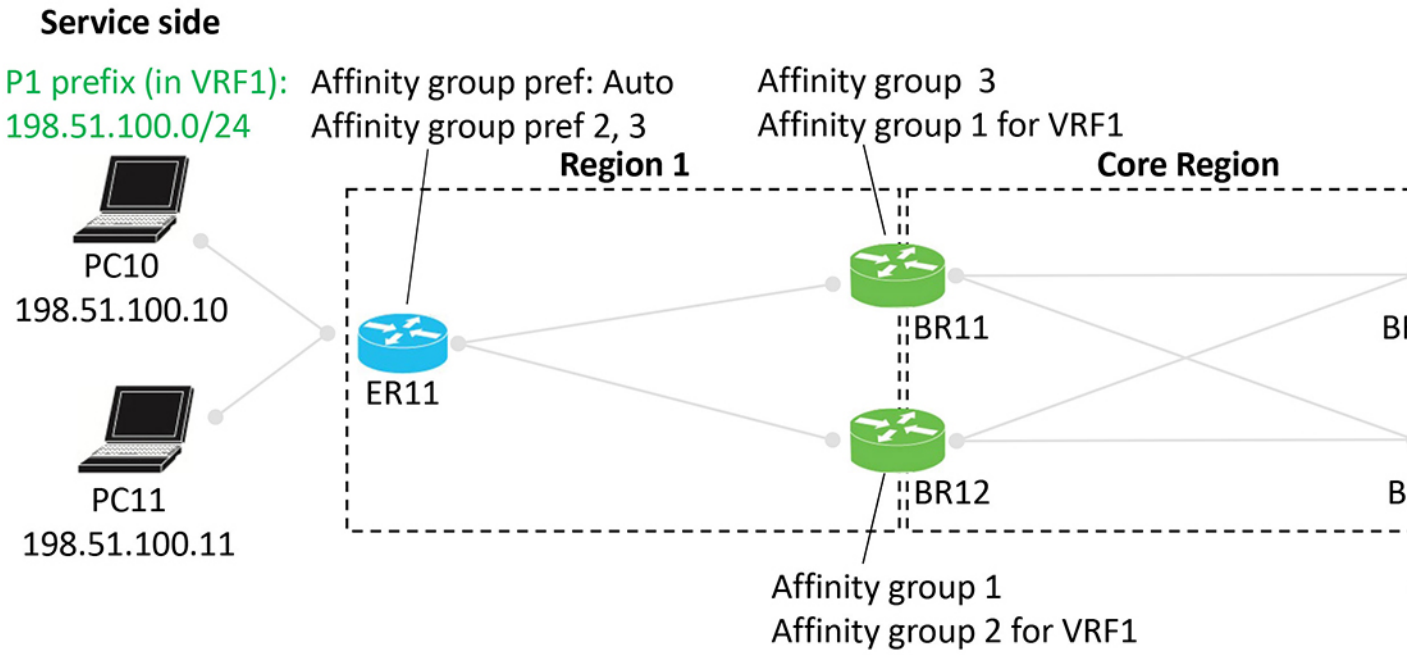
Advertising P2 routes

Edge routers ER21 and ER22 advertise P2 routes. These routes are re-originated from Region 2 back toward Region 1, with border routers again assigning affinity groups and derived affinity groups during the process.

Route preference is determined using the same rules:

- Outside the core region: Affinity group preference
- Inside the core region: Lowest derived affinity group (dag) value

Figure 24: Edge router ER21 advertises P2 routes



····▶	ER11 advertises P2 routes

Figure 25: Border routers BR21 and BR22 re-originate the P2 routes

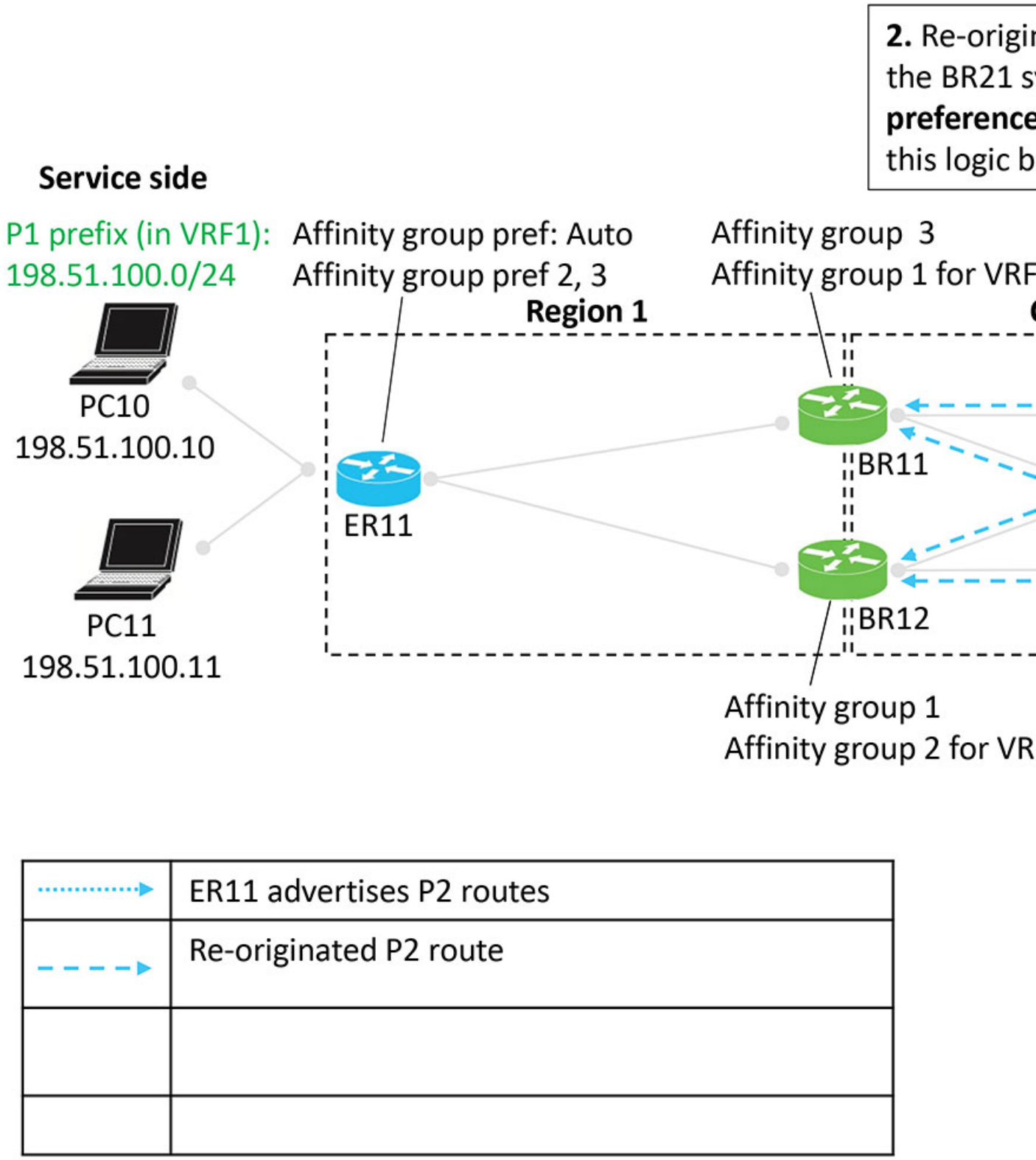


Figure 26: Border routers BR11 and BR12 re-originate the P2 routes

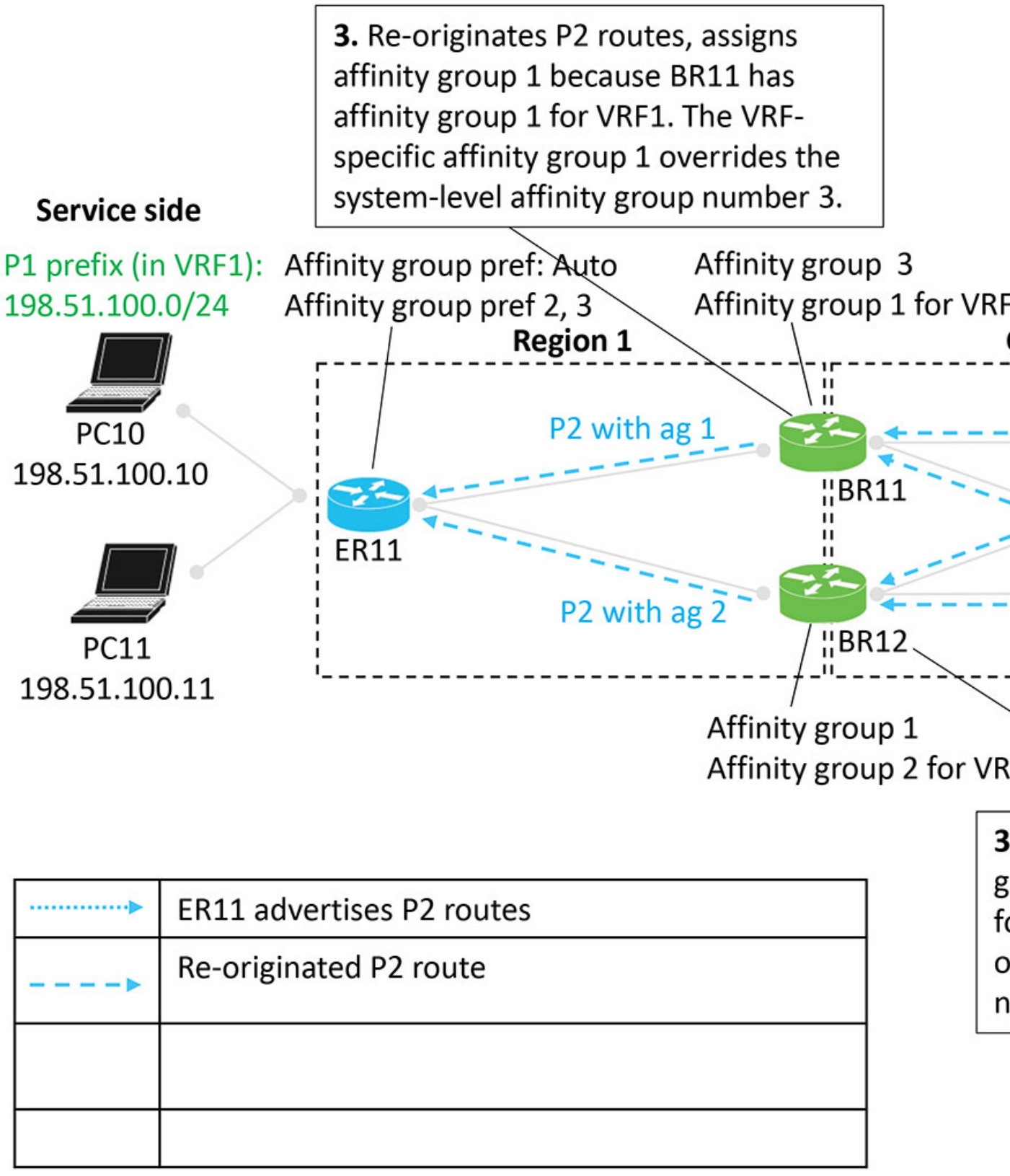


Figure 27: Route preference according to affinity group and derived affinity group

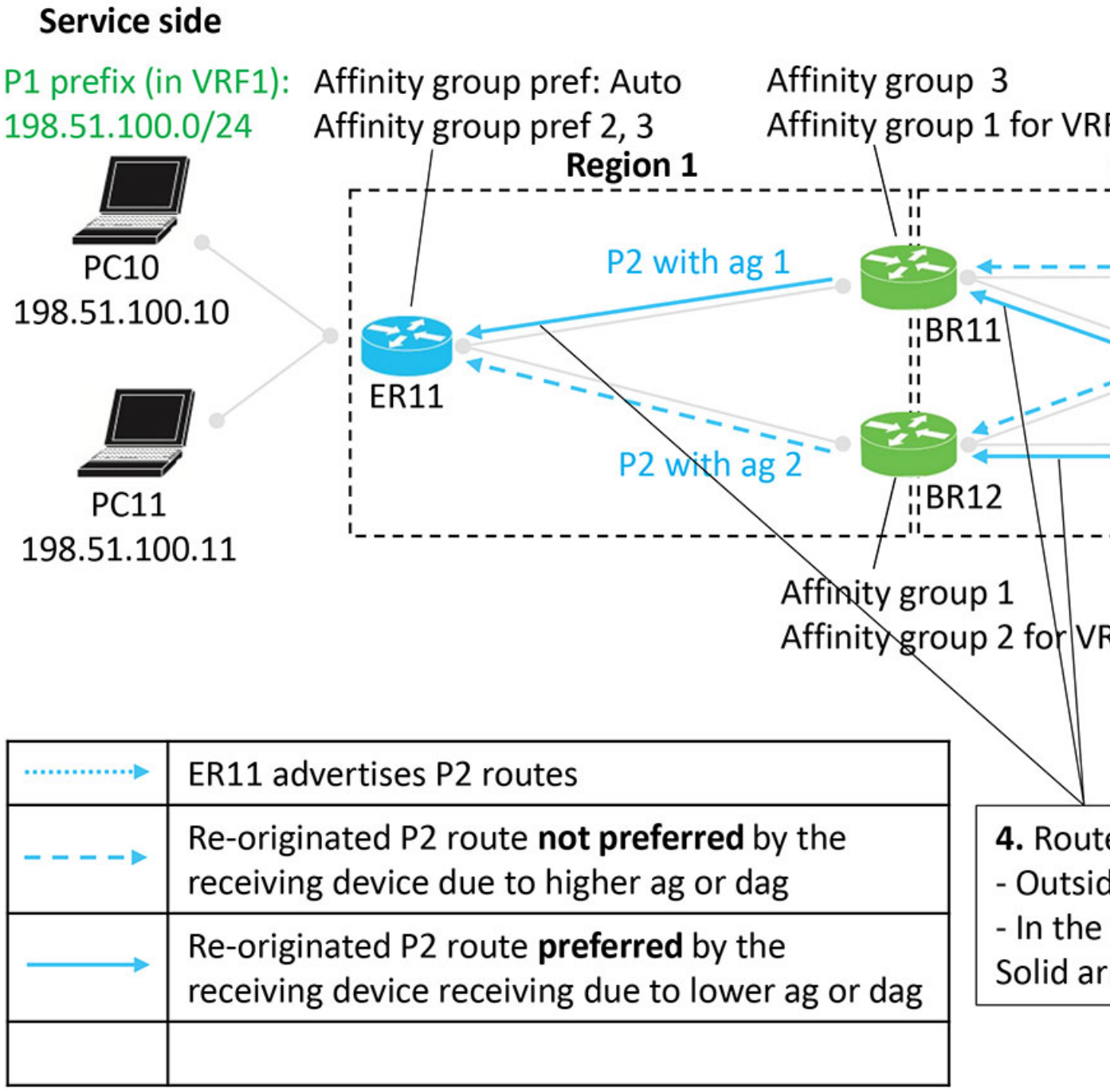
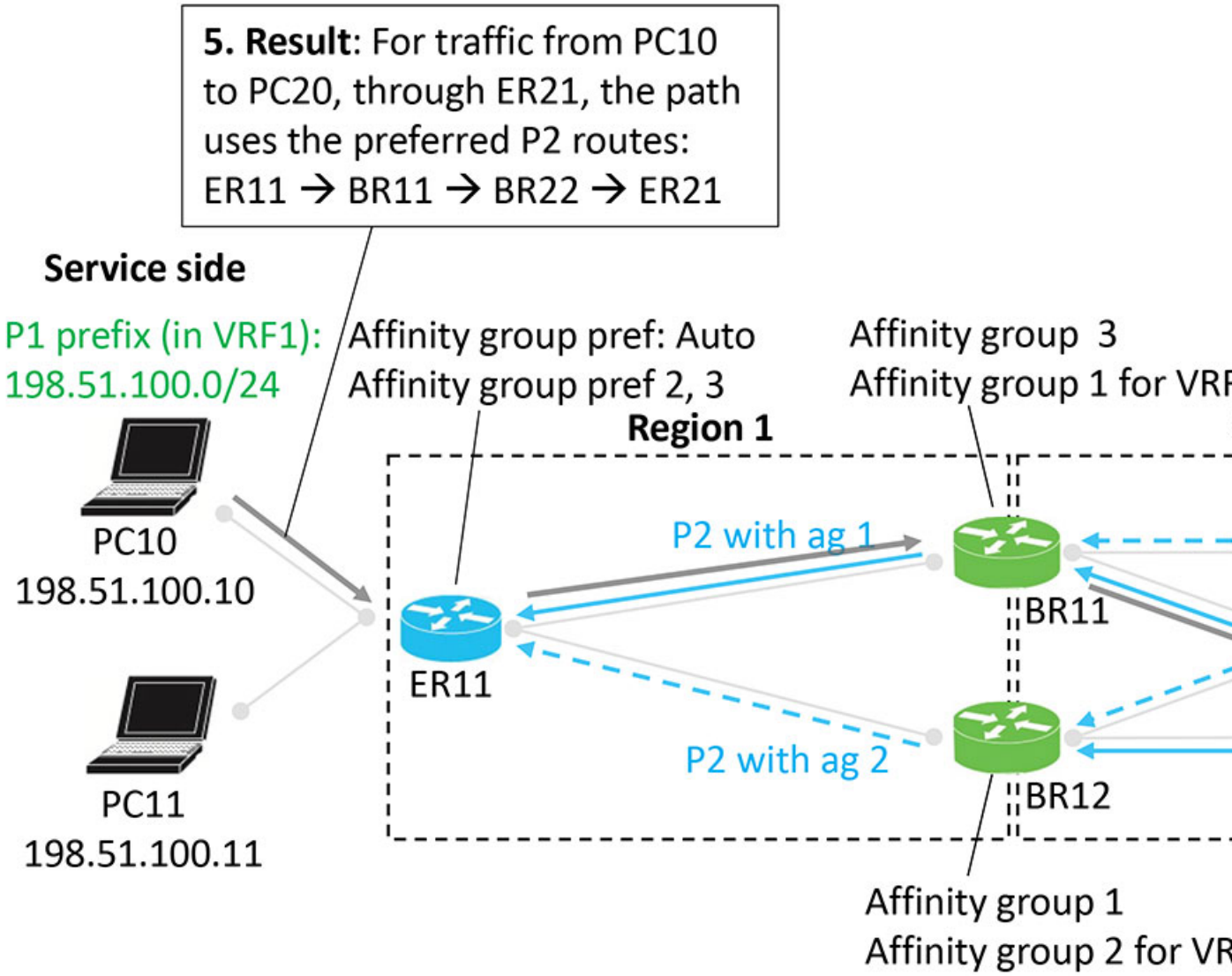


Figure 28: Resulting path of traffic to P2

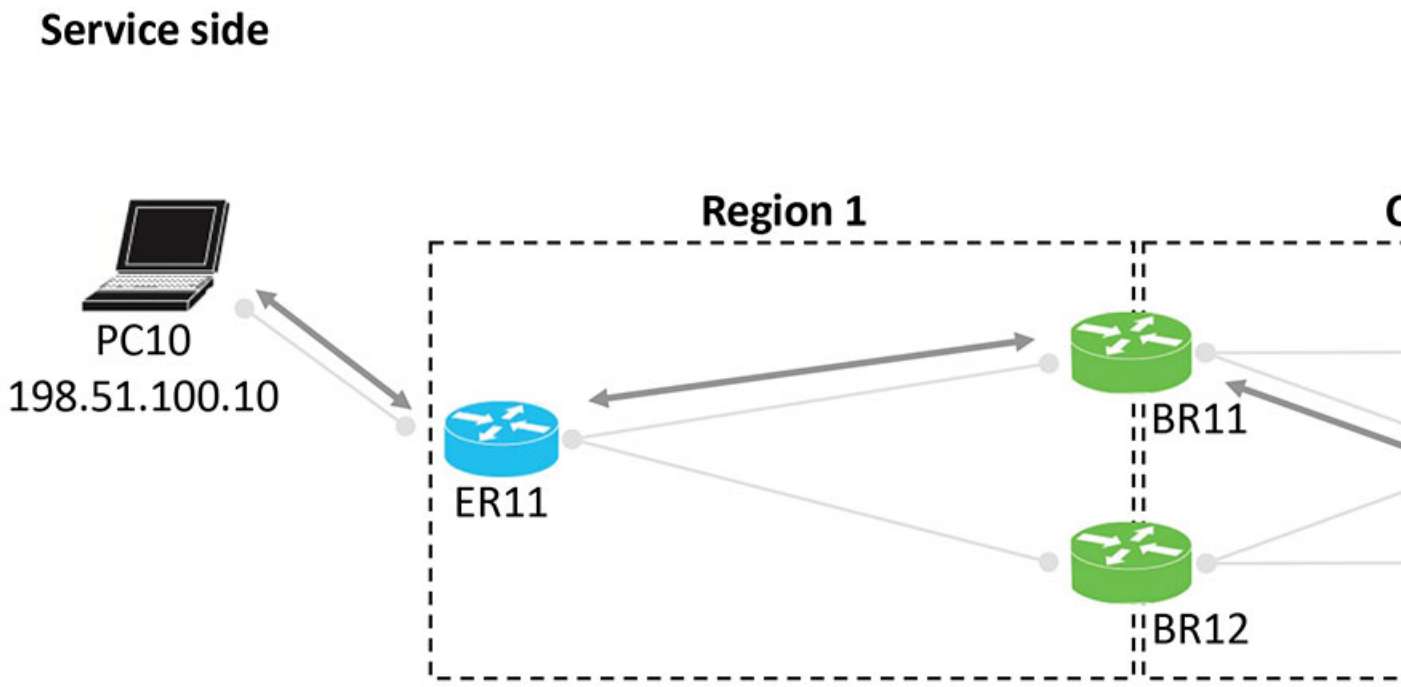


····▶	ER11 advertises P2 routes
- - -▶	Re-originated P2 route **not preferred** by the receiving device due to higher ag or dag
——▶	Re-originated P2 route **preferred** by the receiving device receiving due to lower ag or dag
——▶	Resulting path between PC10 and PC20

Result

The following figure shows that the result of the configuration is symmetric routing for flows between, in this example, PC10 and PC20:

Figure 29: Result is symmetric routing



Supported scenarios

Identify network architectures that support symmetric routing, including compatible topologies such as hub-and-spoke with active/active hubs and Multi-Region Fabric designs, to ensure symmetry requirements are met.

The symmetric routing configuration method described in this document applies to the following deployment scenarios:

- Hub-and-spoke topology with multiple hub routers: includes deployments where the hub router provides connectivity to a multi-homed data center.
- Multi-region fabric with multiple border routers: Covers scenarios where an MRF region contains a multi-homed data center, requiring consistent bidirectional path selection across regions.
- Multi-region fabric with transport gateways serving subregions: Applies to MRF deployments where transport gateways (TGWs) connect subregions and influence route propagation.

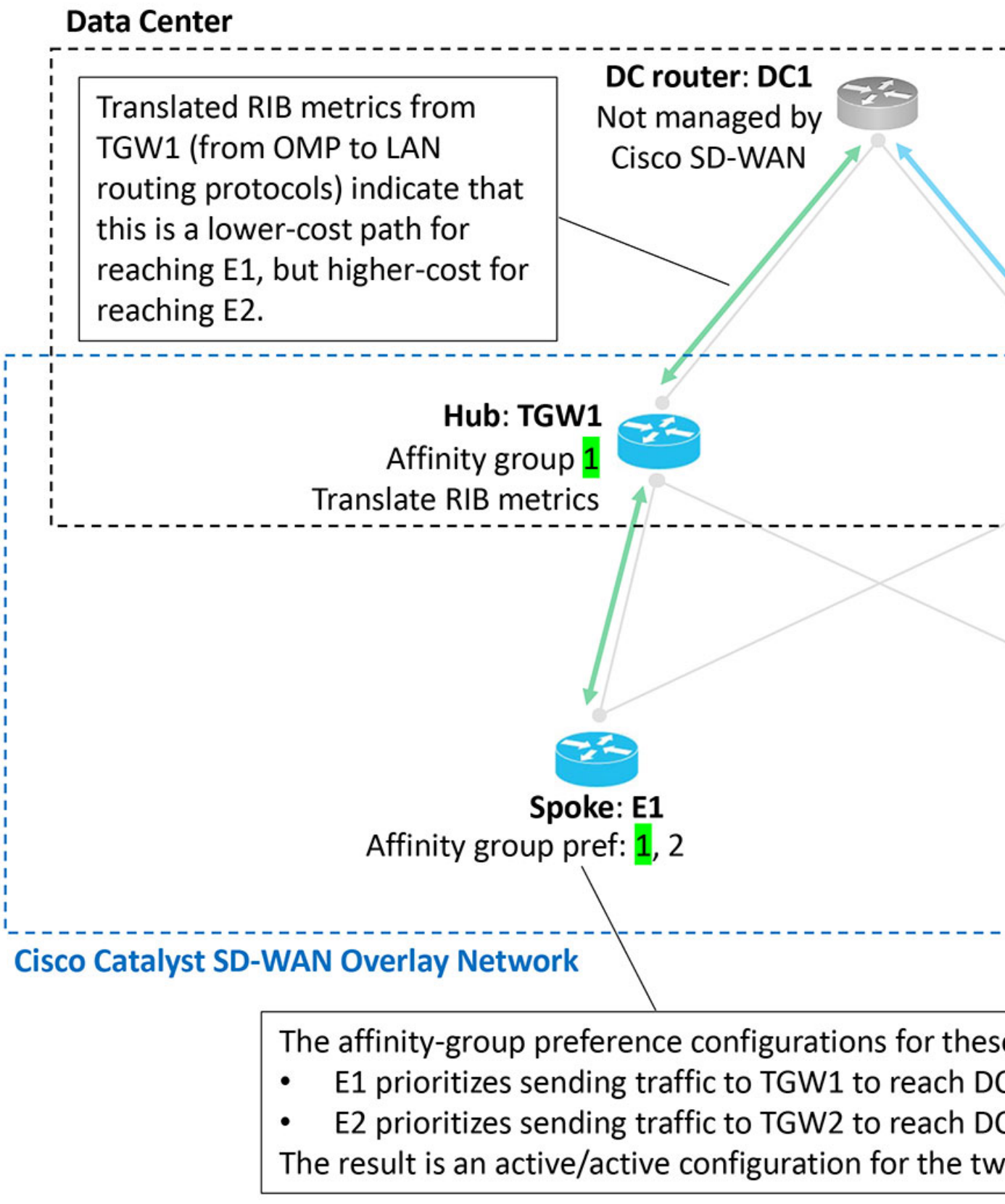
Scenario: Hub-and-spoke topology, multiple hubs serving a data center, active/active

In this scenario, two hubs serve a data center. The two hubs are both active, for an active/active arrangement.

The data center LAN is not part of the Cisco Catalyst SD-WAN overlay network.

For information about the redistribute omp translate-rib-metric command shown in the illustration see Configure a Router to Translate OMP Metrics to BGP or OSPF Using a CLI Template.

Figure 30: Data center, two hubs, active/active

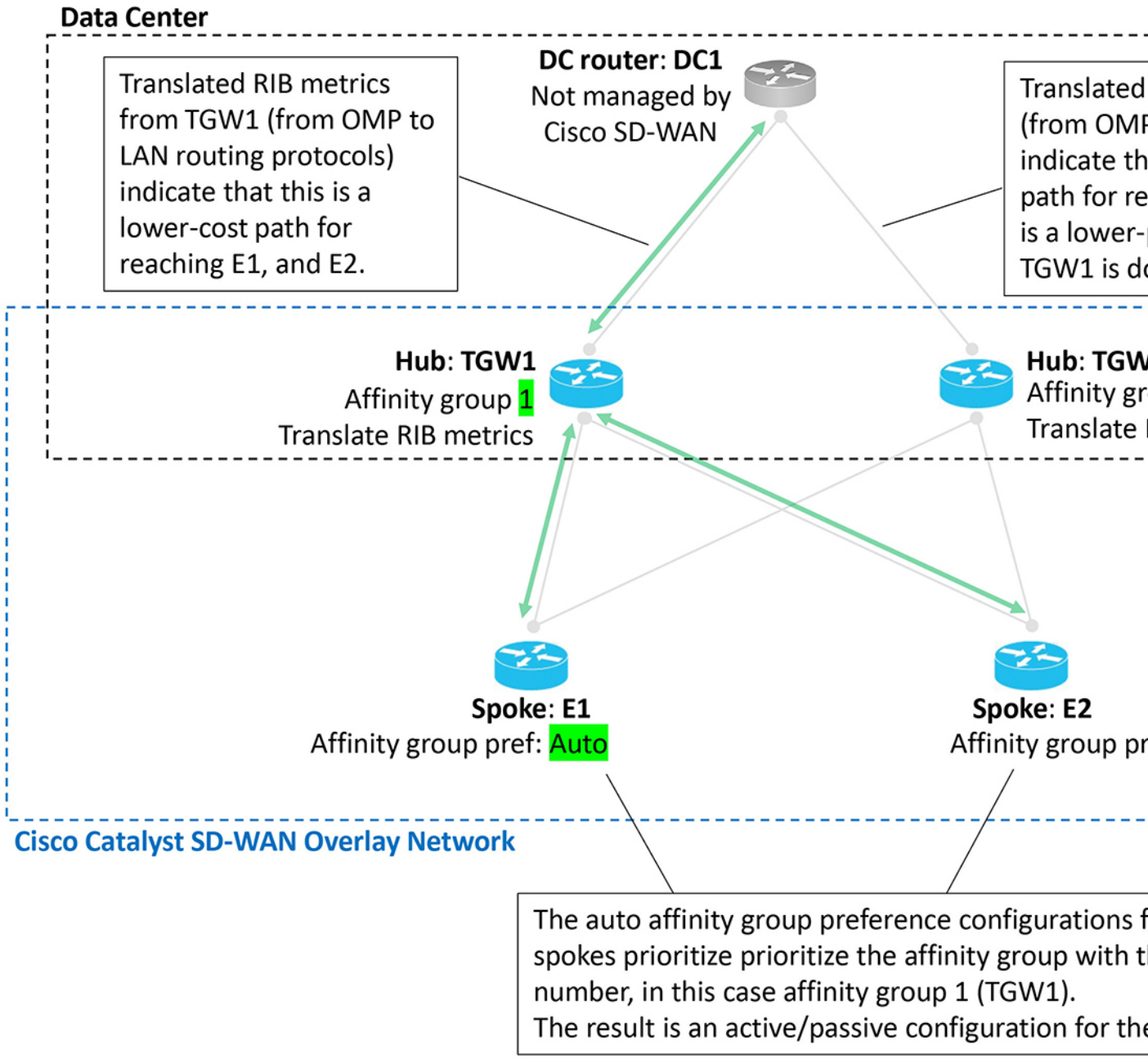

Data Center
Translated RIB metrics from TGW1 (from OMP to LAN routing protocols) indicate that this is a lower-cost path for reaching E1, but higher-cost for reaching E2.
DC router: DC1
Not managed by Cisco SD-WAN
Hub: TGW1
Affinity group 1
Translate RIB metrics
Spoke: E1
Affinity group pref: 1, 2
Cisco Catalyst SD-WAN Overlay Network
The affinity-group preference configurations for these
• E1 prioritizes sending traffic to TGW1 to reach DC
• E2 prioritizes sending traffic to TGW2 to reach DC
The result is an active/active configuration for the tw

Scenario: Hub-and-spoke topology, multiple hubs serving a data center, active/passive

In this scenario, two hubs serve a data center. Only one hub is typically active, and the other is stand-by, in case the active hub becomes unavailable. This is an active/passive arrangement.

The data center LAN is not part of the Cisco Catalyst SD-WAN overlay network.

Figure 31: Data center, two hubs, active/passive

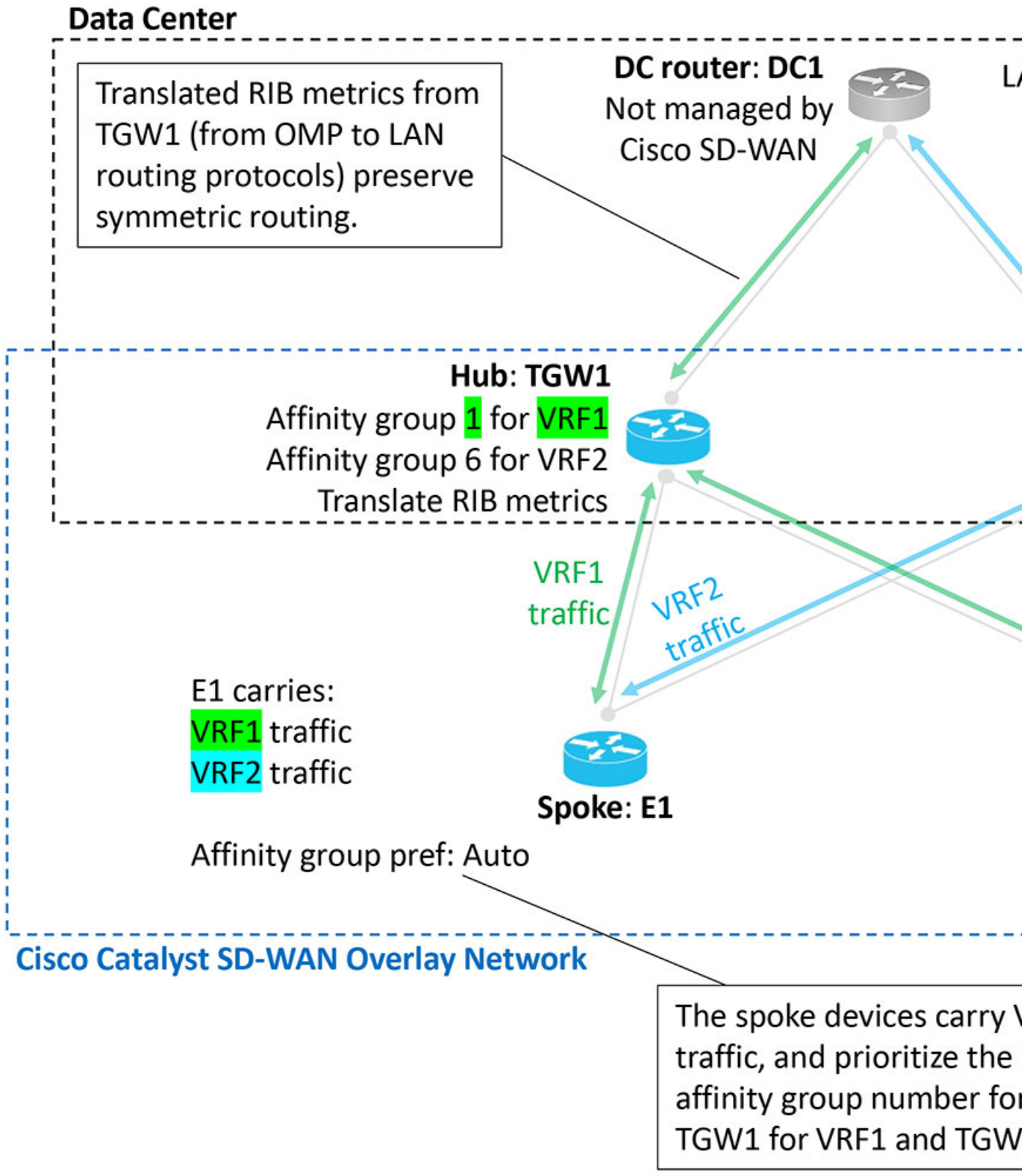


Scenario: Hub-and-spoke topology, multiple hubs serving a data center, active/active by VRF

In this scenario, two hubs serve a data center. The two hubs are both active, for traffic in one of the two VRFs. This is an active/active arrangement, segregated by VRF. The hub TGW1 is active for VRF1 and the hub TGW2 is active for VRF2. Both hubs can operate as stand-by for the other VRF.

The data center LAN is not part of the Cisco Catalyst SD-WAN overlay network.

Figure 32: Data center, two hubs, active/active, segregated by VRF



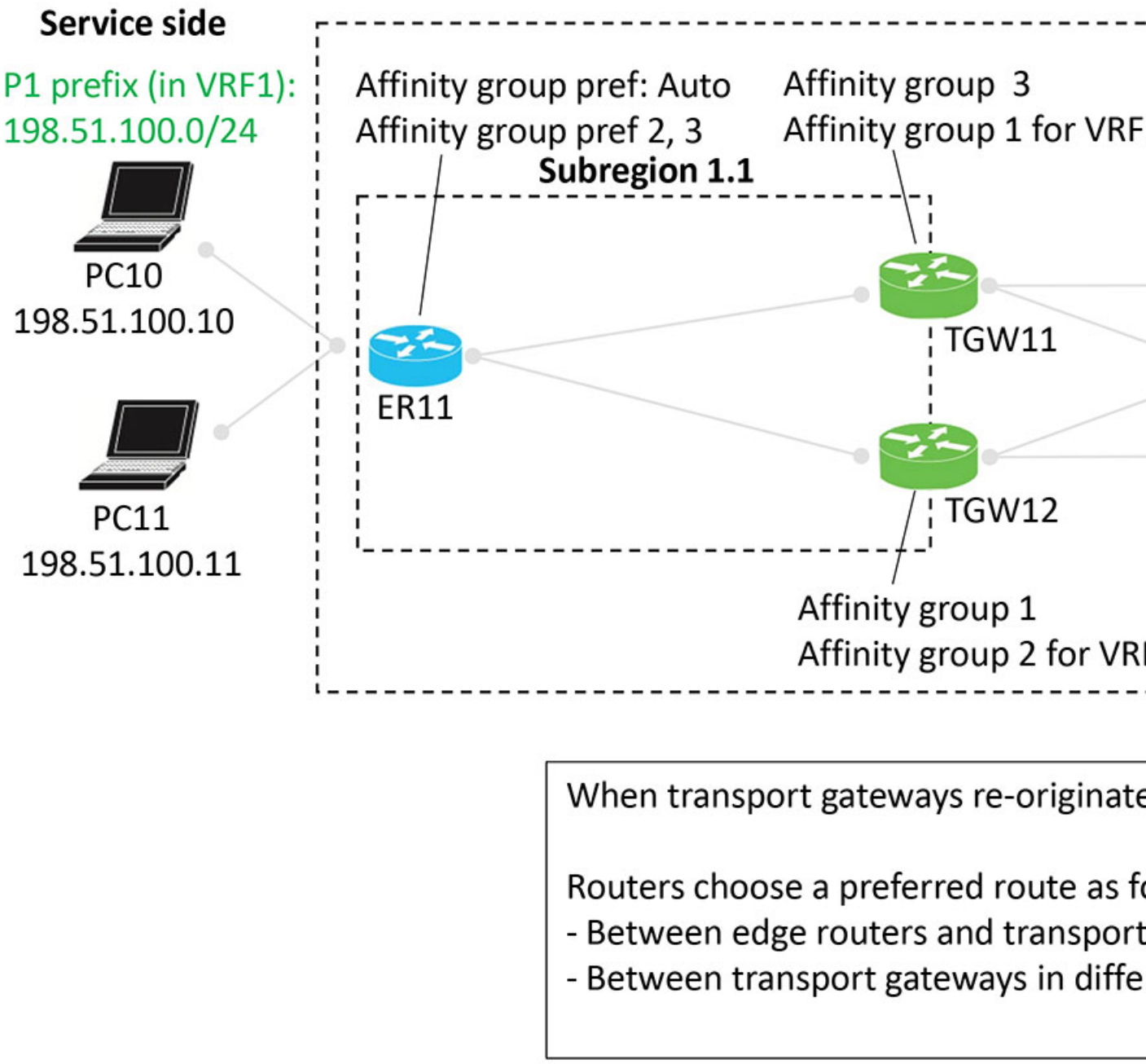
Scenario: Multi-region fabric, transport gateways serving subregions

A Multi-Region Fabric scenario in which transport gateways serve two subregions closely resembles the comprehensive example described in Example of Configuration for Symmetric Routing and the Mechanism.

Similarly to the border routers in the comprehensive example, transport gateways assign a derived affinity group (dag) to routes that they re-originate to other transport gateways. As described in the illustration:

- When transport gateways re-originate routes, they assign derived affinity group (dag) values to the routes.
- Routers choose a preferred route as follows:
  - Between edge routers and transport gateways: According to affinity group preference
  - Between transport gateways in different subregions: According to the lowest derived affinity group value

Figure 33: Multi-region fabric with transport gateways serving subregions



Scenario: Multi-region fabric with route leaking

A Multi-Region Fabric scenario in which transport gateways serve two subregions, with route leaking, closely resembles the comprehensive example described in Example of Configuration for Symmetric Routing and the Mechanism.

Similarly to the border routers in the comprehensive example, transport gateways assign a derived affinity group (dag) to routes that they re-originate to other transport gateways. This scenario is similar to the one described in Scenario: Multi-Region Fabric, Transport Gateways Serving Subregions, but with route leaking. As described in the illustration:

When transport gateways re-originate routes, they assign derived affinity group (DAG) values to those routes.

Routers select a preferred route in the following ways:

- Between edge routers and transport gateways: They use affinity group preference.

- Between transport gateways in different subregions: They choose the route with the lowest derived affinity group value.

In this scenario, a control policy on the Cisco SD-WAN Controllers leaks routes from VRF1 to VRF2 and from VRF2 to VRF1. This route leaking enables endpoints in different VRFs to communicate.

This route-leaking scenario clearly shows how transport gateways (or border routers) assign a derived affinity group (DAG) when they re-originate routes. The logic works subtly, but this example highlights it well.

Default behavior

In this example, the edge routers and transport gateway routers operate as follows:

- ER11 subscribes only to VRF1 and advertises prefix P1 in VRF1.
- ER21 subscribes only to VRF2 and advertises prefix P2 in VRF2.

All transport gateway routers handle traffic for both VRF1 and VRF2, so they re-originate both P1 (in VRF1) and P2 (in VRF2).

By default, the network enforces VRF isolation. When a device advertises routes in different VRFs, the Cisco SD-WAN Controllers filter those routes before sending them to other devices. A controller advertises a VRF x route only to devices that subscribe to VRF x.

Therefore, in this example:

- ER11, which subscribes only to VRF1, does not receive P2 routes from VRF2.
- ER21, which subscribes only to VRF2, does not receive P1 routes from VRF1.

As a result, VRF isolation blocks traffic between ER11 and ER21 because each router subscribes exclusively to a different VRF.

Route leaking

Route leaking allows devices to advertise routes across VRFs by exporting (“leaking”) a route from one VRF into another.

- Source VRF: the route’s original VRF
- Current VRF: the VRF into which the route was exported

When routers advertise exported routes, they track both the source VRF and the current VRF, preserving the background of each route. This tracking becomes important in the logic described later.

In this example, the following route-leaking policies apply:

- An inbound control policy for ER11 instructs it to receive VRF1 routes and export them into VRF2.

  Result: ER11 advertises prefix P1 in both VRF1 and VRF2 to its transport gateways, TGW11 and TGW12.

- An inbound control policy for ER21 instructs it to receive VRF2 routes and export them into VRF1.

  Result: ER21 advertises prefix P2 in both VRF2 and VRF1 to its transport gateways, TGW21 and TGW22.

As mentioned earlier, after leaking routes, devices continue to track each route’s source VRF and current VRF.

Calculating the derived affinity group (DAG)

A transport gateway device or a border router in a similar scenario assigns a derived affinity group (DAG) to any route it re-originates using the following logic:

1. If the originating router is configured with affinity group preference auto (see ER11 in the example), then the re-originating device (for example, TGW11) determines the dag according to its own (TGW11's) affinity group configuration, as follows:

a. For the leaked route, consider its source VRF and current VRF. Choose the numerically lower of the two values. Call this *x*.

b. Do one of the following:

- If the re-originating device only has a system-level affinity group, not VRF-specific affinity groups, then:

  Use the system-level affinity group number for the dag. Assign a dag of that number when re-originating the route.

- If the re-originating device has a VRF-specific affinity group configured for VRF *x* described in step **a**, then:

  Use this VRF-specific affinity group number for the dag. Assign a dag of this number when re-originating the route.

2. If the originating router **is not** configured with affinity group preference auto (see ER21 in the example), then the re-originating device (for example, TGW21) must consider the affinity preference order configured on the originating device when determining the dag for re-originated routes, as follows:

a. For the leaked route, consider its source VRF and current VRF. Choose the numerically lower of the two values. Call this *x*.

b. Do one of the following

- If the re-originating device only has a system-level affinity group, not VRF-specific affinity groups, then:

  Check the affinity group preference order of the originating device (see ER21). Determine the item number of where the system-level affinity group number occurs in the preference order (item 1, 2, 3, and so on, in the preference order list). Assign a dag of this item number when re-originating the route.

  In the example of TGW21 and ER21, determine where affinity group 2 occurs in the preference order of ER21, which is (1, 2). It is item 2 in the list. So assign a dag of 2 when re-originating the route.

- If the re-originating device has a VRF-specific affinity group configured for VRF *x* described in step **a**, then:

  Using this VRF-specific affinity group, check the affinity group preference order of the originating device. Determine the item number of where the VRF-specific affinity group number occurs in the preference order (item 1, 2, 3, and so on, in the preference order list). Assign a dag of this item number when re-originating the route.

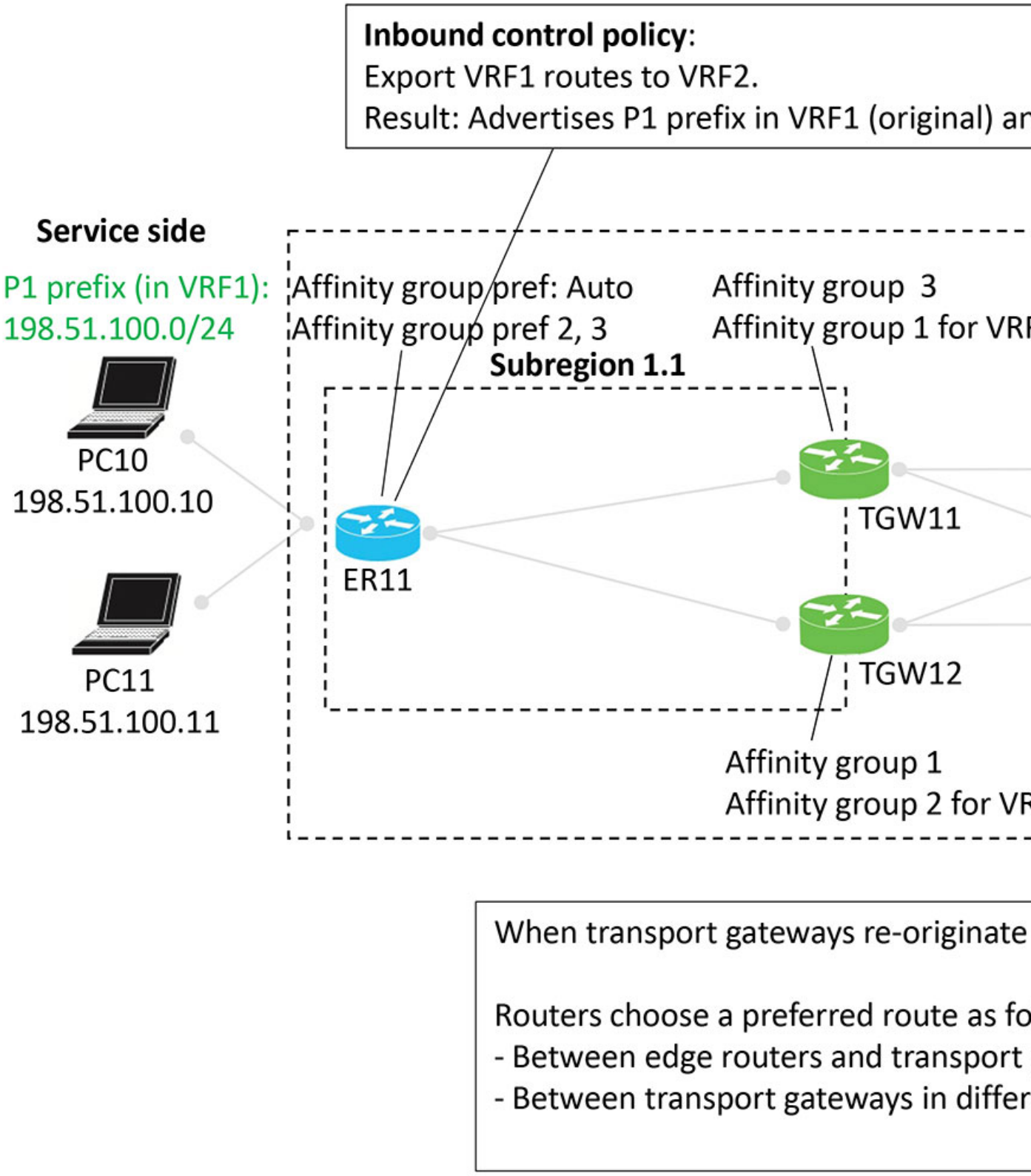
  Hypothetically, in the example, if TGW21, in addition to having a system-level affinity group of 2, also had a VRF-specific affinity group of 1 for VRF1, then when TGW21 received from ER21 a P2 route leaked to VRF1, it would consider the preference order of the originating device (ER21). In this hypothetical example with a VRF-specific affinity group of 1, for a route received from ER21, it would check where affinity group 1 occurs in the preference order of ER21, which is (1, 2). It is item 1 in the list. So TGW2 would assign a dag of 1 when re-originating the route.

Example

In the scenario shown in the illustration, a route leaked from VRF2 to VRF1 has a source VRF value of 2 and a current VRF value of 1. When a transport gateway re-originates this route, it assigns a DAG based on the number 1, which is the lower of the two VRF numbers. For example, if TGW12 re-originates a route with a source VRF value of 1 and a current VRF value of 2, it chooses 1 because it is the lower of the two VRF numbers. It therefore calculates the DAG according to VRF1. TGW12 has a system-level affinity group of 1 and a VRF-specific affinity group of 2 for VRF1. Since it calculates the DAG according to VRF1, it assigns the re-originated route a DAG value of 2, taken from the VRF-specific affinity group.

In a hypothetical scenario, if TGW12 had a system-level affinity group of 5 and a VRF1-specific affinity group of 7, then for a route with source VRF 1 and current VRF 2, TGW12 would assign a DAG of 7, taken from the VRF-specific affinity group of 7 for VRF1.

Figure 34: Multi-region fabric with subregions, route leaking


Inbound control policy:
Export VRF1 routes to VRF2.
Result: Advertises P1 prefix in VRF1 (original) ar
Service side
P1 prefix (in VRF1):
198.51.100.0/24
Affinity group pref: Auto
Affinity group pref 2, 3
Affinity group 3
Affinity group 1 for VR
Subregion 1.1
PC10
198.51.100.10
PC11
198.51.100.11
ER11
TGW11
TGW12
Affinity group 1
Affinity group 2 for VR
When transport gateways re-originate
Routers choose a preferred route as fo
- Between edge routers and transport
- Between transport gateways in differ

Configure symmetric routing

Follow these procedures to implement symmetric routing on your devices.

Use these procedures to configure symmetric routing.

- Configure automatic affinity group preference
- Configure router affinity groups for specific VRFs
- Configure a router to translate OMP metrics to BGP or OSPF using CLI commands on page 287
- Configure an affinity group on a router
- Configure Affinity Group Preference on a Router

Configure a router to use automatic affinity group preference

Use any one of these methods to configure a router to use automatic affinity group preference.

- Feature template
- CLI commands

Configure a router to use automatic affinity group preference using templates

If you configure both a manual affinity preference order and an auto preference order on a router, the router gives priority to the auto preference order when selecting the next hop.

However, the manually configured preference list is still useful for path filtering using the **filter route outbound affinity-group preference** command. For information about filtering out paths for routers that are not on the device's affinity list, see Information About Router Affinity Groups and see the filter route outbound affinity-group preference command reference in the Cisco IOS XE SD-WAN Qualified Command Reference.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **ConfigurationTemplates**.
2. Click **Feature Templates**.
3. Do one of the following:
   - To create a System template for a device, click **Add Template**, choose a device type, and click **Cisco System**.
   - To edit an existing System template, locate a System template in the table of existing feature templates, click **...** adjacent to the template, and choose **Edit**.
4. In the **Affinity Group Preference Auto** field, choose **On**.
5. Click **Save** if creating a new template, or **Update** if editing an existing template.

Configure a router to use automatic affinity group preference, using CLI commands

If you configure a router with both **affinity-group preference-auto** and **affinity-group preference list**, the **affinity-group preference-auto** command has priority for selecting a next hop.

However, the **affinity-group preference list** command is still useful for path filtering using the **filter route outbound affinity-group preference** command.

For information about filtering out paths for routers that are not on the device's affinity list, see Information About Router Affinity Groups and see the filter route outbound affinity-group preference command reference in the Cisco IOS XE SD-WAN Qualified Command Reference.

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates. By default, CLI templates execute commands in global configuration mode.

Procedure

1. Enter system configuration mode.

```
system
```

2. Configure automatic affinity group preference.

```
affinity-group preference-auto
```

Configure router affinity groups for specific VRFs

Use any one of these methods configure router affinity groups for specific VRFs

- Feature template
- CLI commands

Configure router affinity groups for specific VRFs using templates

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates. By default, CLI templates execute commands in global configuration mode.

Procedure

1. Enter system configuration mode.

```
system
```

2. Configure an affinity group to apply to a specific VRF or range of VRFs.

```
affinity-per-vrf affinity-group vrf-range vrf-range
```

The following example configures affinity group 1 for VRF1:

```
system
  affinity-per-vrf 1 vrf-range 1
```

The following example configures affinity group 4 for the VRF range 3 to 6:

```
system
  affinity-per-vrf 4 vrf-range 3-6
```

Configure router affinity groups for specific VRFs using CLI commands

Use these steps to configure router affinity groups for specific VRFs.

Procedure

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates**.
3. Do one of the following:
   - To create a System template for a device, click **Add Template**, choose a device type, and click **Cisco System**.
   - To edit an existing System template, locate a System template in the table of existing feature templates, click **...** adjacent to the template, and choose **Edit**.
4. For Affinity Group Number for VRFs, there are two fields. In the left field, enter an affinity group number. In the right field, enter a VRF number or a range of numbers-for example, 2-4. To configure addition group numbers for specific VRFs, click the plus button.

   In Cisco SD-WAN Manager, you can configure up to four ranges. If you need to configure more, you can use a CLI template or CLI add-on template. See Configure Router Affinity Groups for Specific VRFs Using a CLI Template.
5. Click **Save** if creating a new template, or **Update** if editing an existing template.

Configure a router to translate OMP metrics to BGP or OSPF using CLI commands

Provides instructions for configuring a router to translate OMP route metrics to BGP, OSPF, or OSPFv3 using CLI commands, including mode selection, metric translation, AS-Path propagation for BGP, and practical examples for each protocol scenario.

This configuration is not available through a feature template.

For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates. By default, CLI templates execute commands in global configuration mode.

Procedure

1. Do one of the following.
   a) If the underlay network uses the border gateway protocol (BGP), enter router configuration mode and specify a BGP autonomous system. For information about the BGP autonomous system parameter, see the IP Routing Configuration Guide, Cisco IOS XE 17.x.

   ```
   router bgp bgp-AS
   ```

   b) If the underlay network uses the open shortest path first (OSPF) protocol, enter router configuration mode and specify an OSPF.

   ```
   router ospf process-id [vrf vrf-name]
   ```

   c) If the underlay network uses the open shortest path first version 3 (OSPFv3) protocol, enter router configuration mode and specify an OSPFv3.

   ```
   router ospfv3 process-id
   ```

2. If you specified BGP or OSPFv3 in the previous step, enter address-family mode, specify IPv4 or IPv6, and specify the VRF for which to translate OMP metrics.

   ```
   address-family {ipv4 | ipv6} vrf vrf-name
   ```

3. Enable translation of OMP route metrics to BGP, OSPF, or OSPFv3 during redistribution of routes to a device outside of the Cisco Catalyst SD-WAN overlay network.

   You cannot use both the **redistribute omp translate-rib-metric** command and the **redistribute omp metric** command together on the same device. The **translate-rib-metric** option generates BGP attributes and OSPF metrics from OMP metrics, whereas the **metric** option configures the metrics explicitly.

   ```
   redistribute omp translate-rib-metric
   ```

4. In a scenario in which the underlay network uses BGP, enable propagation of the AS-Path metric. Omitting this causes a router to treat the AS-Path metric as empty.

   ```
   propagate-aspath
   ```

This example applies to a scenario in which the underlay network uses BGP.

```
router bgp 1
 address-family ipv4 vrf 2
  redistribute omp translate-rib-metric
  propagate-aspath
```

This example applies to a scenario in which the underlay network uses OSPF.

```
router ospf 1 vrf 1
 redistribute omp translate-rib-metric
```

This example applies to a scenario in which the underlay network uses OSPFv3 IPv4.

```
router ospfv3 1
 address-family ipv4 vrf 1
  redistribute omp translate-rib-metric
```

This example applies to a scenario in which the underlay network uses OSPFv3 IPv6.

```
router ospfv3 1
 address-family ipv6 vrf 1
  redistribute omp translate-rib-metric
```

Verify symmetric routing

Use these verification commands to confirm that your symmetric routing configuration is operating as intended.

Use these commands to verify the configurations required for symmetric routing. For more information see, Cisco IOS XE Catalyst SD-WAN Qualified Command Reference

Verify the next hops for a specific prefix on a router

Use **show sdwan omp routes** prefix on a router to show the next hops for a specific prefix.

```
Device# show sdwan omp routes 10.1.1.0/24
```

Verify the path to a destination router

Use **traceroute vrf** *vrf-number destination-ip-address* **numeric** on any device in the network to show the path from the device to a specified destination device, for a specified VRF.

The output shows a list of each hop in the path to the destination device. The last item in the list is the destination device.

```
Device# traceroute vrf 1 10.1.1.1 numeric
Type escape sequence to abort.
Tracing the route to 10.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.200.225 3 msec 1 msec 1 msec
  2 209.165.200.226 2 msec 1 msec 1 msec
  3 10.1.1.1 4 msec * 4 msec
```

Verify the VRF-specific affinity group configuration on a router

Use **show platform software sdwan rp active internal "omp daemon"** on a transport gateway, or a border router in a Multi-Region Fabric scenario, to show the VRF-specific affinity group configuration on a router. The output shows the affinity group for each configured VRF range.

See the procedures below for configuring VRF-specific affinity groups:

- Configure Router Affinity Groups for Specific VRFs Using Cisco SD-WAN ManagerConfigure Router Affinity Groups for Specific VRFs Using Cisco SD-WAN Manager
- Configure Router Affinity Groups for Specific VRFs Using a CLI Template Configure Router Affinity Groups for Specific VRFs Using a CLI Template

```
Device# show platform software sdwan rp active internal "omp daemon" | include
 Affinity
…
Affinity per VRF:

Affinity Group Number: 1 for VRF Range: 1-1
Affinity Group Number: 5 for VRF Range: 2-8
```

Verify a control policy for route leaking

Use **show running-config policy control-policy** on a Cisco SD-WAN Controller to show a control policy that configures route leaking from one VRF to another, if such a policy exists. Exporting routes from one VRF to another is called leaking routes.

For information about configuring a control policy that matches routes of a VRF list and exports the routes to a specific VRF, see Configure Centralized Policies Using the CLI.

Verify control policy application

Use **show running-config apply-policy** on a Cisco SD-WAN Controller to show the sites to which a control policy is applied.

This example shows a control policy that matches VRF1 routes and exports them to VRF2, and matches VRF2 routes and exports them to VRF1.

```
sdwanController# show running-config policy control-policy
policy
 control-policy LEAK_1_TO_2
  sequence 1
   match route
```

```
    vpn-list VRF1
   !
   action accept
    export-to
     vpn 2
    !
   !
  !
  default-action accept
 !
 control-policy LEAK_2_TO_1
  sequence 1
   match route
    vpn-list VRF2
   !
   action accept
    export-to
     vpn 1
    !
   !
  !
  default-action accept
 !
!
Example 2
The following example shows the sites to which the two policies configured in
 the previous example are applied.

sdwanController#show running-config apply-policy
apply-policy
 site-list SL1100
  control-policy LEAK_1_TO_2 in
 !
 site-list SL1300
  control-policy LEAK_2_TO_1 in
 !
!
```

Verify the derived affinity group of a route

Use **show sdwan omp routes***prefix***detail** on a transport gateway, or a border router in a Multi-Region Fabric scenario, to show the derived affinity group assigned to a prefix. The derived-affinity-group parameter in the output shows the value.

In this example the derived affinity group is 2.

```
Device# show sdwan omp routes 192.168.1.0/24 detail
…
  preference       not set
  affinity group  None
  derived-affinity-group  2
  affinity-preference-order  None
  region-id          0
  br-preference      not set
```

Monitor RIB metric translation

Explains how to monitor the translation of OMP metrics into service-side routing protocols.

For complete information about how a transport gateway translates RIB metrics, see Translating OMP Metrics for Devices Outside of the Overlay Network.

- OMP Metrics
- BGP Metrics
- OSPF Metrics

View OMP metrics

To view the OMP RIB metrics for a route, use the **show ip route** command on a transport gateway that is translating OMP RIB metrics.

The example below shows the OMP RIB metrics for the 10.1.1.1 route. The following metrics are shown in bold in the output:

- OMP Route Metric: 3
- OMP AS-PATH: 100 101

```
Device# show ip route vrf 1 10.1.1.1 protocol-internal
Routing Table: 1
Routing entry for 10.1.1.1/32
  Known via "omp", distance 251, metric 3, type omp
  Redistributing via bgp 1
  Advertised by bgp 1
  Last update from 10.100.1.2 00:04:35 ago
  Routing Descriptor Blocks:
  * 10.100.1.2 (default), from 10.100.1.2, 00:04:35 ago
      opaque_ptr 0x7FC8D1470748
        pdb 0x111111111110, ndb 0x111111111120, rdb 0x111111111130
        OMP attribute 0x7FC8D1470748, ref 2
        aspath 0x7FC8D1474870, ref 2, length 10, value 100 101
        Total OMP attr count 1, aspath 1, community 0
      Route metric is 3, traffic share count is 1
```

OMP route metric for IPv4 routes

To show the OMP route metric for each IPv4 route prefix that a transport gateway is redistributing, use the **show ip route** command on the transport gateway. The OMP route metric, which is 66, is shown in bold in the output, and the administrative distance is 251.

```
device# show ip route vrf 1 omp

Routing Table: 1

      10.0.0.0/32 is subnetted, 1 subnets
m        10.10.10.10 [251/66] via 172.16.0.1, 00:09:15
```

OMP route metric for IPv6 routes

To show the OMP route metric for each IPv6 route prefix that a transport gateway is redistributing, use the **show ipv6 route** command on the transport gateway. The OMP route metric, which is 66, is shown in bold in the output, and the administrative distance is 251.

```
device# show ipv6 route vrf 1 omp
m   2001:DB8::/128 [251/66]
```

```
     via 172.16.0.1%default
…
```

View BGP metrics

To view the derived BGP metrics for a route, use the **show ip bgp**command on a transport gateway that is translating OMP RIB metrics.

This example shows the derived BGP metrics for the 10.1.1.1 route. Though the example shows an IPv4 route, IPv6 routes are also supported. The following metrics are shown in bold in the output:

- BGP MED: 3
- BGP LOCAL_PREF: 252
- BGP AS_PATH: 100 100 100 100 101 (This is 100 100 100 (3 copies), plus the original 100 101 of the OMP AS-PATH value.)

```
device# show ip bgp vpnv4 all 10.1.1.1
BGP routing table entry for 1:1:10.1.1.1/32, version 2
Paths: (1 available, best #1, table 1)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  100 100 100 100 101
    10.100.1.2 (via default) from 0.0.0.0 (10.100.1.1)
      Origin incomplete, metric 3, localpref 252, valid, sourced, best
      Extended Community: SoO:0:0
      mpls labels in/out 16/nolabel
      rx pathid: 0, tx pathid: 0x0
      Updated on Apr 12 2023 19:08:17 EST
```

View OSPF metrics

To show that the redistribute omp translate-rib-metric command is active on a router, use the **show ip ospf** command. The result shown in bold in the output shows that the router is configured to translate RIB metrics.

- OSPF Metric for IPv4 Routes
- OSPF Metric for IPv6 Routes

OSPF Metric for IPv4 Routes

To show the OSPF metric that the transport gateway uses when distributing IPv4 routes to OSPF, use the **show ip ospf** command on the transport gateway. The OSPF metric, which is determined by the OMP route metric, is 66 in this example, and is shown in bold in the output.

```
Router#show ip ospf 1 rib redistribution
            OSPF Router with ID (192.168.0.1) (Process ID 1)

  Base Topology (MTID 0)

OSPF Redistribution
10.10.10.10/32, type 2, metric 66, tag 0, from OMP_AGENT Router
   via 172.16.0.1, unknown interface
…
```

OSPF Metric for IPv6 Routes

TTo show the OSPF metric that the transport gateway uses when distributing IPv6 routes to OSPF, use the **show ospfv3** command on the transport gateway. The OSPF metric, which is determined by the OMP route metric, is 66 in this example, and is shown in bold in the output.

```
Router#show ospfv3 vrf 1  ipv6 rib redistribution
          OSPFv3 10 address-family ipv6 vrf 1 (router-id 192.168.0.1)

2001:DB8::/128, type 2, metric 66, tag 0, from omp
  via 172.16.0.1
…
```

25 Radio Aware Routing

Topics:

- Feature history for RAR
- RAR for Cisco IOS XE Catalyst SD-WAN devices
- Configure RAR

Describes the RAR technology.

Feature history for RAR

Lists the development milestones and release information for RAR, including feature description.

This table describes the developments of this feature, by release.

Table 62: Feature History

Feature name	Release information	Description
Radio-Aware Routing Support	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	This feature enables Radio-Aware Routing (RAR) support on Cisco IOS XE Catalyst SD-WAN devices. RAR uses radio signals to interact with the OSPFv3 routing protocol. Through these signals it can indicate the appearance, disappearance, and link conditions of one-hop routing neighbors. In large mobile networks, connections to routing neighbors may be interrupted by distance and radio obstructions. RAR addresses the challenges that arise when integrating IP routing with radio communications in mobile networks.

RAR for Cisco IOS XE Catalyst SD-WAN devices

Describes the deployment topology and how SD-WAN tunnels are established over a PPPoE-RAR underlay network.

Radio-Aware Routing (RAR) is a mechanism that leverages radio interfaces to communicate directly with the Open Shortest Path First version 3 (OSPFv3) protocol, enabling real-time signaling of the presence and link conditions of one-hop routing neighbors.

- RAR improves routing responsiveness in mobile environments by allowing immediate link state updates.
- Standard protocol timers may be too slow for rapidly changing mobile network conditions.
- PPPoE provides the underlying connectivity, with support for OSPFv3 and EIGRP.
- An AX license is required to enable this feature.

In large mobile networks, factors such as distance and radio obstructions can frequently disrupt connections to routing neighbors. If these disruptions are not reported directly to the routing protocols, the protocols rely on their built-in timers to update neighbor status. However, these protocol timers are typically lengthy, which is not ideal for the dynamic conditions of mobile networks.

Connectivity between two Cisco IOS XE Catalyst SD-WAN devices is established over a PPPoE connection, which features variable bandwidth and limited buffering capacity. OSPFv3 and EIGRP are the supported routing protocols for this deployment.

Mobile Ad Hoc Networks (MANETs)

MANETs facilitate device-to-radio communications by addressing the challenges of integrating IP routing with mobile radio technologies in ad hoc networking environments. MANET routing protocols enable signaling between MANET routers, supporting features such as scope-limited flooding and point-to-point delivery of routing protocol messages within the network.

System components of RAR

The RAR feature uses the MANET infrastructure and includes several components: PPPoE, Virtual Multipoint Interface (VMI), QoS, routing protocol interface, and RAR protocols.

Point-to-Point Protocol over Ethernet PPPoE or PPPoE

PPPoE is a well-defined communication mechanism between the client and the server. In the RAR implementation, radio takes the role of the PPPoE client and router takes the role of the PPPoE server. This allows a loose coupling of radio and router, while providing a well-defined and predictable communication mechanism.

As PPPoE is a session or a connection-oriented protocol, it extends the point to point radio frequency (RF) link from an external radio to an IOS router.

PPPoE Extensions

In the Cisco IOS implementation of PPPoE, each session is represented by a virtual access interface, which connects to a radio neighbor. QoS can be applied on these interfaces using PPPoE extensions.

RFC5578 provides extensions to PPPoE to support credit-based flow control and session-based real-time link metrics, which are very useful for connections with variable bandwidth and limited buffering capabilities (such as radio links).

Virtual Multipoint Interface (VMI)

Though PPPoE Extensions provides most of the setup to communicate between a router and a radio, VMI addresses the need to manage and translate events that higher layers (for example, routing protocols) consume.

In bypass mode, every Virtual Access Interface (VAI) representing a radio neighbor is exposed to the routing protocols OSPFv3 and EIGRP so that the routing protocol directly communicates with the respective VAI for both unicast and multicast routing protocol traffic.

In Aggregate mode, VMI is exposed to the routing protocols (such as OSPF) so these protocols can optimize efficiency.

When the network neighbors are viewed as a collection of networks on a point-to-multipoint link with broadcast and multicast capability at VMI, VMI aggregates the multiple virtual access interfaces created from PPPoE. VMI provides a single, multi-access Layer 2 interface with broadcast capability. Using a single interface for the routing protocol reduces the size of the topology database without impacting network integrity. The VMI layer:

- Redirects unicast routing protocol traffic to the appropriate P2P link (Virtual-Access interface)
- Replicates multicast or broadcast traffic as needed

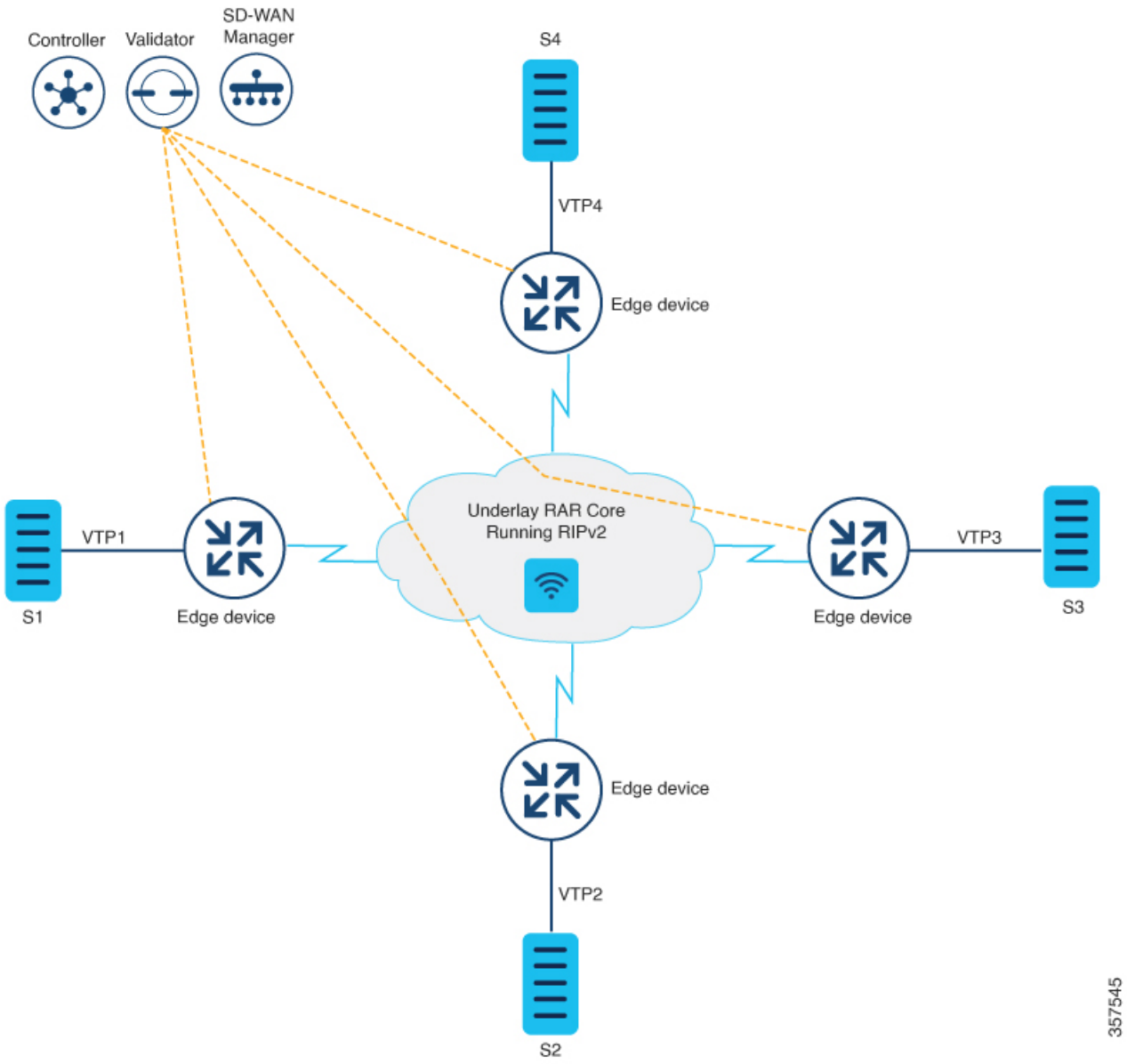
RAR architecture

Summary

The topology shows the RAR deployment on Cisco IOS XE Catalyst SD-WAN devices.

Workflow

Figure 35: RAR architecture



- The four Cisco IOS XE Catalyst SD-WAN devices connect to each other through a radio connected to a physical interface on the device
- PPPoE-RAR configurations happen on all three routers and once the underlay RAR network is established, the Cisco Catalyst SD-WAN tunnels form on the network.
- The loopback interface acts as a WAN interface and binds to the Virtual Multipoint interface (VMI). The VMI interface in turn binds to the physical interface
- The PPP connections between any two devices act as the underlay network.
- The Cisco Catalyst SD-WANCisco SD-WAN Manager tunnels are established over the PPPoE-RAR underlay network.
- Cisco SD-WAN Manager, Cisco SD-WAN Controller, and Cisco SD-WAN Validator connect through a radio connection in the deployment scenario.

Benefits of RAR

This section highlights the advantages of the feature, such as faster network convergence, optimal route selection for fading links, and efficient bandwidth usage in mobile environments.

The RAR feature offers these benefits:

- Provides faster network convergence through immediate recognition of changes.
- Enables routing for failing or fading radio links.
- Allows easy routing between line-of-sight and non-line-of-sight paths.
- Provides faster convergence and optimal route selection so that delay-sensitive traffic, such as voice and video, is not disrupted
- Provides efficient radio resources and bandwidth usage.
- Reduces impact on the radio links by performing congestion control in the router.
- Allows route selection based on radio power conservation.
- Enables decoupling of the routing and radio functionalities.
- Provides simple Ethernet connection to RFC 5578, R2CP, and DLEP compliant radios.

Supported devices for RAR

This section lists the specific hardware platforms compatible with RAR, including various ISR 1000/4000 and ASR 1000 series routers.

These platforms support RAR:

- Cisco 4000 Series Integrated Services Routers
- Cisco 1000 Series Integrated Services Routers
- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco CSR 1000 Series Cloud Service Routers
- Cisco CSR 8000 Series Cloud Service Routers

Restrictions for RAR

This section details the limitations of the feature, including lack of support for DLEP/R2CP protocols and Cisco High Availability (HA) technology.

The Radio Aware Routing feature has these restrictions:

- The Dynamic Link Exchange Protocol (DLEP) and Router to Radio Control Protocol (R2CP) protocols are not supported.
- Multicast traffic is not supported in aggregate mode.
- Cisco High Availability technology is not supported.

Prerequisites for RAR

This section outlines the necessary technical requirements.

The RAR configuration requires Mobile Ad-hoc Networks (MANETs) support. To use the PPP over Ethernet (PPPoE) and virtual multipoint interface (VMI) features for RAR, a unified representation of the MANET to routing protocols (OSPFv3 or EIGRP) is required.

Configure RAR

Explains about setting up RAR with CLI add-on feature templates in SD-WAN Manager, and configure RAR in bypass and aggregate modes.

To configure RAR using SD-WAN Manager, Create a CLI add-on feature template and attach it to the device template.

This section provides examples of RAR configurations that you can add to the CLI add-on template.

Configure a service for RAR:

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configure OSPF routing:

```
router ospfv3 1
 router-id 10.0.0.1
!
 address-family ipv4 unicast
  redistribute connected metric 1 metric-type 1
  log-adjacency-changes
 exit-address-family
 !
 address-family ipv6 unicast
  redistribute connected metric-type 1
  log-adjacency-changes
 exit-address-family
!
ip local pool PPPoEpool2 192.0.2.0 192.0.2.1
```

Configuration of RAR:

```
interface GigabitEthernet0/0/0
no shutdown
no mop enabled
no mop sysid
negotiation auto
pppoe enable group PPPOE_RAR

interface vmi1
ip address 10.0.0.0 255.255.255.0
ipv6 enable
physical-interface GigabitEthernet0/0/0
mode bypass
exit
interface Virtual-Template1
no shutdown
ip unnumbered vmi1
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
exit

interface Tunnel100
 no shutdown
 ip unnumbered Loopback100
```

```
 tunnel source Loopback100
 tunnel mode sdwan
exit

interface Loopback100
  tunnel-interface
   encapsulation ipsec
   color mpls
   no allow-service bgp
   allow-service dhcp
exit

router ospfv3 1
router-id 10.0.0.1
address-family ipv4 unicast
log-adjacency-changes
redistribute connected
redistribute connected metric 1 metric-type 1
exit-address-family
!
address-family ipv6 unicast
log-adjacency-changes
redistribute connected
redistribute connected metric-type 1
exit-address-family
```

The following example describes QoS provisioning on PPPoE extension session:

```
policy-map rar_policer
 class class-default
  police 10000 2000 1000 conform-action transmit  exceed-action drop
violate-action drop
policy-map rar_shaper
 class class-default
  shape average percent 1

interface Virtual-Template2
 ip address 192.0.2.255 255.255.255.0
 no peer default ip address
 no keepalive
 service-policy input rar_policer
end
```

Configure RAR in bypass mode

Before you begin the RAR configuration, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enbaling authorization, the Point-to-Point protocol does not recognize this as a RAR session and may not tag manet_radio in PPPoE protocol. By default, bypass mode does not appears in the configuration. It appears only if the mode is configured as bypass.

Configure a service for RAR:

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configure Broadband:

```
interface pppoe VMI2
 virtual-template 2
service profile rar-lab
!
interface GigabitEthernet0/0/0
 description Connected to Client1
  negotiation auto
  pppoe enable group VMI2
!
```

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configuration in bypass mode:

This exmple shows IP Address configured under virtual template explicitly:

```
interface Virtual-Template2
ip address 192.0.2.255 255.255.255.0
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

This example shows VMI unnumbered configured under virtual template:

```
interface Virtual-Template2
ip unnumbered vmi2
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

Configure the virtual multipoint interface in bypass mode:

```
interface vmi2 //configure the virtual multi interface
ip address 192.0.2.255 255.255.255.0
physical-interface GigabitEthernet0/0/0
mode bypass
interface vmi3//configure the virtual multi interface
 ip address 192.0.2.255 255.255.255.0
 physical-interface GigabitEthernet0/0/1
mode bypass
```

Configure RAR in aggregate mode

Before you configure RAR, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enabling authorization, the Point-to-Point protocol does not recognize this as a RAR session and may not tag manet_radio in PPPoE.

The following example is an end-to-end configuration of RAR in the aggregate mode:

Configure a Service for RAR:

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configure Broadband:

```
bba-group pppoe VMI2
 virtual-template 2
service profile rar-lab

!
interface GigabitEthernet0/0/0
 description Connected to Client1
  negotiation auto
  pppoe enable group VMI2

!
```

```
policy-map type service rar-lab
 pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configuration in Aggregate Mode:

```
interface Virtual-Template2
ip unnumbered vmi2
no ip redirects
no peer default ip address
ipv6 enable
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```