Cisco Catalyst SD-WAN EtherChannel

Table 1: Feature History

Feature Name	Release Information	Description
Cisco Catalyst SD-WAN EtherChannel	Cisco IOS XE Catalyst SD-WAN Release 17.6.1a Cisco vManage Release 20.6.1	This feature allows you to configure EtherChannels on Cisco IOS XE Catalyst SD-WAN devices on the service side. An EtherChannel provides fault-tolerant high speed link, redundancy, and increased bandwidth between Cisco IOS XE Catalyst SD-WAN devices and other devices such as routers, switches, or servers connected in a network. You can configure EtherChannels only using the CLI device templates and CLI add-on feature templates.

Feature Name	Release Information	Description
EtherChannels on the Transport Side	Cisco IOS XE Catalyst SD-WAN Release 17.13.1a Cisco Catalyst SD-WAN Manager Release 20.13.1	Adds support for configuring EtherChannels on the transport side of a Cisco IOS XE Catalyst SD-WAN device. This feature also introduces support for aggregate EtherChannel Quality of Service (QoS) on the transport side. By combining EtherChannel and QoS, you can optimize network utilization, enhance performance, and maintain quality for specific traffic types. **Note** This feature has limited availability.
Load Balancing for EtherChannels on the Transport Side	Cisco IOS XE Catalyst SD-WAN Release 17.14.1a Cisco Catalyst SD-WAN Manager Release 20.14.1	This feature adds the ability to configure load balancing for EtherChannels on the transport side for Cisco IOS XE Catalyst SD-WAN devices.
Configure EtherChannels using Configuration Groups	Cisco IOS XE Catalyst SD-WAN Release 17.15.1a Cisco Catalyst SD-WAN Manager Release 20.15.1	With this feature you can configure EtherChannels on service and transport side using configuration groups.
Load Balancing for EtherChannels on Individual Port Channels	Cisco IOS XE Catalyst SD-WAN Release 17.15.1a Cisco Catalyst SD-WAN Manager Release 20.15.1	With this feature you can load balance EtherChannels for individual port channels on service and transport side using CLI templates.

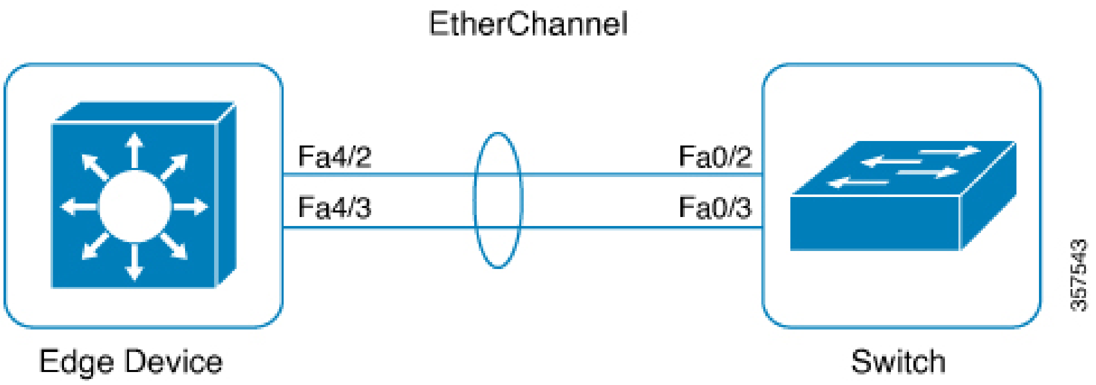
- Information About Cisco Catalyst SD-WAN EtherChannel, on page 2
- Benefits of Cisco Catalyst SD-WAN EtherChannel, on page 7
- EtherChannels on the Service Side, on page 7
- EtherChannels on the Transport Side, on page 14
- Monitor Configured EtherChannel Using CLI, on page 20
- Aggregate EtherChannel Quality of Service, on page 20

Information About Cisco Catalyst SD-WAN EtherChannel

An EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. You can use the EtherChannel to increase bandwidth between the wiring closets and the data center, and also deploy it at any place in a network where bottlenecks are likely to occur. An EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, an EtherChannel redirects traffic from the failed link to the remaining links in the channel.

An EtherChannel comprises a channel group and a port-channel interface. The channel group binds physical ports to the port-channel interface. Configuration changes applied to the port-channel interface apply to all the physical ports bound together in the channel group.

Figure 1: EtherChannel



- Using EtherChannels in a network provides increased bandwidth and resilience.
  - Bandwidth: An EtherChannel allows multiple links to be combined into one logical link. Because an EtherChannel offers redundancy of links, you can configure EtherChannels to increase the speed in a network.
  - Resilience: An EtherChannel also provides network resilience. Even if a link within an EtherChannel fails, traffic that is previously carried over the failed link switches to the remaining links within the EtherChannel. Thus, EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links.
- The number of supported port channels differs based on the specific device model.
- The number of supported member interfaces for a port channel differs based on the specific device model.
- EtherChannel supports the following combinations:
  - Two active links
  - Active and passive links
  - Single member link
  - Loopback interface in bind or unbind mode to the port channel

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.13.1a and Cisco Catalyst SD-WAN Manager Release 20.13.1, EtherChannels configured on the transport side support the following:

- Control and management connections (DTLS, OMP) to Cisco Catalyst SD-WAN Manager, Cisco Catalyst SD-WAN Validator, and Cisco Catalyst SD-WAN Controller
- IPSEC tunnels for data traffic
- IPv4 forwarding

- L2 TLOC extension
- Explicit ACL (Access Control Lists)
- Implicit ACL on a port channel TLOC
- IPv4 static routing
- Loopback TLOC (ability to bind loopback to port channel)
- Port channel sub-interfaces
- Control policies on Cisco Catalyst SD-WAN Controller

Starting from Cisco IOS XE Catalyst SD-WAN Release 17.14.1a and Cisco Catalyst SD-WAN Manager Release 20.14.1, EtherChannels configured on the transport side support the following:

- IPv6 for EtherChannels, allowing for the transmission of IPv6 traffic across aggregated links.
- Handling traffic through Generic Routing Encapsulation (GRE) tunnels, facilitating the encapsulation of various network protocols.
- Advanced routing protocols such as OSPF and BGP over EtherChannels enables dynamic routing in Cisco Catalyst SD-WAN.
- NAT-DIA across EtherChannels, providing direct internet access by converting private IP addresses to public ones for efficient internet-bound traffic routing. For more information, see Configure NAT.

EtherChannel in Cisco Catalyst SD-WAN

To create an EtherChannel, begin by configuring a port channel. A port channel is a logical interface on a Cisco IOS XE Catalyst SD-WAN device. After you create an EtherChannel, the configuration changes that are applied to the port-channel interface are also applied to all the physical ports assigned to the port-channel interface.

The maximum number of interfaces that can be combined into a single EtherChannel using LACP is eight, although the actual limit may depend on the specific model of the device.

You can configure an EtherChannel using one these methods:

- Link Aggregation Control Protocol (LACP) mode
- Static mode

  Use the LACP mode to configure an EtherChannel if it is supported on both ends of a device. If either of the device does not support LACP mode, use a static mode to configure an EtherChannel.

LACP Mode

LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between the Ethernet ports.

This table shows the user-configurable EtherChannel LACP modes.

Table 2: EtherChannel LACP Modes

Mode	Description
active	Places a port in an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.
passive	Places a port in a passive negotiating state in which the port responds to the packets that it receives, but does not start LACP packet negotiation. This setting minimizes the transmission of LACP packets.

Both the **active** and **passive** modes enable ports to negotiate with partner ports based on port speed.

Ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A port in the **active** mode can form an EtherChannel with another port that is in the **active** or **passive** mode.
- A port in the **passive** mode cannot form an EtherChannel with another port that is also in the **passive** mode because neither port starts LACP negotiation.

In addition to the standard LACP configuration, the following LACP-related commands are supported:

- **lacp min-bundle**
- **lacp max-bundle**
- **lacp system-priority**
- **lacp port-priority**
- **lacp fast-switchover**
- **lacp rate fast**

Static Mode

You can manually create an EtherChannel by using the **interface port-channel** command in the global configuration mode. You then use the **channel-group interface** command in the global configuration mode to assign an interface to the EtherChannel. After you configure an EtherChannel, the configuration changes applied to the port-channel interface are applied to all the physical ports assigned to the port-channel interface. Unlike an LACP mode, in a static mode, no packets are sent for negotiations with the other ports. Instead, you must manually configure the ports as part of an EtherChannel.

Information related to LACP on port-channel interfaces can be obtained using the **show lacp** command. See show lacp.

EtherChannel Load Balancing

An EtherChannel balances traffic load across the links in a channel. You can specify one of several different load-balancing modes. EtherChannels can use either dynamic flow-based load balancing or virtual LAN (VLAN) manual load balancing.

You can configure the load-balancing method globally for all the port channels or directly on specific port channels. The global configuration applies only to those port channels for which you have not explicitly configured load balancing. The port-channel configuration overrides the global configuration.

The following load-balancing methods are supported on Cisco IOS XE Catalyst SD-WAN devices:

- Flow-Based

  VLAN-Based

Flow-Based Load Balancing

Flow-based load balancing is the default load-balancing method, and is enabled by default at the global level. Flow-based load balancing identifies different flows of traffic based on the key fields in the data packet. For example, IPv4 source and destination IP addresses can be used to identify a flow. The various data traffic flows are then mapped to the different member links of a port channel. After the mapping is done, the data traffic for a flow is transmitted through the assigned member link. The flow mapping is dynamic and changes when there is any change in the state of a member link to which a flow is assigned. The flow mapping is dynamic when member links are added or deleted.

VLAN-Based Load Balancing

VLAN-based load balancing allows you to configure static assignment of user traffic, as identified by a VLAN ID, to a given member link of an EtherChannel. You can manually assign VLAN subinterfaces to a primary and secondary link. This feature allows load balancing to downstream equipment regardless of vendor equipment capabilities, and provides failover protection by redirecting traffic to the secondary member link if the primary link fails. Member links are supported with up to 16 bundles per chassis.

EtherChannels Load Balancing on the Transport Side of Cisco IOS XE Catalyst SD-WAN Devices

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.14.1a and Cisco Catalyst SD-WAN Manager Release 20.14.1.

Load balancing for EtherChannels on the transport side is achieved by using the inner IP headers, which include the original source and destination IP addresses found in encapsulated packets. Cisco IOS XE Catalyst SD-WAN devices use a hash algorithm to analyze the inner IP addresses for distribution of network traffic across available paths.

Configure load balancing for EtherChannels on the transport side using the **port-channel load-balance-hash-algo sdwan** command. With load balancing configured, a router distributes network traffic among all available paths within the EtherChannel. By default, **sdwan** uses the inner packet source and destination IP address.

Information About Configuring EtherChannels using Configuration Groups

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a and Cisco Catalyst SD-WAN Manager Release 20.15.1

You can configure port channel interfaces and member links using configuration groups in Cisco SD-WAN Manager.

Load Balancing on the Transport Side for Individual Port Channels

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a and Cisco Catalyst SD-WAN Manager Release 20.15.1

You can enable load balancing on per port channel in the interface using the **load-balance-hash-algo sdwan** command.

Benefits of Cisco Catalyst SD-WAN EtherChannel

- Provides fault-tolerance. If any one of the links in an EtherChannel fail, the EtherChannel automatically redistributes traffic across the remaining links.
- Helps increase bandwidth between Cisco IOS XE Catalyst SD-WAN devices and other devices such as switches and servers that are connected in a network.

EtherChannels on the Service Side

Supported Devices for Cisco Catalyst SD-WAN EtherChannel on the Service Side

Service Side

The following platforms support EtherChannel and also offer load balancing for EtherChannel on the service side:

- Cisco 4000 Series Integrated Services Routers
  - Cisco 4451-X Integrated Services Router
  - Cisco 4461 Integrated Services Router
  - Cisco 4431 Integrated Services Router
  - Cisco 4331 Integrated Services Router
  - Cisco 4351 Integrated Services Router
- Cisco ASR 1000 Series Aggregation Services Routers
  - Cisco ASR 1001-X Router
  - Cisco ASR 1006-X Router
  - Cisco ASR 1001-HX Router
  - Cisco ASR 1002-HX Router
  - Cisco ASR 1002-X Router
- Cisco Catalyst 8000V Edge Software
- Cisco Catalyst 8200 Router
- Cisco Catalyst 8300 Router
- Cisco Catalyst 8500 Series Edge Router

Supported NIMs

The following NIMs are supported on Integrated Services Routers, for service side:

- NIM-1GE-CU-SFP
- NIM-2GE-CU-SFP
- SM-X-4x1G-1x10G
- SM-X-6X1G
- C-NIM-2T
- C-NIM-1X
- C-NIM-1M

Note Network Interface Modules (NIMs) with L2 ports do not support EtherChannels on the service side.

Prerequisites for Cisco Catalyst SD-WAN EtherChannel on the Service Side

- All the LAN ports in each EtherChannel must be of the same speed.
- All the LAN ports must be configured on Layer 3 service-side ports.
- All member interfaces in a portchannel must have the same speed and duplex, when using platforms that support multiple rate SFPs on the same port.

Restrictions for Cisco Catalyst SD-WAN EtherChannel on the Service Side

- The maximum number of port channel interfaces that a device can support varies, depending on the particular model of the device.
- You can configure EtherChannels on a device by using the CLI, or using only the CLI templates or CLI add-on feature templates in Cisco SD-WAN Manager.
- Network Interface Modules (NIMs) with L2 ports do not support EtherChannels on the service side.
- The EtherChannel Quality of Service (QoS) feature on port channels is not supported on the service side.
- The Aggregate EtherChannel QoS feature on port channels is not supported on the service side.
- An EtherChannel does not support Digital Signal Processor (DSP) farm services and voice services.
- Sub interfaces cannot be added as member of EtherChannel.

Configure Load Balancing for EtherChannels on the Service Side Using CLI Commands

For more information about using CLI templates, see CLI Templates.

Note

- From Cisco Catalyst SD-WAN Manager Release 20.15.1, you can configure any other hash algorithms for flow-based load balancing on per port-channel interface on the service side.
- Load balancing uses a flow-based method by default, with the default hash algorithm being **src-dst-ip**.
- The Hash Algorithms For Flow-Based Load Balancing feature is supported on Cisco Aggregation Services Routers platforms, and Cisco Catalyst Router platforms, where the hardware load-balancing for Etherchannel is supported. This command is not supported on Cisco Integrated Services Routers.

Enable Load Balancing on an Individual Port Channel

1. Enter the port channel interface configuration mode.

   ```
   interface Port-channel channel-number
   ```

2. Enable load balancing on an individual port channel.

   ```
   load-balancing flow
   ```

This example shows how to set the load-balancing method to flow, when VLAN-manual method is configured globally:

```
Device# config-transaction
Device(config)# interface port-channel 1
Device(config-if)# load-balancing flow
```

This example shows how to set the load-balancing method to VLAN:

```
Device# config-transaction
Device(config)# interface port-channel 1
Device(config-if)# load-balancing vlan
```

This example shows a configuration where flow-based load balancing is configured on port channel 2 while the VLAN-manual method is configured globally:

```
  port-channel load-balancing vlan-manual
  interface Port-channel2
   ip address 10.0.0.1 255.255.255.0
   load-balancing flow

  interface GigabitEthernet2/1/0
   no ip address
   channel-group 2

  interface GigabitEthernet2/1/1
   no ip address
   channel-group 2
```

This example shows configuration for VLAN when the load balancing is set to default on the global level:

```
port-channel load-balancing vlan-manual


interface Port-channel1
interface Port-channel1.100
 encapsulation dot1Q 100 primary GigabitEthernet 1/1/1
 secondary GigabitEthernet 1/2/1
 ip address 10.16.2.100 255.255.255.0
```

```
interface Port-channel1.200
 encapsulation dot1Q 200 primary GigabitEthernet 1/2/1
 ip address 10.16.3.200 255.255.255.0
interface Port-channel1.300
 encapsulation dot1Q 300
 ip address 10.16.4.300 255.255.255.0

interface GigabitEthernet 1/1/1
 no ip address
 channel-group 1!
interface GigabitEthernet 1/2/1
 no ip address
 channel-group 1
```

Note Interface 1 and interface 2 must be member ports of a port channel when **encapsulation dot1q** is configured.

Enable Hash Algorithms for Flow-Based Load Balancing on a Global level

To configure specific flow-based hash algorithms on a global level use:

port-channel load-balance-hash-algo *hash-algo*

This example shows configuration for enabling a hash algorithm on a global level flow-based load balancing:

```
device(config)# port-channel load-balance-hash-algo src-mac
```

Enable Flow-Based Load Balancing on an Individual Port Channel Interface

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a and Cisco Catalyst SD-WAN Manager Release 20.15.1

1. Enter the port channel interface configuration mode.

   ```
   interface Port-channel channel-number
   ```

2. Enable flow-based load balancing hash algorithm.

   ```
   load-balance-hash-algo dst-ip
   ```

This example shows configuration of hash algorithms for flow-based load balancing on an individual port channel interface. When **sdwan** hash algorithm is configured on the transport side, you can enable different hash algorithm options on the service side.

```
device(config)# interface Port-channel 1
device(config-if)# load-balance-hash-algo sdwan
device(config-if)# exit
device(config)# interface Port-channel 2
device(config-if)# load-balance-hash-algo src-dst-mixed-ip-port
device(config-if)# exit
device(config)# interface Port-channel 3
device(config-if)# no shut
device(config-if)# commit
device(config-if)# end
```

Enable VLAN Load Balancing Per Port Channel on the Service Side

1. Enter the port channel interface configuration mode.

```
interface Port-channel channel-number
```

2. Enable vlan on per port channel.

```
load-balancing vlan
```

This example shows configuration for VLAN load balancing on the service side, when the flow-based load balancing is set to default on the global level:

```
interface Port-channel 1
load-balancing vlan
```

Configure a Service-Side Port Channel and Member Link Interface using Configuration Groups in Cisco SD-WAN Manager

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Under **Service Profile**, click **Add New** to add a new service profile or select any existing service profiles, and click **Edit**.
3. Select +, and click on the **Ethernet Interface** to add a new interface. See Ethernet Interface for more details.
4. Select **Add New** from the Ethernet Interface drop-down menu.
5. Under **Basic Configurations**, enter the interface name and description.
6. Click **Ether Channel**, and assign the EtherChannel to a **member interface** or a **port channel** interface from the drop down menu.

   If you configure the EtherChannel interface as a port channel, then the default port is LACP. You can assign a different port channel mode using the **Port Channel Mode** option from Cisco SD-WAN Manager.

Note You can create multiple interfaces which can be either member or port channel interfaces.

7. Click **Save**.
8. Click **Configuration** tab, and deploy the newly created port channels.

Configure Load Balancing for EtherChannels on the Service Side Using CLI Commands

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

Enable Flow Based Load Balancing on a Global Level

Note The default hash algorithm for flow-based load balancing is **src-dst-ip**.

```
Port-channel load-balancing flow
```

Enable Flow Based Load Balancing Per Port Channel

```
interface Port-channel channel-number
load-balancing flow
```

Enable Hash Algorithms for Flow based Load Balancing for Each Portchannel Interface

```
interface Port-channel 1
load-balance-hash-algo [src-dst-ip dst-ip | dst-mac | src-dst-ip |
src-dst-mac | src-dst-mixed-ip-port | src-ip | src-mac ]
```

Enable Hash Algorithms for Flow based Load Balancing on a Global level

```
port-channel load-balance-hash-algo {dst-ip | dst-mac | src-dst-ip |
src-dst-mac | src-dst-mixed-ip-port | src-ip | src-mac}
```

Note From Cisco Catalyst SD-WAN Manager Release 20.15.1, you can use any other hash algorithims for load balancing on the service side.

The **Hash Algorithms For Flow-based Load Balancing** feature is supported only on Cisco Aggregation Services Routers platforms, where the hardware load-balancing for Etherchannel is supported. This command is not supported on Cisco Integrated Services Routers and Cisco Catalyst Router platforms.

This example shows a configuration where flow-based load balancing is configured on port channel 2 while the VLAN manual method is configured globally:

```
!
port-channel load-balancing vlan-manual

.
.
.
interface Port-channel2
 ip address 10.0.0.1 255.255.255.0
 no negotiation auto
 load-balancing flow
!

interface GigabitEthernet2/1/0
 no ip address
 negotiation auto
 cdp enable
 channel-group 2
!
interface GigabitEthernet2/1/1
 no ip address
 negotiation auto
 cdp enable
 channel-group 2
!
```

This example shows a configuration for each port channel interface where port-channel1 uses **sdwan** hash algorithm for the transport side, Port-channel2 uses the **src-dst-mixed-ip-port** for the service side, and Port-channel3 uses the globally default **src-dst-ip** hash algorithm for the service side. :

```
device(config)# interface Port-channel 1
device(config-if)# load-balance-hash-algo sdwan
device(config-if)# exit

device(config)# interface Port-channel 2
device(config-if)# load-balance-hash-algo src-dst-mixed-ip-port
device(config-if)# exit

device(config)# interface Port-channel 3
device(config-if)# no shut
device(config-if)# commit
device(config-if)# end
```

The following is a sample output to view the configuration for each interface port channel using **show etherchannel load-balancing** command.

```
device# show etherchannel load-balancing
flow-based
LB Algo type: Source Destination IP

 Port-Channel:                      LB Method
    Port-channel1                  :  flow-based (SDWAN Inner packet LB)
    Port-channel2                  :  flow-based (Source Destination Port, IP addr)
    Port-channel3                  :  flow-based (Source Destination IP)
```

Manual Traffic Distribution Based on VLAN ID

port-channel load-balancing vlan-manual

Note This command is available for configuration in the global configuration mode, and applies to all the port-channel configured on the device.

This example shows how the load-balancing configuration can be globally applied to define policies for handling traffic by using the **port-channel load-balancing** command.

```
port-channel load-balancing vlan-manual

!
interface Port-channel1
!
interface Port-channel1.100
 encapsulation dot1Q 100 primary GigabitEthernet 1/1/1
 secondary GigabitEthernet 1/2/1
 ip address 10.16.2.100 255.255.255.0
!
interface Port-channel1.200
 encapsulation dot1Q 200 primary GigabitEthernet 1/2/1
 ip address 10.16.3.200 255.255.255.0
!
interface Port-channel1.300
 encapsulation dot1Q 300
 ip address 10.16.4.300 255.255.255.0
 !
interface GigabitEthernet 1/1/1
 no ip address
 channel-group 1!
interface GigabitEthernet 1/2/1
 no ip address
 channel-group 1
```

Enable VLAN Load Balancing Per Port Channel on the Service Side

```
interface Port-channel channel-number
 load-balancing vlan
```

This example shows configuration for VLAN load balancing on the service side, when the flow-based load balancing is set to default on the global level:

```
 interface Port-channel channel-number
 interface GigabitEthernet slot/subslot/port
  channel-group channel-group-number
  interface GigabitEthernet slot/subslot/port
  channel-group channel-group-number
 interface Port-channel channel-number
  load-balancing vlan
 interface Port-channel channel-number
  encapsulation dot1Q vlan_id primary interface1 secondaryinterface2
```

Note Interface 1 and interface 2 must be member ports of a port channel when **encapsulation dot1q** is configured.

EtherChannels on the Transport Side

Supported Devices for Cisco Catalyst SD-WAN EtherChannel on the Transport Side

From Cisco IOS XE Catalyst SD-WAN Release 17.13.1a, the following platforms support EtherChannels on the transport side. From Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, the following platforms support load balancing:

- Cisco 4000 Series Integrated Services Routers
    - Cisco 4461 Integrated Services Router
- Cisco ASR 1000 Series Aggregation Services Routers
    - Cisco ASR 1001-HX Router
    - Cisco ASR 1002-HX Router
- Cisco Catalyst 8200 Series Edge Routers
- Cisco Catalyst 8300 Series Routers
- Cisco Catalyst 8500 Series Edge Routers

Note Starting with Cisco IOS XE Catalyst SD-WAN Release 17.14.1a, the load balancing configuration command **portchannel load-balance-hash-algo sdwan** is supported only on the Cisco 4461 Integrated Services Router and Cisco Catalyst 8300 Series routers.

Prerequisites for Cisco Catalyst SD-WAN EtherChannel on the Transport Side

- All the member links in each EtherChannel must be of the same speed.
- All the member links must be configured on Layer 3 transport side ports.
- All member interfaces in a portchannel must have the same speed and duplex, when using platforms that support multiple rate SFPs on the same port.

Restrictions for Cisco Catalyst SD-WAN EtherChannel on the Transport Side

- The maximum number of port channel interfaces that a device can support varies, depending on the particular model of the device.
- You can configure EtherChannels on a device by using the CLI, or using only the CLI templates or CLI add-on feature templates in Cisco SD-WAN Manager.
- Network Interface Modules (NIMs) with L2 ports do not support EtherChannels on the transport side.
- The Multichassis Link Aggregation Group (LAG), which involves different member links connecting to different switches, is not supported.
- The use of port channel on virtual devices such as Cisco Catalyst 8000V is not supported.
- Cisco IOS XE Catalyst SD-WAN Release 17.13.1a does not include support for an endpoint tracker on port-channel TLOCs.
- Platforms such as the Cisco Catalyst 8500 Series Edge Routers support multi-rate interfaces, allowing 1G SFP modules to be used in default 10G interfaces. Despite this, in the output of **show** commands, the interfaces appear as TenGigabitEthernet x/x/x. You can bundle the 1G SFP interfaces together to form a port channel.

Configure a Transport Side EtherChannel Using a CLI Template

In Cisco Catalyst SD-WAN Manager, you can configure EtherChannels on the transport side 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.

1. Configure a Layer 3 port channel.

```
interface Port-channel channel-number
ip address ip-address mask
ipv6 address ipv6-address/prefix-length
```

2. Assign Interfaces to a Layer 3 port channel with LACP active or passive options.

   a.

   ```
   interface GigabitEthernet slot/subslot/port
   no ip address
   channel-group channel-group-number mode {active passive}
   exit
   ```

   b. Configure EtherChannel with LACP Paramaters.

   ```
   lacp system-priority priority
   interface GigabitEthernet slot/subslot/port
   lacp port-priority priority
   ```

   c. Configure a static EtherChannel.

   ```
   interface GigabitEthernet slot/subslot/port
   no ip address
   channel-group channel-group-number
   ```

3. Configure tunnels.

   ```
   interface Tunnel tunnel-number
   ip unnumbered Port-channel channel-group-number
   no ip redirects
   tunnel source  Port-channel channel-group-number
   tunnel mode sdwan

   sdwan
    interface Port-channel channel-group-number
     tunnel-interface
      encapsulation {ipsec gre}
      color color-type
   ```

This example shows how to configure a Layer 3 EtherChannel, and how to assign two ports to channel 1 with the LACP mode as active and passive:

```
interface Port-channel1
ip address 10.48.48.15 255.255.255.0
ip ospf priority 0
ip ospf 65535 area 51
load-interval 30
no negotiation auto

interface GigabitEthernet0/0/0
no ip address
negotiation auto
lacp rate fast
channel-group 1 mode active
end
```

```
interface GigabitEthernet0/0/4
no ip address
negotiation auto
lacp rate fast
channel-group 1 mode passive
end
```

The following is a configuration example for creating an EtherChannel on the transport side.

```
interface Tunnel2
ip unnumbered Port-channel1
tunnel source Port-channel1
tunnel mode sdwan

interface Port-channel1
  tunnel-interface
   encapsulation ipsec
   color lte
```

Configure a Transport-Side Port Channel and Member Link Interface using Configuration Groups in Cisco SD-WAN Manager

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.
2. Under **Transport and Management**, click **Add New** to add a new service profile or select any existing service profiles, and click **Edit**.
3. Select **+**, and click on the **Ethernet Interface** to add a new interface. See Ethernet Interface for more details.
4. Select **Add New** from the Ethernet Interface drop-down menu.
5. Under **Basic Configurations**, enter the interface name and description.
6. Click **Ether Channel**, and assign the EtherChannel to a port channel or member interface from the drop down menu.

Note You can create multiple interfaces which can be either member or port channel interfaces.

7. Select **Global** from the **Tunnel** field, to enable a tunnel for the port channel interface.

Note To reassign the ethernet interface type, you must delete the existing ethernet interface feature and create a new one.

8. Click **Save**.
9. Click **Configuration** tab, and deploy the newly created port channels.

Configure Load Balancing for EtherChannels on the Transport Side Using CLI Commands

Enable Load Balancing on Individual Portchannel Interface

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.15.1a Cisco Catalyst SD-WAN Manager Release 20.15.1

Note We recommend using this method to configure load balancing for EtherChannels on the transport side.

1. Enter the port channel interface configuration mode.

   interface Portchannel *channel number*

2. Enable load balancing on an individual port channel.

   load-balance-hash-algo sdwan

Enable Load Balancing Globally for EtherChannels on the Transport Side

Minimum supported releases: Cisco IOS XE Catalyst SD-WAN Release 17.14.1a and Cisco Catalyst SD-WAN Manager Release 20.14.1

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

Enable load balancing globally for EtherChannels on the transport side.

```
port-channel load-balance-hash-algo sdwan
```

Note In this command, **port-channel load-balance-hash-algo sdwan**, the **sdwan** option was added in Cisco IOS XE Catalyst SD-WAN Release 17.14.1a.

Enable Hash Algorithms Globally for EtherChannels on the Transport Side

1. Configure the algorithm used for load balancing.

   To configure load balancing for IPv4 addresses, which is the default setting, use the following configuration:

   ```
   sdwan
   ip load-sharing algorithm {src-dst-ip|ip-and-ports|src-ip-only}
   ```

   To configure load balancing for IPv6 addresses, use the following configuration:

   ```
   sdwan
   ipv6 load-sharing algorithm {src-dst-ip|ip-and-ports|src-ip-only}
   ```

   - **src-dst-ip**: Balances traffic based on both source and destination IP addresses.
   - **ip-and-ports**: Balances traffic using a combination of IP addresses and port numbers.
   - **src-ip-only**: Balances traffic based solely on the source IP address.

The **ip load-sharing algorithm** command is a global configuration that applies to all Cisco Catalyst SD-WAN tunnels. Changing the algorithm with options such as **src-dst-ip** or **src-dst-mixed-ip-port** affects the load-sharing mechanism for other Cisco Catalyst SD-WAN tunnel traffic as well.

When you configure a port channel on both the service side and the transport side, using the **port-channel load-balance-hash-algo sdwan** command applies load balancing to the transport side. For the Service side, the port channel defaults to the **src-dst-ip** load balancing mode. For more information, see Configure Network Interfaces.

To change the load-balancing algorithm for the Service side when a Transport-VPN port-channel is also configured, use the **port-channel load-balance-hash-algo** command. This command allow you to switch from the default **sdwan** mode to alternative modes such as **dst-ip**, **dst-mac**, **src-dst-ip**, **src-dst-mac**, **src-dst-mixed-ip-port**, **src-ip**, or **src-mac**. However, this change disables the SD-WAN-based load balancing for the transport side.

Here's the complete configuration for enabling load balancing and apply the desired hash algorithm for traffic distribution on the transport side of Cisco IOS XE Catalyst SD-WAN devices.

```
port-channel load-balance-hash-algo sdwan
sdwan
 ip load-sharing algorithm src-dst-ip

port-channel load-balance-hash-algo sdwan
sdwan
 ipv6 load-sharing algorithm src-dst-ip
```

This example shows configuration enabling load balancing for each port channel interface. When **sdwan** hash algorithm is configured on the transport side, you can enable different hash algorithm options on the service side.

```
device(config)# interface Port-channel 1
device(config-if)# load-balance-hash-algo sdwan
device(config-if)# exit

device(config)# interface Port-channel 2
device(config-if)# load-balance-hash-algo src-dst-mixed-ip-port
device(config-if)# exit

device(config)# interface Port-channel 3
device(config-if)# no shut
device(config-if)# commit
device(config-if)# end
```

The following is a sample output to view the configuration for per-interface port channel using **show etherchannel load-balancing** command.

```
device# show etherchannel load-balancing
flow-based
LB Algo type: Source Destination IP

 Port-Channel:                       LB Method
    Port-channel1                   :  flow-based (SDWAN Inner packet LB)
    Port-channel2                   :  flow-based (Source Destination Port, IP addr)
    Port-channel3                   :  flow-based (Source Destination IP)
```

Monitor Configured EtherChannel Using CLI

Example 1

The following is a sample output from the **show etherchannel summary** command. This example shows summary for each channel group.

```
Device# show etherchannel summary

Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1       Po1(RU)         LACP     Te0/3/0(bndl) Te0/3/1(hot-sby)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

Example 2

The following is a sample output from the **show etherchannel load-balancing** command. This example displays the load-balancing method that is applied to each port channel.

```
Device# show etherchannel load-balancing

EtherChannel Load-Balancing Method:
Global LB Method: flow-based
LB Algo type: SDWAN Inner packet LB

 Port-Channel:                       LB Method
    Port-channel1                   :  flow-based (SDWAN Inner packet LB)
```

Aggregate EtherChannel Quality of Service

The Aggregate EtherChannel Quality of Service (QoS) feature improves the quality of service by effectively managing various network parameters, such as delay, jitter (or delay variation), bandwidth, and packet loss. Its primary function is to offer improved services for specific types of network traffic. The feature allows the application of an aggregate egress-queuing policy-map on the main or sub-interface of a port channel. Furthermore, it facilitates QoS support on the aggregate port channel's main interface on Cisco IOS XE Catalyst SD-WAN devices.

Prerequisites for Aggregate EtherChannel Quality of Service

- Identify aggregate port channel interfaces before creating them using the **platform qos port-channel-aggregate** command.
- In a port channel, all member links must be of the same speed.

Restrictions for Aggregate EtherChannel Quality of Service

- The aggregate port channel can support four member links and eight aggregate port channel interfaces.
- You can apply a policy map to the aggregate a port channel's main interface or sub-interface only. Member link QoS is not supported.
- You cannot spontaneously convert port channels to and from the aggregate status. You must delete the interface port-channel from the configurations before adding or removing the matching **platform qos port-channel-aggregate** command.
- QoS applications which are used to manage, prioritize and control the behavior of data transmission over a network are not supported on port channel member links.

  QoS policies applied to aggregate port channel main interfaces and port channel sub-interfaces are not supported.
- When you enable aggregate QoS, it is not possible to directly modify a channel group on a member link. To make changes, the old channel group needs to be removed and the new one must be added. First push one template to remove the old member link and port channel configuration, then another template to add the new configuration.

Configure Aggregate EtherChannel Quality of Service Using a CLI Template

In Cisco Catalyst SD-WAN Manager, you can configure aggregate EtherChannel QoS using the 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.

1. Create the aggregated port channel.

```
platform qos port-channel-aggregate port-channel-number
 interface Port-channel channel-number
 no shutdown
 ip address ip-address mask
```

2. Assign member links to port channel.

```
interface GigabitEthernet slot/subslot/port
 no negotiation auto
 channel-group channel-group-number mode {active passive}
 exit
```

3. Configure tunnels.

```
interface Tunnel tunnel-number
 no shutdown
 ip unnumbered port-channel-interface
 tunnel source port-channel-interface
 tunnel mode sdwan

sdwan
 interfacechannel-group-number
  tunnel-interface
   encapsulation ipsec
   color public-internet
```

4. Configure QoS.

```
interface channel-group-number
 service-policy output pre-defined qos policy-map
```

Here's the complete configuration example for configuring aggregate EtherChannel QoS.

```
!
class-map match-any Best-Effort
 match qos-group 2
!
class-map match-any Bulk
 match qos-group 3
!
class-map match-any Business
 match qos-group 1
!
class-map match-any Critical
 match qos-group 0
!
policy-map qos_template
 class Critical
  police rate percent 15
   !
  priority level 1
 !
 class Business
  bandwidth remaining percent 55
 !
 class Best-Effort
  bandwidth remaining percent 10
 !
 class Bulk
  bandwidth remaining percent 20
 !
!
policy-map shape_Port-channel1
 class class-default
  service-policy qos_template
  shape average 100000000
 !
!
interface TenGigabitEthernet0/1/6
 no shutdown
 no negotiation auto
 channel-group 1 mode active
 lacp rate fast
exit
interface TenGigabitEthernet0/1/7
 no shutdown
 no negotiation auto
 channel-group 1 mode active
 lacp rate fast
exit
interface Port-channel1
 no shutdown
 ip address 10.1.15.15 255.255.255.0
 ipv6 nd ra suppress all
 service-policy output shape_Port-channel1
exit
interface Tunnel1
 no shutdown
 ip unnumbered Port-channel1
 tunnel source Port-channel1
 tunnel mode sdwan
exit
!
sdwan
 interface Port-channel1
  tunnel-interface
   encapsulation ipsec
   color lte
   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
 exit
```

Verify Aggregate EtherChannel Quality of Service

To view QoS issues on a port channel interface, use the **show policy-map interface Port-channel** command.

```
Device# show policy-map interface Port-channel 1
Port-channel1

  Service-policy output: shape_Port-channel1

    Class-map: class-default (match-any)
      121 packets, 20797 bytes
      5 minute offered rate 2000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 416 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 121/20797
      shape (average) cir 100000000, bc 400000, be 400000
      target shape rate 100000000
```

```
Service-policy : qos_template

  queue stats for all priority classes:
    Queueing
    priority level 1
    queue limit 512 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 121/20797

  Class-map: Critical (match-any)
    121 packets, 20797 bytes
    5 minute offered rate 2000 bps, drop rate 0000 bps
    Match: qos-group 0
    police:
        rate 15 %
        rate 15000000 bps, burst 468750 bytes
      conformed 121 packets, 20797 bytes; actions:
        transmit
      exceeded 0 packets, 0 bytes; actions:
        drop
      conformed 2000 bps, exceeded 0000 bps
    Priority: Strict, b/w exceed drops: 0

    Priority Level: 1

  Class-map: Business (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 1
    Queueing
    queue limit 416 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    bandwidth remaining 55%

  Class-map: Best-Effort (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 2
    Queueing
    queue limit 416 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    bandwidth remaining 10%

  Class-map: Bulk (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: qos-group 3
    Queueing
    queue limit 416 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    bandwidth remaining 20%

  Class-map: class-default (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any

    queue limit 416 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
```