EtherChannels

Feature history for EtherChannels

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

Table 1: Feature history

Feature Name	Release Information	Description
EtherChannels on the service side	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. EtherChannels provide fault-tolerant, high-speed links, redundancy, and increased bandwidth between Cisco Catalyst SD-WAN devices and other network equipment. You can configure EtherChannels only using the CLI device templates and CLI add-on feature templates.
EtherChannels on the transport side	Cisco IOS XE Catalyst SD-WAN Release 17.13.1aCisco Catalyst SD-WAN Manager Release 20.13.1	This feature adds support for configuring EtherChannels on the transport side of a Cisco IOS XE Catalyst SD-WAN devices. This feature also enables aggregate EtherChannel Quality of Service (QoS) on the transport side, optimizing network utilization and performance for specific traffic types. **Note** This feature has limited availability.

Feature Name	Release Information	Description
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 allows you to configure load balancing for EtherChannels on the transport side for Cisco IOS XE Catalyst SD-WAN devices.
Configuration Groups for EtherChannels	Cisco IOS XE Catalyst SD-WAN Release 17.15.x Cisco Catalyst SD-WAN Manager Release 20.15.1	This feature allows you to configure EtherChannels on both service and transport sides using configuration groups in Cisco SD-WAN Manager.
Load balancing for EtherChannels on individual port channels	Cisco IOS XE Catalyst SD-WAN Release 17.15.x 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.

EtherChannels on the service side

EtherChannels on the service side is a capability that

- extends EtherChannel functionality to the service-facing interfaces of Cisco IOS XE Catalyst SD-WAN device s,
- provides fault-tolerant, high-speed links, redundancy, and increased bandwidth for service-side connections, and
- is configurable through CLI device templates, CLI add-on feature templates, and configuration groups.

EtherChannels on the service side

EtherChannels on the service side is a capability that

- extends EtherChannel functionality to the service-facing interfaces of Cisco IOS XE Catalyst SD-WAN device s,
- provides fault-tolerant, high-speed links, redundancy, and increased bandwidth for service-side connections, and
- is configurable through CLI device templates, CLI add-on feature templates, and configuration groups.

Supported devices for service side EtherChannel

This section provides a list of Cisco platforms that support EtherChannel functionality, including load balancing, on the service side. This information is crucial for planning and deploying Cisco Catalyst SD-WAN solutions that leverage EtherChannels for enhanced bandwidth and redundancy on service-facing interfaces.

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

Any L3 Ethernet interface on Network Interface Modules (NIMs) or Service Modules (SMs) can support EtherChannels on the service side.

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

Prerequisites for EtherChannels on the service side

Before configuring EtherChannels on the service side, ensure that the following prerequisties are met. These prerequisites are essential for your EtherChannel configuration.

- 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 port channel must have the same speed and duplex, when using platforms that support multiple rate SFPs on the same port.

Restrictions for EtherChannels on the service side

Maximum port channels

The maximum number of port channel interfaces that a device can support varies, depending on the particular model of the device.

Port channel configuration

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.

Hardware and interface compatibility

- 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.

Load balancing for service side EtherChannels using CLI commands

This topic describes the CLI commands and methods for configuring load balancing on service side EtherChannels, including enabling load balancing on individual port channels, configuring global and per-port-channel flow-based hash algorithms, and enabling VLAN load balancing.

The following tasks detail how to configure load balancing options for service side EtherChannels using the CLI commands:

- Enable load balancing on an individual port channel, on page 4
- Enable hash algorithms for flow-based load balancing on a global level, on page 6
- Enable hash algorithms flow-based load balancing on an individual port channel interface, on page 6
- Enable VLAN load balancing per port channel on the service side, on page 7
- Configure load balancing for EtherChannels on the service side using CLI commands, on page 8

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.

Enable load balancing on an individual port channel

Use this procedure to apply load balancing on an individual port channel.

Before you begin

You must be in global configuration mode and have an existing port channel created.

Complete these steps to enable load balancing on an individual port channel.

Procedure

Step 1 Enter the port channel interface configuration mode.

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

Step 2 Enable load balancing on an individual port channel.

```
load-balancing flow
```

The specified port channel now uses the configured load balancing method, overriding any global settings.

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

Use this procedure to configure a specific hash algorithm for flow-based load balancing globally across all port channels on the device.

Before you begin

Complete these steps to enable hash algorithms for flow-based load balancing on a global level.

Procedure

Configure the desired flow-based hash algorithm globally.

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

Replace *hash-algo* with one of the following supported hash algorithms:

dst-ip; *dst-macsrc-dst-ipsrc-dst-ipsrc-dst-macsrc-dst-mixed-ip-portsrc-ipsrc-mac*

The selected hash algorithm is now applied globally for flow-based load balancing on all port channels.

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 hash algorithms flow-based load balancing on an individual port channel interface

Use this procedure to configure a flow-based hash algorithm for load balancing on an individual port channel interface

Before you begin

This feature is supported from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a and Cisco Catalyst SD-WAN Manager Release 20.15.1.

Complete these steps to enable flow-based load balancing on an individual port channel interface.

Procedure

Step 1 Enter the port channel interface configuration mode.

```
interface Port-channel
```

Step 2 Enable flow-based load balancing hash algorithm.

```
load-balance-hash-algo hash-algo
```

Replace *hash-algo* with one of the following supported hash algorithms:

dst-ip; *dst-macsrc-dst-ipsrc-dst-ipsrc-dst-macsrc-dst-mixed-ip-portsrc-ipsrc-mac*

The specified port channel interface now uses the configured flow-based load balancing hash algorithm.

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

Use this procedure to configure VLAN-based load balancing on a specific EtherChannel port channel on the service side.

Before you begin

Complete these steps to enable VLAN load balancing per port channel on the service side.

Procedure

Step 1 Enter the port channel interface configuration mode.

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

Step 2 Enable vlan on per port channel.

```
load-balancing vlan
```

The specified service side port channel is now configured to use VLAN-based load balancing.

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.

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
```

Configure load balancing for EtherChannels on the service side using CLI commands

Use this procedure to configure load balancing for EtherChannels on the service side of your Cisco IOS XE Catalyst SD-WAN devices, ensuring efficient traffic distribution across bundled links.

Load balancing optimizes the use of aggregated bandwidth provided by EtherChannels and enhances network performance. You can choose between flow-based or VLAN-based methods and apply configurations globally or on individual port channels, depending on your network requirements.

Before you begin

Complete these steps to configure load balancing for EtherChannels on the service side using CLI commands.

Procedure

Decide on the load balancing method you want to implement, flow-based or VLAN-based:

For flow-based:

- To configure load balancing on a global level or per port channel, seee Enable load balancing on an individual port channel, on page 4.
- To enable a specific hash algorithm for flow-based load balancing across all port channels, see Enable hash algorithms for flow-based load balancing on a global level, on page 6.
- To enable a specific flow-based hash algorithm on an individual port channel interface, see Enable hash algorithms flow-based load balancing on an individual port channel interface, on page 6.

For VLAN-based:

If you choose VLAN-based load balancing, configure it per port channel, see Enable VLAN load balancing per port channel on the service side, on page 7.

Load balancing is now configured for your service side EtherChannels according to your chosen method.

What to do next

Verify your load balancing configuration using the show etherchannel load-balancing command.

Configure a service-side port channel and member link interface using configuration groups

Use this procedure to configure EtherChannel port channels and their member links on the service side using configuration groups within Cisco Catalyst SD-WAN Manager.

Before you begin

Complete these steps to configure a service-side port channel and member link interface using configuration groups in Cisco Catalyst SD-WAN Manager.

Procedure

Step 1 From the Cisco SD-WAN Manager menu, choose **Configuration** > **Configuration Groups**.

Step 2 Under **Service Profile**, click **Add New** to add a new service profile or select any existing service profiles, and click **Edit**.

Step 3 Select +, and click on the **Ethernet Interface** to add a new interface.

See Ethernet Interface for more details.

Step 4 Select **Add New** from the Ethernet Interface drop-down menu.

Step 5 Under **Basic Configurations**, enter the interface name and description.

Step 6 Click **Ether Channel**, and assign the EtherChannel to a **member interface** or a **port channel** interface from the drop down menu.

Note
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.

Step 7 Click **Save**.

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

A service-side port channel and its member links are configured using configuration groups, ready for deployment to your devices.

What to do next

Monitor the status of the deployed EtherChannels and verify traffic flow on the service side. For more information, see Monitor configured EtherChannel using CLI, on page 19.

EtherChannels on the transport side

EtherChannels on the transport side is a capability that

- extends EtherChannel functionality to the transport-facing interfaces of Cisco IOS XE Catalyst SD-WAN devices
- enables advanced network services like load balancing for aggregated links.
- is configurable through CLI device templates, CLI add-on feature templates, and configuration groups.

Supported devices for transport side EtherChannel

This section provides a comprehensive list of Cisco platforms that support EtherChannel functionality on the transport side. This information is crucial for planning and deploying Cisco Catalyst SD-WAN solutions that leverage EtherChannels for enhanced connectivity and resilience on transport-facing interfaces. It also indicates which platforms support load balancing for these EtherChannels.

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

- 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 EtherChannels on the transport side

Before configuring EtherChannels on the transport side, ensure that the following prerequisties are met. These prerequisites are essential for your EtherChannel configuration.

- 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 EtherChannels on the transport side

Maximum port channel interfaces

The maximum number of port channel interfaces that a device can support varies depending on the particular model of the device.

Port channel configuration

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.

Hardware and platform compatibility

- Network Interface Modules (NIMs) with L2 ports do not support EtherChannels on the transport side.
- The use of port channel on virtual devices such as Cisco Catalyst 8000V is not supported.
- 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.

Deployment

- In a deployment involving an EtherChannel Link Aggregation Group (LAG) from a Cisco IOS XE Catalyst SD-WAN device to a multichassis LAG (MC-LAG) between two upstream paths, SLA-based Application-Aware Routing (AAR) forwarding can be inaccurate if the traffic load on the two upstream paths is not symmetric.
- Cisco IOS XE Catalyst SD-WAN Release 17.13.1a does not include support for an endpoint tracker on port-channel TLOCs.

Configure a transport side EtherChannels using a CLI template

Use this procedure to create a logical EtherChannel interface on the transport side of a Cisco IOS XE Catalyst SD-WAN device, bundling multiple physical links for increased bandwidth and redundancy.

This procedure describes how to configure EtherChannels on the transport side using CLI templates in Cisco SD-WAN Manager.

In Cisco 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 for Cisco IOS XE Catalyst SD-WAN devices.

.

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

Before you begin

Complete these steps to configure a transport side EtherChannel using a CLI template.

Procedure

Step 1 Configure a Layer 3 port channel.

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

Step 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
```

Step 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
```

A transport side EtherChannel is configured and operational, providing aggregated bandwidth and redundancy for your Cisco Catalyst SD-WAN transport connections.

What to do next

Verify the EtherChannel status using CLI commands like **show etherchannel summary** and **show etherchannel load-balancing**.

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 for Cisco IOS XE Catalyst SD-WAN devices.

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.

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)
```

Enable load balancing on individual portchannel interface on the transport side

Use this procedure to enable load balancing on a specific EtherChannel port channel interface on the transport side.

This method is recommended for configuring load balancing on the transport side, providing granular control over traffic distribution for individual port channels. This feature is supported from Cisco IOS XE Catalyst SD-WAN Release 17.15.1a and Cisco Catalyst SD-WAN Manager Release 20.15.1.

Before you begin

An EtherChannel port channel must already be configured on the transport side.

Complete these steps to enable load balancing on an individual port channel interface.

Procedure

Step 1 Enter the port channel interface configuration mode.

interface Portchannel *channel number*

Replace *channel number* with the number of your port channel.

Step 2 Enable load balancing on an individual port channel.

load-balance-hash-algo sdwan

The specified transport side port channel interface is now configured for SD-WAN load balancing.

What to do next

Verify the individual port channel load balancing configuration using the **show etherchannel load-balancing** command.

Enable load balancing globally for EtherChannels on the transport side

Use this procedure to enable load balancing globally for all EtherChannels on the transport side of your device.

This configuration applies a default load balancing method to all transport side EtherChannels for which no individual load balancing method is explicitly configured. This feature is supported from Cisco IOS XE Catalyst SD-WAN Release 17.14.1a and Cisco Catalyst SD-WAN Manager Release 20.14.1.

Before you begin

Complete these steps to enable load balancing globally for EtherChannels on the transport side.

Procedure

Enable load balancing globally for EtherChannels on the transport side.

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

SD-WAN load balancing is now enabled globally for transport side EtherChannels.

What to do next

Verify the global load balancing configuration using the **show etherchannel load-balancing** command.

Enable hash algorithms globally for EtherChannels on the transport side

Use this procedure to configure specific hash algorithms for IP and IPv6 load balancing globally for EtherChannels on the transport side.

This global configuration applies to all Cisco Catalyst SD-WAN tunnels. Changing these algorithms affects the load-sharing mechanism for other SD-WAN tunnel traffic as well. This feature is supported from Cisco IOS XE Catalyst SD-WAN Release 17.14.1a and Cisco Catalyst SD-WAN Manager Release 20.14.1.

Before you begin

Complete these steps to enable hash algorithms globally for EtherChannels on the transport side.

Procedure

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}
  ```

  - **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.

- 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}
  ```

  The options are the same as for IPv4 load balancing.

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.

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)
```

The specified IPv4 and IPv6 load-sharing algorithms are now applied globally for transport side EtherChannels.

What to do next

Verify the global load balancing configuration using the **show etherchannel load-balancing** command.

Monitor configured EtherChannel using CLI

This section provides CLI commands and their sample outputs for monitoring the status and configuration of EtherChannels. These commands allow you to verify the operational state, member links, and load balancing methods applied to your EtherChannels.

View EtherChannel summary

Use the **show etherchannel summary** command to display a summary of each configured 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
```

View EtherChannel load balancing configuration

Use the **show etherchannel load-balancing** command to display the load-balancing method 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) is a EtherChannel-related feature that

- improves quality of service on a port channel main interface or subinterface,

- effectively manages network parameters such as delay, jitter, bandwidth, and packet loss, and
- allows the application of an aggregate egress-queuing policy-map on the main or sub-interface of a port channel.

The Aggregate EtherChannel QoS facilitates QoS support on the aggregate port channel's main interface on Cisco IOS XE Catalyst SD-WAN device.

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

Aggregate port channel member and interface limits

The aggregate port channel can support four member links and eight aggregate port channel interfaces.

QoS policy application restrictions on aggregate port channels

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.

Limitations of aggregate port channel conversion

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.

Unsupported QoS applications on port channel member links

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.

Channel group modification process with aggregate QoS enabled

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 CLI Template

Configure aggregate EtherChannel Quality of Service using a CLI template.

In SD-WAN Manager, you can configure aggregate EtherChannel QoS using the CLI templates to manage bandwidth and prioritize traffic across bundled links.

Before you begin

For more informationabout 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 config mode.

Follow these steps to configure aggregate EtherChannel QoS using a CLI template.

Procedure

Step 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
```

Step 2 Assign member links to port channel.

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

Step 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
```

Step 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
```

What to do next

Verify aggregate EtherChannel QoS.

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
```