# Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The distributed Gigabit Ethernet and 10-Gigabit, 40-Gigabit, 100-Gigabit Ethernet architecture and features deliver network scalability and performance, while enabling service providers to offer high-density, high-bandwidth networking solutions designed to interconnect the router with other systems in POPs, including core and edge routers and Layer 2 and Layer 3 switches.

**Tip** You can programmatically configure and manage the Ethernet interfaces using `openconfig-ethernet-if.yang` and `openconfig-interfaces.yang` OpenConfig data models. To get started with using data models, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

### Feature History for Configuring Ethernet Interfaces

| Release | Modification |
|---|---|
| Release 3.7.2 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 40-Port Gigabit Ethernet Medium Queue and High Queue Line Cards (A9K-40GE-B and A9K-40GE-E)<br>• 4-Port 10-Gigabit Ethernet Medium Queue and High Queue Line Cards (A9K-4T-B and A9K-4T-E)<br>• 8-Port 10-Gigabit Ethernet Medium Queue and High Queue DX Line Cards (A9K-8T/4-B and A9K-8T/4-E) (2:1 oversubscribed) |

| | |
|---|---|
| Release 3.9.0 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 40-Port Gigabit Ethernet Low Queue Line Card (A9K-40GE-L)<br>• 4-Port 10-Gigabit Ethernet Low Queue Line Card (A9K-4T-L)<br>• 8-Port 10-Gigabit Ethernet Low Queue DX Line Card (A9K-8T/4-L) (2:1 oversubscribed)<br>• 8-Port 10-Gigabit Ethernet Low and High Queue Line Card (A9K-8T-L and A9K-8T-E)<br>• 2-Port 10-Gigabit Ethernet, 20-Port Gigabit Ethernet Medium Queue and High Queue Combination Line Cards (A9K-2T20GE-B and A9K-2T20GE-L)<br>Support for these features was added:<br>• Frequency Synchronization<br>• SyncE |
| Release 3.9.1 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 8-Port 10-Gigabit Ethernet Medium Queue Line Card (A9K-8T-B)<br>• 16-Port 10-Gigabit Ethernet SFP+ Line Card (A9K-16T/8-B and A9K-16T/8-B+AIP) |
| Release 4.0.1 | Support for Layer 2 statistics collection for performance monitoring on Layer 2 subinterfaces (EFPs) is added. |
| Release 4.1.0 | Support for Link Layer Discovery Protocol (LLDP) was added.<br>**Note**<br>LLDP is not supported under management interface for this platform. |
| Release 4.1.1 | Support was added for MAC address accounting feature. |
| Release 4.2.2 | Support for Unidirectional Link Routing (UDLR) was introduced. |

| | |
|---|---|
| Release 4.3.1 | Support was added on the Cisco ASR 9000 Series Router for these line cards:<br>• A9K-MOD80-SE<br>• A9K-MOD80-TR<br>• A9K-MOD160-SE<br>• A9K-MOD160-TR<br>Support was added on the Cisco ASR 9000 Series Router for these Modular Port Adaptors (MPAs):<br>• A9K-MPA-20X1GE<br>• A9K-MPA-4X10GE<br>• A9K-MPA-2X10GE<br>• A9K-MPA-8X10GE (supported only with MOD160 Line cards)<br>• A9K-MPA-2X40GE (supported only with MOD160 Line cards)<br>• A9K-MPA-1X40GE |
| Release 5.3.1 | Support for IRB with Provider Backbone Bridge (PBB). |
| Release 6.2.1 | Link Degrade Alarm in Case of Link Loss Changing by 2 dB from the Preset Value |

• Configuring Ethernet Interfaces, on page 3
• Prerequisites for Configuring Ethernet Interfaces, on page 5
• Information About Configuring Ethernet, on page 6
• How to Configure Ethernet, on page 29
• Configuration Examples for Ethernet, on page 54
• How to Configure Interfaces in Breakout Mode, on page 56

# Configuring Ethernet Interfaces

This module describes the configuration of Ethernet interfaces.

The distributed Gigabit Ethernet and 10-Gigabit, 40-Gigabit, 100-Gigabit Ethernet architecture and features deliver network scalability and performance, while enabling service providers to offer high-density, high-bandwidth networking solutions designed to interconnect the router with other systems in POPs, including core and edge routers and Layer 2 and Layer 3 switches.

**Tip** You can programmatically configure and manage the Ethernet interfaces using `openconfig-ethernet-if.yang` and `openconfig-interfaces.yang` OpenConfig data models. To get started with using data models, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

### Feature History for Configuring Ethernet Interfaces

| Release | Modification |
|---|---|
| Release 3.7.2 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 40-Port Gigabit Ethernet Medium Queue and High Queue Line Cards (A9K-40GE-B and A9K-40GE-E)<br>• 4-Port 10-Gigabit Ethernet Medium Queue and High Queue Line Cards (A9K-4T-B and A9K-4T-E)<br>• 8-Port 10-Gigabit Ethernet Medium Queue and High Queue DX Line Cards (A9K-8T/4-B and A9K-8T/4-E) (2:1 oversubscribed) |
| Release 3.9.0 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 40-Port Gigabit Ethernet Low Queue Line Card (A9K-40GE-L)<br>• 4-Port 10-Gigabit Ethernet Low Queue Line Card (A9K-4T-L)<br>• 8-Port 10-Gigabit Ethernet Low Queue DX Line Card (A9K-8T/4-L) (2:1 oversubscribed)<br>• 8-Port 10-Gigabit Ethernet Low and High Queue Line Card (A9K-8T-L and A9K-8T-E)<br>• 2-Port 10-Gigabit Ethernet, 20-Port Gigabit Ethernet Medium Queue and High Queue Combination Line Cards (A9K-2T20GE-B and A9K-2T20GE-L)<br>Support for these features was added:<br>• Frequency Synchronization<br>• SyncE |
| Release 3.9.1 | Support was added on the Cisco ASR 9000 Series Router for the following line cards:<br>• 8-Port 10-Gigabit Ethernet Medium Queue Line Card (A9K-8T-B)<br>• 16-Port 10-Gigabit Ethernet SFP+ Line Card (A9K-16T/8-B and A9K-16T/8-B+AIP) |
| Release 4.0.1 | Support for Layer 2 statistics collection for performance monitoring on Layer 2 subinterfaces (EFPs) is added. |
| Release 4.1.0 | Support for Link Layer Discovery Protocol (LLDP) was added.<br>**Note**<br>LLDP is not supported under management interface for this platform. |
| Release 4.1.1 | Support was added for MAC address accounting feature. |

| | |
|---|---|
| Release 4.2.2 | Support for Unidirectional Link Routing (UDLR) was introduced. |
| Release 4.3.1 | Support was added on the Cisco ASR 9000 Series Router for these line cards:<br>• A9K-MOD80-SE<br>• A9K-MOD80-TR<br>• A9K-MOD160-SE<br>• A9K-MOD160-TR<br>Support was added on the Cisco ASR 9000 Series Router for these Modular Port Adaptors (MPAs):<br>• A9K-MPA-20X1GE<br>• A9K-MPA-4X10GE<br>• A9K-MPA-2X10GE<br>• A9K-MPA-8X10GE (supported only with MOD160 Line cards)<br>• A9K-MPA-2X40GE (supported only with MOD160 Line cards)<br>• A9K-MPA-1X40GE |
| Release 5.3.1 | Support for IRB with Provider Backbone Bridge (PBB). |
| Release 6.2.1 | Link Degrade Alarm in Case of Link Loss Changing by 2 dB from the Preset Value |

# Prerequisites for Configuring Ethernet Interfaces

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Before configuring Ethernet interfaces, be sure that these tasks and conditions are met:

- Confirm that at least one of these line cards supported on the router is installed:
  - 2-Port 10-Gigabit Ethernet, 20-Port Gigabit Ethernet Combination line card (A9K-2T20GE-B and A9K-2T20GE-L)
  - 4-Port 10-Gigabit Ethernet line card (A9K-4T-L, -B, or -E)
  - 8-Port 10-Gigabit Ethernet DX line card (A9K-8T/4-L, -B, or -E)
  - 8-Port 10-Gigabit Ethernet line card (A9K-8T-L, -B, or -E)
  - 16-Port 10-Gigabit Ethernet SFP+ line card (A9K-16T/8-B and A9K-16T/8-B+AIP)
  - 40-Port Gigabit Ethernet line card (A9K-40GE-L, -B, or -E)
  - 24-Port 10-Gigabit Ethernet Line Card

- 36-Port 10-Gigabit Ethernet Line Card
- 2-Port 100-Gigabit Ethernet Line Card
- ASR 9000 Mod80 Modular Line Card, Service Edge Optimized with modular port adapters
- ASR 9000 Mod80 Modular Line Card, Packet Transport Optimized with modular port adapters
- ASR 9000 Mod160 Modular Line Card, Service Edge Optimized with modular port adapters
- ASR 9000 Mod160 Modular Line Card, Packet Transport Optimized with modular port adapters

- Know the interface IP address.
- You know how to apply the specify the generalized interface name with the generalized notation *rack/slot/module/port*.

# Information About Configuring Ethernet

Ethernet is defined by the IEEE 802.3 international standard. It enables the connection of up to 1024 nodes over coaxial, twisted-pair, or fiber-optic cable.

The Cisco ASR 9000 Series Router supports Gigabit Ethernet (1000 Mbps),10-Gigabit Ethernet (10 Gbps), 40-Gigabit Ethernet (40 Gbps), and 100-Gigabit Ethernet (100 Gbps) interfaces.

### Limitation for Configuring Ethernet

- Cisco IOS XR software counts IPv4 directed broadcast packets with the destination MAC address **ffff.ffff.ffff** on the main interface but does not count them on ingress sub-interfaces, including both physical and bundle sub-interfaces, on the fifth generation of the ASR 9000 series Ethernet line cards.

This section provides the following information sections:

# 16-Port 10-Gigabit Ethernet SFP+ Line Card

The 16-Port10-Gigabit Ethernet SFP+ line card is a Small Form Factor (SFP transceiver) optical line card introduced in Cisco IOS XR Release 3.9.1 on the Cisco ASR 9000 Series Router. The 16-Port10-Gigabit Ethernet SFP+ line card supports all of the Gigabit Ethernet commands and configurations currently supported on the router.

The 16-Port10-Gigabit Ethernet SFP+ line card is compatible with all existing Cisco ASR 9000 Series Router line cards, route/switch processors (RSPs), and chassis.

## Features

The 16-Port10-Gigabit Ethernet SFP+ line card supports these features:

- 16 10-Gigabit Ethernet ports
- 128 10-Gigabit Ethernet ports per system
- 1.28 Tbps per system
- 160 Gbps forwarding

- 120 Gbps bidirectional performance
- SR/LR/ER SFP+ optics
- Feature parity with existing line cards
- Unicast and multicast forwarding at 160 Gbps, with zero packet loss during RSP switchover

**Restrictions**

The following features are not supported on the 16-Port10-Gigabit Ethernet SFP+ line card:

- DWDM (G.709)

# Cisco ASR 9000 Modular Line Cards

The Cisco ASR 9000 Series modular line cards provides a flexible solution to support multiple combinations of Ethernet ports, all in a single slot of the Cisco ASR 9000 Series Aggregation Services Routers. Modular line cards support a wide range of interfaces and densities offering the benefits of network scalability with lower initial costs and ease of upgrades.

The Cisco ASR 9000 Series modular line cards are designed for the Cisco ASR9000 Series Router which accepts pluggable modules. It allows you to cost effectively address lower density Gigabit Ethernet,10-Gigabit Ethernet, and 40-Gigabit Ethernet traffic. This line card is developed based on the ASR 9000 Enhanced Ethernet Network Processor (NP) and allows you to configure different interface types and also conserve chassis slots.

The Cisco ASR 9000 Series modular line cards accept two Ethernet Plugs (EP). Each Ethernet Plug provides optics, and support circuitry in order to provide GE, 10GE or 40GE ports.

The two versions of Modular Line Cards are:

- Cisco ASR 9000 Mod80 Modular Line Card – 2 ASR 9000 Enhanced Ethernet Network Processors (NP) which supports 2 pluggable Ethernet Plugs(EP), and 1 NP for each EP.
- Cisco ASR 9000 Mod160 Modular Line Card – 4 ASR 9000 Enhanced Ethernet Network Processors which supports 2 pluggable Ethernet Plugs, and 2 NPs for each EP.

**Note** A9K-MPA-20X1GE supports a speed of 10Mbps or 100Mbps when using only GLC-TE optics, regardless of MOD models.

## Restrictions on Module Port Adaptors

The two MPAs, A9K-MPA-8X10GE and A9K-MPA-2X40GE are supported only in A9K-MOD160-SE and A9K-MOD160-TR Line cards.

These are the specifications of the MPAs:

- A9K-MPA-8X10GE is supported only on the 160 Gigabyte Modular Line Card.
- A9K-MPA-8X10GE is not supported on 80 Gigabyte Modular Line Card.
- A9K-MPA-8X10GE is not supported on the Cisco ASR 9001 Chassis.

- A9K-MPA-8X10GE uses SFP+ Optics. The supported optics are SFP+ LR, SFP+ SR and SFP+ DWDM optics.

All other MPAs are supported in both the flavors of A9K-MOD80-SE/TR and A9K-MOD160-SE/TR Line Cards. For more information on these line cards, see *Cisco ASR 9000 Series Aggregation Services Router Ethernet Line Card Installation Guide* and *Cisco ASR 9000 Series Aggregation Services Router Overview and Reference Guide*.

# Default Configuration Values for Gigabit Ethernet and 10-Gigabit Ethernet

This table describes the default interface configuration parameters that are present when an interface is enabled on a Gigabit Ethernet or 10-Gigabit Ethernet modular services card and its associated PLIM.

**Note** You must use the **shutdown** command to bring an interface administratively down. The interface default is **no shutdown**. When a modular services card is first inserted into the router, if there is no established preconfiguration for it, the configuration manager adds a shutdown item to its configuration. This shutdown can be removed only be entering the **no shutdown** command.

***Table 1: Gigabit Ethernet and 10-Gigabit Ethernet Modular Services Card Default Configuration Values***

| Parameter | Configuration File Entry | Default Value |
|---|---|---|
| MAC accounting | **mac-accounting** | **off** |
| Flow control | **flow-control** | egress on ingress off |
| MTU | **mtu** | • 1514 bytes for normal frames<br>• 1518 bytes for 802.1Q tagged frames.<br>• 1522 bytes for Q-in-Q frames. |
| MAC address | **mac address** | Hardware burned-in address (BIA) |

# Default Configuration Values for Fast Ethernet

***Table 2: Fast Ethernet Default Configuration Values***

| Parameter | Configuration File Entry | Default Value |
|---|---|---|
| MAC accounting | **mac-accounting** | **off** |
| Duplex operation | **duplex full**<br>**duplex half** | Auto-negotiates duplex operation |

| Parameter | Configuration File Entry | Default Value |
|---|---|---|
| MTU | **mtu** | 1500 bytes |
| Interface speed | **speed** | 100 Mbps |
| Auto-negotiation | **negotiation auto** | **disable** |

# Layer 2 VPN on Ethernet Interfaces

Layer 2 Virtual Private Network (L2VPN) connections emulate the behavior of a LAN across an L2 switched, IP or MPLS-enabled IP network, allowing Ethernet devices to communicate with each other as if they were connected to a common LAN segment.

The L2VPN feature enables service providers (SPs) to provide Layer 2 services to geographically disparate customer sites. Typically, an SP uses an access network to connect the customer to the core network. On the Cisco ASR 9000 Series Router, this access network is typically Ethernet.

Traffic from the customer travels over this link to the edge of the SP core network. The traffic then tunnels through an L2VPN over the SP core network to another edge router. The edge router sends the traffic down another attachment circuit (AC) to the customer's remote site.

On the Cisco ASR 9000 Series Router, an AC is an interface that is attached to an L2VPN component, such as a bridge domain, pseudowire, or local connect.

The L2VPN feature enables users to implement different types of end-to-end services.

Cisco IOS XR Software supports a point-to-point end-to-end service, where two Ethernet circuits are connected together. An L2VPN Ethernet port can operate in one of two modes:

- Port Mode—In this mode, all packets reaching the port are sent over the PW (pseudowire), regardless of any VLAN tags that are present on the packets. In VLAN mode, the configuration is performed under the l2transport configuration mode.
- VLAN Mode—Each VLAN on a CE (customer edge) or access network to PE (provider edge) link can be configured as a separate L2VPN connection (using either VC type 4 or VC type 5). In VLAN mode, the configuration is performed under the individual subinterface.

**Note** The system sets a limit of 24K single vlan tags per NP and a 64K LC limit on the following line cards:

- A9K-MOD400-SE
- A9K-MOD400-CM
- A9K-MOD200-SE/CM
- Cisco ASR 9000 Series 24-port and 48-port dual-rate 10GE/1GE SE/CM
- A9K-8x100 SE/CM
- A99-8x100 SE/CM

Switching can take place in three ways:

- AC-to-PW—Traffic reaching the PE is tunneled over a PW (and conversely, traffic arriving over the PW is sent out over the AC). This is the most common scenario.
- Local switching—Traffic arriving on one AC is immediately sent out of another AC without passing through a pseudowire.
- PW stitching—Traffic arriving on a PW is not sent to an AC, but is sent back into the core over another PW.

Keep the following in mind when configuring L2VPN on an Ethernet interface:

- L2VPN links support QoS (Quality of Service) and MTU (maximum transmission unit) configuration.
- If your network requires that packets are transported transparently, you may need to modify the packet's destination MAC (Media Access Control) address at the edge of the Service Provider (SP) network. This prevents the packet from being consumed by the devices in the SP network.

Use the **show interfaces** command to display AC and PW information.

To configure a point-to-point pseudowire xconnect on an AC, refer to these documents:

- *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide.*
- *Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*

To attach Layer 2 service policies, such as QoS, to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

# Gigabit Ethernet Protocol Standards Overview

The Gigabit Ethernet interfaces support the following protocol standards:

These standards are further described in the sections that follow.

## IEEE 802.3 Physical Ethernet Infrastructure

The IEEE 802.3 protocol standards define the physical layer and MAC sublayer of the data link layer of wired Ethernet. IEEE 802.3 uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access at a variety of speeds over a variety of physical media. The IEEE 802.3 standard covers 10 Mbps Ethernet. Extensions to the IEEE 802.3 standard specify implementations for Gigabit Ethernet, 10-Gigabit Ethernet, and Fast Ethernet.

## IEEE 802.3ab 1000BASE-T Gigabit Ethernet

The IEEE 802.3ab protocol standards, or Gigabit Ethernet over copper (also known as 1000BaseT) is an extension of the existing Fast Ethernet standard. It specifies Gigabit Ethernet operation over the Category 5e/6 cabling systems already installed, making it a highly cost-effective solution. As a result, most copper-based environments that run Fast Ethernet can also run Gigabit Ethernet over the existing network infrastructure to dramatically boost network performance for demanding applications.

## IEEE 802.3z 1000 Mbps Gigabit Ethernet

Gigabit Ethernet builds on top of the Ethernet protocol, but increases speed tenfold over Fast Ethernet to 1000 Mbps, or 1 Gbps. Gigabit Ethernet allows Ethernet to scale from 10 or 100 Mbps at the desktop to 100 Mbps up to 1000 Mbps in the data center. Gigabit Ethernet conforms to the IEEE 802.3z protocol standard.

By leveraging the current Ethernet standard and the installed base of Ethernet and Fast Ethernet switches and routers, network managers do not need to retrain and relearn a new technology in order to provide support for Gigabit Ethernet.

## IEEE 802.3ae 10 Gbps Ethernet

Under the International Standards Organization's Open Systems Interconnection (OSI) model, Ethernet is fundamentally a Layer 2 protocol. 10-Gigabit Ethernet uses the IEEE 802.3 Ethernet MAC protocol, the IEEE 802.3 Ethernet frame format, and the minimum and maximum IEEE 802.3 frame size. 10 Gbps Ethernet conforms to the IEEE 802.3ae protocol standards.

Just as 1000BASE-X and 1000BASE-T (Gigabit Ethernet) remained true to the Ethernet model, 10-Gigabit Ethernet continues the natural evolution of Ethernet in speed and distance. Because it is a full-duplex only and fiber-only technology, it does not need the carrier-sensing multiple-access with the CSMA/CD protocol that defines slower, half-duplex Ethernet technologies. In every other respect, 10-Gigabit Ethernet remains true to the original Ethernet model.

## IEEE 802.3ba 100 Gbps Ethernet

IEEE 802.3ba is supported on the Cisco 1-Port 100-Gigabit Ethernet PLIM beginning in Cisco IOS XR 4.0.1.

# MAC Address

A MAC address is a unique 6-byte address that identifies the interface at Layer 2.

# MAC Accounting

The MAC address accounting feature provides accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces. This feature calculates the total packet and byte counts for a LAN interface that receives or sends IP packets to or from a unique MAC address. It also records a time stamp for the last packet received or sent.

These statistics are used for traffic monitoring, debugging and billing. For example, with this feature you can determine the volume of traffic that is being sent to and/or received from various peers at NAPS/peering points. This feature is currently supported on Ethernet, FastEthernet, and bundle interfaces and supports Cisco Express Forwarding (CEF), distributed CEF (dCEF), flow, and optimum switching.

**Note** A maximum of 512 MAC addresses per trunk interface are supported for MAC address accounting.

# Ethernet MTU

*Table 3: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Support for 9646 MTU size | Release 7.6.2 | This release supports an MTU size of 9646 on bundle and physical interfaces.<br>A larger MTU size helps send more data in a lower number of packets, thus enabling faster data transmission.<br>In earlier releases, the supported MTU size was 9216. |

The Ethernet maximum transmission unit (MTU) is the size of the largest frame, minus the 4-byte frame check sequence (FCS), that can be transmitted on the Ethernet network. Every physical network along the destination of a packet can have a different MTU.

Cisco IOS XR software supports two types of frame forwarding processes:

- Fragmentation for IPv4 packets–In this process, IPv4 packets are fragmented as necessary to fit within the MTU of the next-hop physical network.

**Note** IPv6 does not support fragmentation.

- MTU discovery process determines largest packet that size–This process is available for all IPv6 devices, and for originating IPv4 devices. In this process, the originating IP device determines the size of the largest IPv6 or IPv4 packet that can be sent without being fragmented. The largest packet is equal to the smallest MTU of any network between the IP source and the IP destination devices. If a packet is larger than the smallest MTU of all the networks in its path, that packet is fragmented as necessary. This process ensures that the originating device does not send an IP packet that is too large.

Jumbo frame support is automatically enabled for frames that exceed the standard frame size. The default value is 1514 for standard frames and 1518 for 802.1Q tagged frames. These numbers exclude the 4-byte frame check sequence (FCS).

You can set the data traffic with jumbo frames of a maximum size of 9646 bytes for the line cards that are mentioned in this section.

**Note** ASIC on 9000v considers all the packets greater than 1514 byte as oversized frame.

On the following line cards, you can configure a maximum MTU size of 9646.

- A99-32X100GE-CM
- A99-32X100GE-DENS

- A99-32X100GE-SE
- A99-32X100GE-TR
- A99-32X100GE-X-CM
- A99-32X100GE-X-SE
- A99-32X100GE-X-TR
- A99K-48x10GE-SE
- A99K-48x10GE-SE-TAA
- A99K-48x10GE-TAA
- A99K-48x10GE-TR
- A99K-48x10GE-TR-TAA
- A9K-20HG-FLEX-CM
- A9K-20HG-FLEX-SE
- A9K-20HG-FLEX-TR
- A9K-48x10GE-SE
- A9K-48x10GE-SE-TAA
- A9K-48x10GE-TAA
- A9K-48x10GE-TR
- A9K-48x10GE-TR-TAA
- A9K-8HG-FLEX-CM
- A9K-8HG-FLEX-SE
- A9K-8HG-FLEX-TR

When the MTU size is larger, more data can be sent in fewer number of packets. As each packet contains a header, a small number of packets means less header data to be transmitted. Thus, larger size MTU helps in transmitting more actual data faster and efficiently.

To determine an optimal MTU size for your network, you can use the ping test. The Ping command tests the reachability of devices in a network using the echo request and reply messages within the Internet Control Message Protocol (ICMP). Within the router, the supported maximum local ping size is 9580 for these LCs.

For the list of line cards mentioned in this section, the MTU size of the Cisco ASR 9000 platform control-packets is 9580.

For details of configuring the MTU size, see Configuring MTU, on page 48.

# Flow Control on Ethernet Interfaces

The flow control used on 10-Gigabit Ethernet interfaces consists of periodically sending flow control pause frames. It is fundamentally different from the usual full- and half-duplex flow control used on standard

management interfaces. Flow control can be activated or deactivated for ingress traffic only. It is automatically implemented for egress traffic.

# 802.1Q VLAN

A VLAN is a group of devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, it is very flexible for user and host management, bandwidth allocation, and resource optimization.

The IEEE's 802.1Q protocol standard addresses the problem of breaking large networks into smaller parts so broadcast and multicast traffic does not consume more bandwidth than necessary. The standard also helps provide a higher level of security between segments of internal networks.

The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.

# VRRP

The Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VPN concentrators on a LAN. The VRRP VPN concentrator controlling the IP addresses associated with a virtual router is termed as the primary concentrator, and forwards packets sent to those IP addresses. When the primary concentrator becomes unavailable, a backup VPN concentrator takes over.

For more information on VRRP, see the *Implementing VRRP* module of *Cisco ASR 9000 Series Router IP Addresses and Services Configuration Guide*.

# HSRP

Hot Standby Routing Protocol (HSRP) is a proprietary protocol from Cisco. HSRP is a routing protocol that provides backup to a router in the event of failure. Several routers are connected to the same segment of an Ethernet, FDDI, or token-ring network and work together to present the appearance of a single virtual router on the LAN. The routers share the same IP and MAC addresses and therefore, in the event of failure of one router, the hosts on the LAN are able to continue forwarding packets to a consistent IP and MAC address. The transfer of routing responsibilities from one device to another is transparent to the user.

HSRP is designed to support non disruptive switchover of IP traffic in certain circumstances and to allow hosts to appear to use a single router and to maintain connectivity even if the actual first hop router they are using fails. In other words, HSRP protects against the failure of the first hop router when the source host cannot learn the IP address of the first hop router dynamically. Multiple routers participate in HSRP and in concert create the illusion of a single virtual router. HSRP ensures that one and only one of the routers is forwarding packets on behalf of the virtual router. End hosts forward their packets to the virtual router.

The router forwarding packets is known as the *active router*. A standby router is selected to replace the active router should it fail. HSRP provides a mechanism for determining active and standby routers, using the IP addresses on the participating routers. If an active router fails a standby router can take over without a major interruption in the host's connectivity.

HSRP runs on top of User Datagram Protocol (UDP), and uses port number 1985. Routers use their actual IP address as the source address for protocol packets, not the virtual IP address, so that the HSRP routers can identify each other.

For more information on HSRP, see the *Implementing HSRP* module of *Cisco ASR 9000 Series Router Cisco IOS XR*

# Link Autonegotiation on Ethernet Interfaces

***Table 4: Feature History Table***

| Feature Name | Release Information | Description |
|---|---|---|
| Autonegotiation Support on Cisco ASR 9901 Router | Release 7.4.1 | This release introduces support for Autonegotiation feature on the 1GE fiber optic plugged in Cisco ASR 9901 Router's dual-rate (1/10 GE) interface.<br>This feature enables devices to automatically exchange, over a link, information about port speed and duplex modes.<br>To enable the auto negotiation on 1GE interface, use the negotiation auto command. |

Link autonegotiation ensures that devices that share a link segment are automatically configured with the highest performance mode of interoperation. Use the **negotiation auto** command in interface configuration mode to enable link autonegotiation on an Ethernet interface. On line card Ethernet interfaces, link autonegotiation is disabled by default.

**Note** The **negotiation auto** command is available on Gigabit Ethernet interfaces only.

This table describes the performance of the system for different combinations of the speed modes. The specified command produces the resulting system action, provided that you have configured autonegotiation on the interface.

***Table 5: Relationship Between duplex and speed Commands***

| duplex Command | speed Command | |
|---|---|---|
| **full-duplex** | **no speed** | Forces full duplex and auto-negotiates for speed. |
| **full-duplex** | **speed 1000** | Forces full duplex and 1000 Mbps. |
| **full-duplex** | **speed 100** | Forces full duplex and 100 Mbps. |
| **full-duplex** | **speed 10** | Forces full duplex and 10 Mbps. |
| **half-duplex** | **no speed** | Forces half duplex and auto-negotiates for speed. |
| **half-duplex** | **speed 1000** | Forces half duplex and 1000 Mbps. |

| duplex Command | speed Command | |
|---|---|---|
| **half-duplex** | **speed 100** | Forces half duplex and 100 Mbps. |
| **half-duplex** | **speed 10** | Forces half duplex and 10 Mbps. |

# Fast Polling for WAN-PHY

Fast Ethernet interfaces must be continuously monitored in order to detect any link that is not working due to BER errors (bit error rate) and to bring down the interface connected to that link. The Fast Polling feature polls interfaces at a fast rate and brings down the interface in case of BER errors, thereby minimizing service impact. The Fast Polling feature for WAN-PHY reduces convergence time to 150 ms on the ASR 9000 Enhanced Ethernet line card and the ASR 9000 High Density 100GE Ethernet line cards with 10G and higher rate ports.

To configure the fast polling for WAN-PHY on ASR 9000 Enhanced Ethernet line card, use the **wanphy poll-timer** *value-in-milliseconds* command in configuration mode.

The fast polling for WAN-PHY on ASR 9000 High Density 100GE Ethernet line card is enabled by default.

# Early Indication of Link Loss Change

This feature helps in early detection of a link loss between two devices and prevents any service impact. To enable this feature user must configure a receiving optical power threshold value. Whenever the receiving power crosses the threshold, the power degrade alarm is raised. The alram resets after the power exceeds threshold value by 2db.

For example, consider you have configured receiving optical power threshold value to -10db. When the input power on the interface reduces below -10db due to fiber degradation, then the optical power alarm is raised. After fiber degradation is attended power value will improve. Once the power value crosses above -8db, that is 2db more than threshold configured, then the alarm resets.

To configure the receiving power optical threshold value, use **optical-power alarm rx** <*value in db*> command in interface configuration mode.

Use this show command to view the alarm status:

- **show controllers <*ethernet-interface*> internal**

Use this command to view threshold value configured, and the minimum and maximum threshold value:

- **show controllers <*ethernet-interface*> control**

# Subinterfaces on the Cisco ASR 9000 Series Router

In Cisco IOS XR, interfaces are, by default, main interfaces. A main interface is also called a trunk interface, which is not to be confused with the usage of the word trunk in the context of VLAN trunking.

There are three types of trunk interfaces:

- Physical
- Bundle

On the Cisco ASR 9000 Series Router, physical interfaces are automatically created when the router recognizes a card and its physical interfaces. However, bundle interfaces are not automatically created. They are created when they are configured by the user.

The following configuration samples are examples of trunk interfaces being created:

- interface gigabitethernet 0/5/0/0
- interface bundle-ether 1

A subinterface is a logical interface that is created under a trunk interface.

To create a subinterface, the user must first identify a trunk interface under which to place it. In the case of bundle interfaces, if one does not already exist, a bundle interface must be created before any subinterfaces can be created under it.

The user then assigns a subinterface number to the subinterface to be created. The subinterface number must be a positive integer from zero to some high value. For a given trunk interface, each subinterface under it must have a unique value.

Subinterface numbers do not need to be contiguous or in numeric order. For example, the following subinterfaces numbers would be valid under one trunk interface:

1001, 0, 97, 96, 100000

Subinterfaces can never have the same subinterface number under one trunk.

In the following example, the card in slot 5 has trunk interface, GigabitEthernet 0/5/0/0. A subinterface, GigabitEthernet 0/5/0/0.0, is created under it.

```
RP/0/RSP0/CPU0:router#  conf
Mon Sep 21 11:12:11.722 EDT
RP/0/RSP0/CPU0:router(config)#  interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit

RP/0/RSP0/CPU0:Sep 21 11:12:34.819 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000152' to view the
changes.

RP/0/RSP0/CPU0:router(config-subif)# end

RP/0/RSP0/CPU0:Sep 21 11:12:35.633 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router#
```

The **show run** command displays the trunk interface first, then the subinterfaces in ascending numerical order.

```
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:15:42.654 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 shutdown
!
interface GigabitEthernet0/5/0/0.0
 encapsulation dot1q 100
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

When a subinterface is first created, the Cisco ASR 9000 Series Router recognizes it as an interface that, with few exceptions, is interchangeable with a trunk interface. After the new subinterface is configured further, the **show interface** command can display it along with its unique counters:

The following example shows the display output for the trunk interface, GigabitEthernet 0/5/0/0, followed by the display output for the subinterface GigabitEthernet 0/5/0/0.0.

```
RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/5/0/0
Mon Sep 21 11:12:51.068 EDT
GigabitEthernet0/5/0/0 is administratively down, line protocol is administratively down.
  Interface state transitions: 0
  Hardware is GigabitEthernet, address is 0024.f71b.0ca8 (bia 0024.f71b.0ca8)
  Internet address is Unknown
  MTU 1514 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN,
  Full-duplex, 1000Mb/s, SXFD, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions


RP/0/RSP0/CPU0:router# show interface gigabitEthernet0/5/0/0.0
Mon Sep 21 11:12:55.657 EDT
GigabitEthernet0/5/0/0.0 is administratively down, line protocol is administratively down.
  Interface state transitions: 0
  Hardware is VLAN sub-interface(s), address is 0024.f71b.0ca8
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 100,  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

This example shows two interfaces being created at the same time: first, the bundle trunk interface, then a subinterface attached to the trunk:

```
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 10:57:31.736 EDT
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1
```

```
RP/0/RSP0/CPU0:router(config-if)# no shut
RP/0/RSP0/CPU0:router(config-if)# interface bundle-Ether1.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:Sep 21 10:58:15.305 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : C
onfiguration committed by user 'root'. Use 'show configuration commit changes 10
00000149' to view the changes.
RP/0/RSP0/CPU0:router# show run | begin Bundle-Ether1
Mon Sep 21 10:59:31.317 EDT
Building configuration..
interface Bundle-Ether1
!
interface Bundle-Ether1.0
 encapsulation dot1q 100
!
```

You delete a subinterface using the **no interface** command.

```
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:42:27.100 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/0.0
 encapsulation dot1q 100
!
interface GigabitEthernet0/5/0/1
 shutdown
!
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 11:42:32.374 EDT
RP/0/RSP0/CPU0:router(config)# no interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:Sep 21 11:42:47.237 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000159' to view the
changes.
RP/0/RSP0/CPU0:router(config)# end
RP/0/RSP0/CPU0:Sep 21 11:42:50.278 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:42:57.262 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

## Layer 2, Layer 3, and EFP's

On the Cisco ASR 9000 Series Router, a trunk interface can be either a Layer 2 or Layer 3 interface. A Layer 2 interface is configured using the **interface** command with the **l2transport** keyword. When the **l2transport** keyword is not used, the interface is a Layer 3 interface. Subinterfaces are configured as Layer 2 or Layer 3 subinterface in the same way.

A Layer 3 trunk interface or subinterface is a routed interface and can be assigned an IP address. Traffic sent on that interface is routed.

A Layer 2 trunk interface or subinterface is a switched interface and cannot be assigned an IP address. A Layer 2 interface must be connected to an L2VPN component. Once it is connected, it is called an access connection.

Subinterfaces can only be created under a Layer 3 trunk interface. Subinterfaces cannot be created under a Layer 2 trunk interface.

A Layer 3 trunk interface can have any combination of Layer 2 and Layer 3 interfaces.

The following example shows an attempt to configure a subinterface under an Layer 2 trunk and the commit errors that occur. It also shows an attempt to change the Layer 2 trunk interface to an Layer 3 interface and the errors that occur because the interface already had an IP address assigned to it.

```
RP/0/RSP0/CPU0:router# config
Mon Sep 21 12:05:33.142 EDT
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.0.0.1/24
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:Sep 21 12:05:57.824 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000160' to view the
changes.
RP/0/RSP0/CPU0:router(config-if)# end
RP/0/RSP0/CPU0:Sep 21 12:06:01.890 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 12:06:19.535 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 ipv4 address 10.0.0.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/5/0/1
 shutdown
!
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 12:08:07.426 EDT
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed' from this session
 to view the errors
RP/0/RSP0/CPU0:router(config-if-l2)# no ipv4 address
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:Sep 21 12:08:33.686 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000161' to view the
changes.
RP/0/RSP0/CPU0:router(config-if)# end
RP/0/RSP0/CPU0:Sep 21 12:08:38.726 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run interface GigabitEthernet0/5/0/0
Mon Sep 21 12:09:02.471 EDT
interface GigabitEthernet0/5/0/0
 negotiation auto
 l2transport
 !
!
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 12:09:08.658 EDT
```

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0.0
                                                        ^
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# commit

% Failed to commit one or more configuration items during a pseudo-atomic operation. All
changes made have been reverted. Please issue 'show configuration failed' from this session
 to view the errors
RP/0/RSP0/CPU0:router(config-subif)#
RP/0/RSP0/CPU0:router(config-subif)# interface GigabitEthernet0/5/0/0
RP/0/RSP0/CPU0:router(config-if)# no l2transport
RP/0/RSP0/CPU0:router(config-if)# interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 99
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 11.0.0.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface GigabitEthernet0/5/0/0.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 700
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:Sep 21 12:11:45.896 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
 committed by user 'root'. Use 'show configuration commit changes 1000000162' to view the
changes.
RP/0/RSP0/CPU0:router(config-subif)# end
RP/0/RSP0/CPU0:Sep 21 12:11:50.133 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
 console by root
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run | b GigabitEthernet0/5/0/0
Mon Sep 21 12:12:00.248 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/0.0
 ipv4 address 11.0.0.1 255.255.255.0
 encapsulation dot1q 99
!
interface GigabitEthernet0/5/0/0.1 l2transport
 encapsulation dot1q 700
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

All subinterfaces must have unique encapsulation statements, so that the router can send incoming packets and frames to the correct subinterface. If a subinterface does not have an encapsulation statement, the router will not send any traffic to it.

In Cisco IOS XR, an Ethernet Flow Point (EFP) is implemented as a Layer 2 subinterface, and consequently, a Layer 2 subinterface is often called an EFP. For more information about EFPs, see the Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide.

A Layer 2 trunk interface can be used as an access connection. However, a Layer 2 trunk interface is not an EFP because an EFP, by definition, is a substream of an overall stream of traffic.

Cisco IOS XR also has other restrictions on what can be configured as a Layer 2 or Layer 3 interface. Certain configuration blocks only accept Layer 3 and not Layer 2. For example, OSPF only accepts Layer 3 trunks and subinterface. Refer to the appropriate Cisco IOS XR configuration guide for other restrictions.

## Enhanced Performance Monitoring for Layer 2 Subinterfaces (EFPs)

Beginning in Cisco IOS XR Release 4.0.1, the Cisco ASR 9000 Series Router adds support for basic counters for performance monitoring on Layer 2 subinterfaces.

This section provides a summary of the new support for Layer 2 interface counters. For information about how to configure Performance Monitoring, see the "Implementing Performance Management" chapter of the Cisco ASR 9000 Series Aggregation Services Router System Monitoring Configuration Guide.

The **interface basic-counters** keyword has been added to support a new entity for performance statistics collection and display on Layer 2 interfaces in the following commands:

- **performance-mgmt statistics interface basic-counters**
- **performance-mgmt threshold interface basic-counters**
- **performance-mgmt apply statistics interface basic-counters**
- **performance-mgmt apply threshold interface basic-counters**
- **performance-mgmt apply monitor interface basic-counters**
- show performance-mgmt monitor interface basic-counters
- show performance-mgmt statistics interface basic-counters

The **performance-mgmt threshold interface basic-counters** command supports the following attribute values for Layer 2 statistics, which also appear in the **show performance-mgmt statistics interface basic-counters** and **show performance-mgmt monitor interface basic-counters** command:

| Attribute | Description |
|---|---|
| InOctets | Bytes received (64-bit) |
| InPackets | Packets received (64-bit) |
| InputQueueDrops | Input queue drops (64-bit) |
| InputTotalDrops | Inbound correct packets discarded (64-bit) |
| InputTotalErrors | Inbound incorrect packets discarded (64-bit) |
| OutOctets | Bytes sent (64-bit) |
| OutPackets | Packets sent (64-bit) |
| OutputQueueDrops | Output queue drops (64-bit) |
| OutputTotalDrops | Outband correct packets discarded (64-bit) |
| OutputTotalErrors | Outband incorrect packets discarded (64-bit) |

## Other Performance Management Enhancements

The following additional performance management enhancements are included in Cisco IOS XR Release 4.0.1:

- You can retain performance management history statistics across a process restart or route processor (RP) failover using the new **history-persistent** keyword option for the **performance-mgmt statistics interface** command.
- You can save performance management statistics to a local file using the **performance-mgmt resources dump local** command.

- You can filter performance management instances by defining a regular expression group (**performance-mgmt regular-expression** command), which includes multiple regular expression indices that specify strings to match. You apply a defined regular expression group to one or more statistics or threshold templates in the **performance-mgmt statistics interface** or **performance-mgmt thresholds** interface commands.

# Frequency Synchronization and SyncE

Cisco IOS XR Software provides support for SyncE-capable Ethernet on the Cisco ASR 9000 Series Router. Frequency Synchronization provides the ability to distribute precision clock signals around the network. Highly accurate timing signals are initially injected into the Cisco ASR 9000 Series Router in the network from an external timing technology (such as Cesium atomic clocks, or GPS), and used to clock the physical interfaces of the router. Peer routers can then recover this precision frequency from the line, and also transfer it around the network. This feature is traditionally applicable to SONET/SDH networks, but is now provided over Ethernet for Cisco ASR 9000 Series Aggregation Services Routers with Synchronous Ethernet capability. For more information, see *Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*.

# LLDP

**Note** LLDP is not supported on the FP-X line cards.

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the Data Link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches). CDP allows network management applications to automatically discover and learn about other Cisco devices connected to the network.

To support non-Cisco devices and to allow for interoperability between other devices, the Cisco ASR 9000 Series Router also supports the IEEE 802.1AB LLDP. LLDP is also a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the Data Link Layer, which allows two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to learn information about neighbor devices. These attributes have a defined format known as a Type-Length-Value (TLV). LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

In addition to the mandatory TLVs (Chassis ID, Port ID, and Time-to-Live), the router also supports the following basic management TLVs, which are optional:

- Port Description
- System Name
- System Description
- System Capabilities
- Management Address

These optional TLVs are automatically sent when LLDP is active, but you can disable them as needed using the **lldp tlv-select disable** command.

## LLDP Frame Format

LLDP frames use the IEEE 802.3 format, which consists of the following fields:

- Destination address (6 bytes)—Uses a multicast address of 01-80-C2-00-00-0E.
- Source address (6 bytes)—MAC address of the sending device or port.
- LLDP Ethertype (2 bytes)—Uses 88-CC.
- LLDP PDU (1500 bytes)—LLDP payload consisting of TLVs.
- FCS (4 bytes)—Cyclic Redundancy Check (CRC) for error checking.

## LLDP TLV Format

LLDP TLVs carry the information about neighboring devices within the LLDP PDU using the following basic format:

- TLV Header (16 bits), which includes the following fields:
  - TLV Type (7 bits)
  - TLV Information String Length (9 bits)
- TLV Information String (0 to 511 bytes)

## Specifying User-Defined LLDP TLV Values

It is possible to override the system default values for some of the mandatory LLDP Type-Length-Values (TLVs) that are advertised by routers to their directly connected neighboring devices. While advertising their identity and capabilities, routers can assign user-defined meaningful names instead of autogenerated values. Using the following CLIs you can specify these user-defined values:

- `Router(config)#lldp system-name` *`system-name`*
- `Router(config)#lldp system-description` *`system-description`*
- `Router(config)#lldp chassis-id-type` *`chassis-type`*
- `Router(config)#lldp chassis-id` *`local-chassis-id`*

**Note** The **chassis-id** value is configurable only when the **chassis-id-type** is set as **Local**. If there is a mismatch, you encounter a configuration failed error message.

The configured values, such as the system name, system description, chassis-id, chassis-type become part of the TLV in the LLDP packets that are sent to its neighbors. Values are transmitted only to LLDP enabled interfaces to which the router is connected.

You can assign any of the following values for the `chassis-id-type`. The chassis-id-types are objects that are part of the management information base (MIB). Depending on the selected chassis-id-type, values are assigned to these objects, and they are advertised by the router to its neighboring devices.

| chassis-id-type | Description |
|---|---|
| chassis-component | Chassis identifier based on the value of entPhysicalAlias object that is defined in IETF RFC 2737. |
| interface-alias | Chassis identifier based on the value of ifAlias object as defined in IETF RFC 2863. |
| interface-name | Chassis identifier based on the name of the interface. |
| local | Chassis identifier based on a locally defined value. |
| mac-address | Chassis identifier based on the value of a unicast source address. |
| network-address | Chassis identifier based on a network address that is associated with a particular chassis. |
| port-component | Chassis identifier based on the value of entPhysicalAlias object defined in IETF RFC 2737 for a port or backplane component. |

**Tip** You can programmatically modify default values of LLDP TLVs by using the `openconfig-lldp` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

### Configuration Example

This example shows the configuration for the LLDP TLVs that will be advertised by routers to their directly connected neighboring devices.

```
Router(config)#lldp system-name cisco-xr
Router(config)#lldp system-description cisco-xr-edge-device
Router(config)#lldp chassis-id-type local
Router(config)#lldp chassis-id ce-device9
```

### Running Configuration

```
Router#show lldp
Tue Sep 13 16:03:44.550 +0530
Global LLDP information:
Status: ACTIVE
LLDP Chassis ID: ce-device9
LLDP Chassis ID Subtype: Locally Assigned Chassis Subtype
LLDP System Name: cisco-xr
LLDP advertisements are sent every 30 seconds
LLDP hold time advertised is 120 seconds
LLDP interface reinitialisation delay is 2 seconds
```

## LLDP Operation

LLDP is a one-way protocol. The basic operation of LLDP consists of a device enabled for transmit of LLDP information sending periodic advertisements of information in LLDP frames to a receiving device.

Devices are identified using a combination of the Chassis ID and Port ID TLVs to create an MSAP (MAC Service Access Point). The receiving device saves the information about a neighbor for a certain amount time specified in the TTL TLV, before aging and removing the information.

LLDP supports the following additional operational characteristics:

- LLDP can operate independently in transmit or receive modes.
- LLDP operates as a slow protocol using only untagged frames, with transmission speeds of less than 5 frames per second.
- LLDP packets are sent when the following occurs:
  - The packet update frequency specified by the **lldp timer** command is reached. The default is 30 seconds.
  - When a change in the values of the managed objects occurs from the local system's LLDP MIB.
  - When LLDP is activated on an interface (3 frames are sent upon activation similar to CDP).
- When an LLDP frame is received, the LLDP remote services and PTOPO MIBs are updated with the information in the TLVs.
- LLDP supports the following actions on these TLV characteristics:
  - Interprets a TTL value of 0 as a request to automatically purge the information of the transmitting device. These shutdown LLDPDUs are typically sent prior to a port becoming inoperable.
  - An LLDP frame with a malformed mandatory TLV is dropped.
  - A TLV with an invalid value is ignored.
  - A copy of an unknown organizationally-specific TLV is maintained if the TTL is non-zero, for later access through network management.

## Supported LLDP Functions

The Cisco ASR 9000 Series Router supports the following LLDP functions:

- IPv4 and IPv6 management addresses—In general, both IPv4 and IPv6 addresses will be advertised if they are available, and preference is given to the address that is configured on the transmitting interface.

  If the transmitting interface does not have a configured address, then the TLV will be populated with an address from another interface. The advertised LLDP IP address is implemented according to the following priority order of IP addresses for interfaces on the Cisco ASR 9000 Series Router:
  - Locally configured address
  - MgmtEth0/RSP0/CPU0/0
  - MgmtEth0/RSP0/CPU0/1
  - MgmtEth0/RSP1/CPU0/0

- MgmtEth0/RSP1/CPU0/1
- Loopback interfaces

  There are some differences between IPv4 and IPv6 address management in LLDP:

- For IPv4, as long as the IPv4 address is configured on an interface, it can be used as an LLDP management address.
- For IPv6, after the IPv6 address is configured on an interface, the interface status must be Up and pass the DAD (Duplicate Address Detection) process before it is can be used as an LLDP management address.
- LLDP is supported for the nearest physically attached, non-tunneled neighbors.
- Port ID TLVs are supported for Ethernet interfaces, subinterfaces, bundle interfaces, and bundle subinterfaces.

## Unsupported LLDP Functions

The following LLDP functions are not supported on the Cisco ASR 9000 Series Router:

- LLDP-MED organizationally unique extension—However, interoperability still exists between other devices that do support this extension.
- Tunneled neighbors, or neighbors more than one hop away.
- LLDP TLVs cannot be disabled on a per-interface basis; However, certain optional TLVs can be disabled globally.
- LLDP SNMP trap lldpRemTablesChange.

# Enabling LLDP Per Interface

When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations. However, if you want to enable LLDP per interface, perform the following configuration steps:

1. `RP/0/RSP0/CPU0:ios(config)# int gigabitEthernet 0/2/0/0`
2. `RP/0/RSP0/CPU0:ios(config-if)# no sh`
3. `RP/0/RSP0/CPU0:ios(config-if)#commit`
4. `RP/0/RSP0/CPU0:ios(config-if)#lldp ?`
5. `RP/0/RSP0/CPU0:ios(config-if)#lldp enable`
6. `RP/0/RSP0/CPU0:ios(config-if)#commit`

### Running configuration

```
RP/0/RSP0/CPU0:ios#sh running-config
Wed Jun 27 12:40:21.274 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Wed Jun 27 00:59:29 2018 by UNKNOWN
!
interface GigabitEthernet0/1/0/0
```

```
 shutdown
!
interface GigabitEthernet0/1/0/1
 shutdown
!
interface GigabitEthernet0/1/0/2
 shutdown
!
interface GigabitEthernet0/2/0/0
 Shutdown
!
interface GigabitEthernet0/2/0/1
 shutdown
!
interface GigabitEthernet0/2/0/2
 shutdown
!
end
```

### Verification

```
Verifying the config
==================
RP/0/RSP0/CPU0:ios#sh lldp interface <===== LLDP enabled only on GigEth0/2/0/0
Wed Jun 27 12:43:26.252 IST


GigabitEthernet0/2/0/0:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
RP/0/RSP0/CPU0:ios#

RP/0/RSP0/CPU0:ios# show lldp neighbors
Wed Jun 27 12:44:38.977 IST
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
ios             Gi0/2/0/0           120        R              Gi0/2/0/0       <====== LLDP
 enabled only on GigEth0/2/0/0 and neighborship seen for the same.

Total entries displayed: 1

RP/0/RSP0/CPU0:ios#
```

# Unidirectional Link Routing

Unidirectional Link Routing(UDLR) feature allows a port to unidirectionally transmit or receive traffic. Therefore, instead of using two strands of fiber for a full-duplex Gigabit Ethernet or 10Gigabit Ethernet port, UDLR uses only one strand of fiber that either transmits or receives the one-way traffic depending on the configuration. This improves the effectiveness and also enables you to double the bandwidth with existing fiber infrastructure.

Cisco IOS XR Software supports Unidirectional Link Routing feature on these line cards:

- A9K- 24T-TR 24-port 10 Gigabit Ethernet line cards
- A9K- 24T-SE 24-port 10 Gigabit Ethernet line cards

- A9K- 36T-TR 36-port 10 Gigabit Ethernet line cards
- A9K- 36T-SE 36-port 10 Gigabit Ethernet line cards

UDLR is used for applications such as video streaming, where most of the traffic is sent as unacknowledged unidirectional video broadcast streams.

# How to Configure Ethernet

This section provides the following configuration procedures:

**Note** When the network experiences a Loss of Signal (LOS) in L0 protection (Client protection) switchover, the physical layer on the router triggers serdes retuning which takes approximately 300 milliseconds to complete. This delay can lead to additional convergence time causing traffic drops, in SNCP (Sub-Network Connection Protection) clients with BFD (Bidirectional Forwarding Detection) configured.

To mitigate this issue, for the ports connected to the protection network configure the LOS holdoff timer. This timer will add a wait on the physical device for the specified duration before initiating the retuning process, thereby preventing delays that exceed the BFD timeout.

This configuration is recommended for the SNCP ports, there is no change in the default behavior for the other ports.

Supported hardware modules:

- A99-10X400GE-X-*
- A9K-20HG-FLEX-*
- A9K-8HG-FLEX-*
- A9K-4HG-FLEX-*
- Cisco 9903
- Cisco 9902

Supported Modes:

- 1x100G
- 1x400G

# Configuring Ethernet Interfaces

This section provides the following configuration procedures:

## Configuring Gigabit Ethernet Interfaces

Use the following procedure to create a basic Gigabit Ethernet or 10-Gigabit Ethernet interface configuration.

## SUMMARY STEPS

1. **show version**
2. **show interfaces** [**GigabitEthernet** | **TenGigE**
3. **configure**
4. **interface** [**GigabitEthernet | TenGigE**
5. **ipv4 address** *ip-address mask*
6. **flow-control** {**bidirectional**| **egress** | **ingress**}
7. **mtu** *bytes*
8. **mac-address** *value1.value2.value3*
9. **negotiation auto**
10. **no shutdown**
11. **end** or **commit**
12. **show interfaces** [**GigabitEthernet | TenGigE**] *interface-path-id*

## DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **show version**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show version` | (Optional) Displays the current software version, and can also be used to confirm that the router recognizes the modular services card. |
| **Step 2** | **show interfaces** [**GigabitEthernet** | **TenGigE**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show interface TenGigE 0/1/0/0` | (Optional) Displays the configured interface and checks the status of each interface port.<br>Possible interface types for this procedure are:<br>• GigabitEthernet<br>• TenGigE |
| **Step 3** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure terminal` | Enters global configuration mode. |
| **Step 4** | **interface** [**GigabitEthernet | TenGigE**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0` | Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are:<br>• GigabitEthernet<br>• TenGigE<br>**Note**<br>• The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **ipv4 address** *ip-address mask*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224` | Assigns an IP address and subnet mask to the interface.<br>• Replace *ip-address* with the primary IPv4 address for the interface.<br>• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:<br>• The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.<br>• The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address. |
| **Step 6** | **flow-control** {**bidirectional**\| **egress** \| **ingress**}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# flow control ingress` | (Optional) Enables the sending and processing of flow control pause frames.<br>• **egress**—Enables the sending of flow control pause frames in egress.<br>• **ingress**—Enables the processing of received pause frames on ingress.<br>• **bidirectional**—Enables the sending of flow control pause frames in egress and the processing of received pause frames on ingress. |
| **Step 7** | **mtu** *bytes*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# mtu 1448` | (Optional) Sets the MTU value for the interface.<br>• The default is 1514 bytes for normal frames and 1518 bytes for 802.1Q tagged frames.<br>• The range for Gigabit Ethernet and 10-Gigabit Ethernet mtu values is 64 bytes to 65535 bytes. |
| **Step 8** | **mac-address** *value1.value2.value3*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# mac address 0001.2468.ABCD` | (Optional) Sets the MAC layer address of the Management Ethernet interface.<br>• The values are the high, middle, and low 2 bytes, respectively, of the MAC address in hexadecimal. The range of each 2-byte value is 0 to ffff. |
| **Step 9** | **negotiation auto**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# negotiation auto` | (Optional) Enables autonegotiation on a Gigabit Ethernet interface.<br>• Autonegotiation must be explicitly enabled on both ends of the connection, or speed and duplex settings |

| | Command or Action | Purpose |
|---|---|---|
| | | must be configured manually on both ends of the connection.<br>• If autonegotiation is enabled, any speed or duplex settings that you configure manually take precedence.<br>**Note**<br>• The **negotiation auto** command is available on Gigabit Ethernet interfaces only. |
| **Step 10** | **no shutdown**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# no shutdown` | Removes the shutdown configuration, which forces an interface administratively down. |
| **Step 11** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 12** | **show interfaces** [**GigabitEthernet** \| **TenGigE**] *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0` | (Optional) Displays statistics for interfaces on the router. |

### What to do next

To configure MAC Accounting on the Ethernet interface, see the “Configuring MAC Accounting on an Ethernet Interface” section later in this module.

To configure an AC on the Ethernet port for Layer 2 VPN implementation, see the "Configuring a L2VPN Ethernet Port" section later in this module.

To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

## Configuring a Fast Ethernet Interface

### What to do next

- To configure an AC on the Fast Ethernet port for Layer 2 VPN implementation, see the "Configuring a L2VPN Ethernet Port" section later in this module.
- To attach Layer 3 service policies, such as Multiprotocol Label Switching (MPLS) or Quality of Service (QoS), to the Fast Ethernet interface, refer to the appropriate Cisco ASR 9000 Series Router or Cisco IOS XR software configuration guide.

## Configuring MAC Accounting on an Ethernet Interface

This task explains how to configure MAC accounting on an Ethernet interface. MAC accounting has special show commands, which are illustrated in this procedure. Otherwise, the configuration is the same as configuring a basic Ethernet interface, and the steps can be combined in one configuration session. See "Configuring Gigabit Ethernet Interfaces" in this module for information about configuring the other common parameters for Ethernet interfaces.

### SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | **fastethernet**] *interface-path-id*
3. **ipv4 address** *ip-address mask*
4. **mac-accounting {egress | ingress}**
5. **end** or **commit**
6. **show mac-accounting** *type* **location** *instance*

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** [**GigabitEthernet** | **TenGigE** | **fastethernet**] *interface-path-id*<br>**Example:** | Physical interface or virtual interface.<br>**Note**<br>• Use the **show interfaces** command to see a list of all interfaces currently configured on the router. |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0` | For more information about the syntax for the router, use the question mark (**?**) online help function. |
| **Step 3** | **ipv4 address** *ip-address mask*<br>**Example:**<br>`RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224` | Assigns an IP address and subnet mask to the interface.<br>• Replace *ip-address* with the primary IPv4 address for the interface.<br>• Replace *mask* with the mask for the associated IP subnet. The network mask can be specified in either of two ways:<br>• The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means that the corresponding address bit belongs to the network address.<br>• The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address. |
| **Step 4** | **mac-accounting {egress \| ingress}**<br>**Example:**<br>`RP/0/RP0/CPU0:router(config-if)# mac-accounting egress` | Generates accounting information for IP traffic based on the source and destination MAC addresses on LAN interfaces.<br>• To disable MAC accounting, use the **no** form of this command. |
| **Step 5** | **end** or **commit**<br>**Example:**<br>`RP/0/RP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **show mac-accounting** *type* **location** *instance*<br>**Example:**<br>`RP/0/RP0/CPU0:router# show mac-accounting TenGigE location 0/2/0/4` | Displays MAC accounting statistics for an interface. |

## Configuring a L2VPN Ethernet Port

Use the following procedure to configure an L2VPN Ethernet port.

**Note** The steps in this procedure configure the L2VPN Ethernet port to operate in port mode.

To configure a point-to-point pseudowire xconnect on an AC, see the Implementing MPLS Layer 2 VPNs module of the Cisco IOS XR L2VPN and Ethernet Services Configuration Guide for the Cisco CRS Router.

To attach Layer 2 service policies, such as quality of service (QoS), to the Ethernet interface, refer to the appropriate Cisco IOS XR software configuration guide.

### SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **l2transport**
4. **l2protocol cpsv** {**tunnel** | **reverse-tunnel**}
5. **end** or **commit**
6. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id*

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0` | Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are:<br>• GigabitEthernet<br>• TenGigE |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **l2transport**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# l2transport` | Enables Layer 2 transport mode on a port and enter Layer 2 transport configuration mode. |
| **Step 4** | **l2protocol cpsv** {**tunnel** \| **reverse-tunnel**}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel` | Configures Layer 2 protocol tunneling and protocol data unit (PDU) filtering on an Ethernet interface.<br>• **tunnel**—Specifies L2PT encapsulation on frames as they enter the interface, and de-encapsulation on frames as they exit they interface.<br>• **reverse-tunnel**—Specifies L2PT encapsulation on frames as they exit the interface, and de-encapsulation on frames as they enter the interface. |
| **Step 5** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if-l2)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if-l2)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 6** | **show interfaces** [**GigabitEthernet** \| **TenGigE**] *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0` | (Optional) Displays statistics for interfaces on the router. |

To configure a point-to-point pseudowire xconnect on an AC, refer to these documents:

- Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide

• Cisco ASR 9000 Series Aggregation Services Router VPN and Ethernet Services Command Reference

# Configuring LLDP

**Note** LLDP is not supported on the FP-X line cards.

This section includes the following configuration topics for LLDP:

## LLDP Default Configuration

This table shows the values of the LLDP default configuration on the Cisco ASR 9000 Series Router. To change the default settings, use the LLDP global configuration and LLDP interface configuration commands.

| LLDP Function | Default |
|---|---|
| LLDP global state | Disabled |
| LLDP holdtime (before discarding) | 120 seconds |
| LLDP timer (packet update frequency) | 30 seconds |
| LLDP reinitialization delay | 2 seconds |
| LLDP TLV selection | All TLVs are enabled for sending and receiving. |
| LLDP interface state | Enabled for both transmit and receive operation when LLDP is globally enabled. |

## Enabling LLDP Globally

To run LLDP on the router, you must enable it globally. When you enable LLDP globally, all interfaces that support LLDP are automatically enabled for both transmit and receive operations.

You can override this default operation at the interface to disable receive or transmit operations. For more information about how to selectively disable LLDP receive or transmit operations for an interface, see the "Disabling LLDP Receive and Transmit Operation for an Interface" section on page 57.

To enable LLDP globally, complete the following steps:

### SUMMARY STEPS

1. **configure**
2. **lldp**
3. **end** or **commit**

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **lldp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)#lldp` | Enables LLDP globally for both transmit and receive operation on the system. |
| **Step 3** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Configuring Global LLDP Operational Characteristics

The "LLDP Default Configuration" section on page 53 describes the default operational characteristics for LLDP. When you enable LLDP globally on the router using the **lldp** command, these defaults are used for the protocol.

To modify the global LLDP operational characteristics such as the LLDP neighbor information holdtime, initialization delay, or packet rate, complete the following steps:

### SUMMARY STEPS

1. **configure**

2. **lldp holdtime** *seconds*
3. **lldp reinit** *seconds*
4. **lldp timer** *seconds*
5. **end** or **commit**

## DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **lldp holdtime** *seconds*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)#lldp holdtime 60` | (Optional) Specifies the length of time that information from an LLDP packet should be held by the receiving device before aging and removing it. |
| **Step 3** | **lldp reinit** *seconds*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# lldp reinit 4` | (Optional) Specifies the length of time to delay initialization of LLDP on an interface. |
| **Step 4** | **lldp timer** *seconds*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)#lldp reinit 60` | (Optional) Specifies the LLDP packet rate. |
| **Step 5** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. |

| | Command or Action | Purpose |
|---|---|---|
| | | • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Disabling Transmission of Optional LLDP TLVs

Certain TLVs are classified as mandatory in LLDP packets, such as the Chassis ID, Port ID, and Time to Live (TTL) TLVs. These TLVs must be present in every LLDP packet. You can suppress transmission of certain other optional TLVs in LLDP packets.

To disable transmission of optional LLDP TLVs, complete the following steps:

### SUMMARY STEPS

1. **configure**
2. **lldp tlv-select** *tlv-name* **disable**
3. **end** or **commit**

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **lldp tlv-select** *tlv-name* **disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# lldp tlv-select system-capabilities disable` | (Optional) Specifies that transmission of the selected TLV in LLDP packets is disabled. The *tlv-name* can be one of the following LLDP TLV types:<br>• **management-address**<br>• **port-description**<br>• **system-capabilities**<br>• **system-description**<br>• **system-name** |
| **Step 3** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# end`<br>or | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config)# commit` | • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Disabling LLDP Receive and Transmit Operation for an Interface

When you enable LLDP globally on the router, all supported interfaces are automatically enabled for LLDP receive and transmit operation. You can override this default by disabling these operations for a particular interface.

To disable LLDP receive and transmit operations for an interface, complete the following steps:

### SUMMARY STEPS

1. **configure**
2. **interface GigabitEthernet 0/2/0/0**
3. **lldp**
4. **receive disable**
5. **transmit disable**
6. **end** or **commit**

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface GigabitEthernet 0/2/0/0**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)#interface GigabitEthernet 0/2/0/0` | Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. Possible interface types for this procedure are:<br>• GigabitEthernet |

| | Command or Action | Purpose |
|---|---|---|
| | | • TenGigE |
| **Step 3** | **lldp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)#lldp` | (Optional) Enters LLDP configuration mode for the specified interface. |
| **Step 4** | **receive disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-lldp)#receive disable` | (Optional) Disables LLDP receive operations on the interface. |
| **Step 5** | **transmit disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-lldp)#transmit disable` | (Optional) Disables LLDP transmit operations on the interface. |
| **Step 6** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Verifying the LLDP Configuration

This section describes how you can verify the LLDP configuration both globally and for a particular interface.

## Verifying the LLDP Global Configuration

To verify the LLDP global configuration status and operational characteristics, use the **show lldp** command as shown in the following example:

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:16:45.510 DST
Global LLDP information:
        Status: ACTIVE
        LLDP advertisements are sent every 30 seconds
        LLDP hold time advertised is 120 seconds
        LLDP interface reinitialisation delay is 2 seconds
```

If LLDP is not enabled globally, the following output appears when you run the **show lldp** command:

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:42:48.221 DST
% LLDP is not enabled
```

## Verifying the LLDP Interface Configuration

To verify the LLDP interface status and configuration, use the **show lldp interface** command as shown in the following example:

```
RP/0/RSP0/CPU0:router# show lldp interface GigabitEthernet 0/1/0/7
Wed Apr 13 13:22:30.501 DST


GigabitEthernet0/1/0/7:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

To monitor and maintain LLDP on the system or get information about LLDP neighbors, use one of the following commands:

| | Description |
|---|---|
| **clear lldp counters** | Resets LLDP traffic counters or LLDP neighbor information. |
| **show lldp entry** | Displays detailed information about LLDP neighbors. |
| **show lldp errors** | Displays LLDP error and overflow statistics. |
| **show lldp neighbors** | Displays information about LLDP neighbors. |
| **show lldp traffic** | Displays statistics for LLDP traffic. |

To collect or clear LLDP interface statistics, you can use the following commands:

| Command | Description |
|---|---|
| **show lldp traffic interface** *interface_name* | Displays LLDP traffic statistics for the specified interface. |

| Command | Description |
|---|---|
| **clear lldp counters interface** *interface_name* | Clears LLDP traffic statistics for the specified interface. Global statistics remains intact. Similarly, clearing global statistics does not impact the interface statistics. |

### Examples for LLDP Interface Statistics

This example shows interface statistics for **gigabitEthernet0/0/0/0**:

```
Router#show lldp traffic interface gigabitEthernet0/0/0/0
```

This example clears the interface statistics for **gigabitEthernet0/0/0/0**.

```
Router#show lldp traffic interface gigabitEthernet0/0/0/0
```

### Running Configuration

```
Router#show lldp traffic interface gigabitEthernet 0/2/0/8
Wed Aug 24 17:38:11.829 IST

LLDP Interface statistics:
        Total frames out: 28786
        Total frames in: 38417
        Total frames received in error: 0
        Total frames out error: 0
        Total frames discarded: 0
        Total TLVs discarded: 0
        Total TLVs unrecognized: 0
```

# Configuring UDLR

Use the following procedure to configure UDLR:

### SUMMARY STEPS

1. **configure**
2. **interface TenGigE** *interface-path-id*
3. **transport-mode** {**rx-only** | **tx-only**}
4. **end** or **commit**

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **interface TenGigE** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/0` | Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. |
| **Step 3** | **transport-mode** {**rx-only** \| **tx-only**}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# transport-mode tx-only` | Configures the 10GE UDLR mode as receive-only or transmit-only. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

# Configuring the Dual-Rate Line Cards

**Note** Oversubscription will be supported on these line cards in a future release of IOS XR 6.2.x train.

The 24-port and 48-port dual-rate line cards support GE and 10GE speeds.

**Note** See *24-Port 10-Gigabit Ethernet/Gigabit Ethernet Line Card with SFP+ or SFP* and *48-Port 10-Gigabit Ethernet/Gigabit Ethernet Line Card with SFP+ or SFP* sections in the *Cisco ASR 9000 Series Aggregation Services Router Ethernet Line Card Installation Guide* for information on the line cards.

**Note** The 24-port line card has a single Network Processor Unit (NPU). The 48-port line card has two NPUs (one for each group of 24 ports). Configuring more than 20x10GE ports per NPU could result in line drops across all ports, depending on the packet size and traffic type.

To configure the port mode for either GE or 10GE, use the **hw-module location** *location* **port-mode** *run-length***x***speed*[**,***run-length***x***speed*] command, where:

- *run-length* – The number of consecutive same-speed ports, divisible by 4. Valid values are:
  - 24-port line card: 4, 8, 12, 16, 20, 24
  - 48-port line card: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48
- *speed* – Valid values are 1 (for GE) or 10 (for 10GE)

**Note** Observe the following restrictions:

- The total for *run-length* must equal the total number of ports (either 24 or 48).
- If you configure the speed of the first port in a set of 12 ports to 1 (GE), then all 12 ports in that set must be 1G (for example: 12x1). If you configure the speed of the first port in a set of 12 ports to 10 (10G), then ports can be mixed in groups of 4 (for example: 4x10,4x1,4x10; or 8x10,4x1; or 12x10).
  - The following example is a valid port-mode configuration on the 48-port line card:

    ```
    port-mode 4x10,8x1,12x10,12x1,12x10
    ```

  - The following example is not a valid port-mode configuration on the 48-port line card:

    ```
    port-mode 4x1,8x10,12x10,12x1,12x10
    ```

The following procedure is for configuring the port speed on the 48-port 10-Gigabit Ethernet/Gigabit Ethernet Line Cards:

1. Enter global configuration mode and specify that the console terminal will be the source of the configuration commands:

   ```
   RP/0/RSP0/CPU0:router# configure terminal
   ```

2. Specify the port mode:

   ```
   RP/0/RSP0/CPU0:router(config)# hw-module location 0/5/CPU0 port-mode
   4x10,8x1,12x10,12x1,12x10
   ```

3. Enter the **commit** command to commit all changes you made to the running configuration:

   ```
   RP/0/RSP0/CPU0:router(config)# commit
   ```

# Convert Speed of a Dual-mode Optic

A dual-mode optic operates in two modes. You can convert the speed of a dual-mode optic from one to another using the `hw-module location` *node-id* `port` *port number* `breakout` *interface* command.

Cisco 100G and 40G SR-BiDi QSFP Transceiver optic (QSFP-40/100G-SRBD) is a dual-mode optic. By default, the optic works in 100G mode.

In 40-Gbps mode, the Cisco QSFP 40/100-Gbps BiDi transceiver supports link lengths of 100 and 150 meters on laser-optimized OM3 and OM4 multimode fibers, respectively. In 100-Gbps mode, it supports 70 and 100 meters on OM3 and OM4, respectively.

FEC is not supported on QSFP-40/100G-SRBD optic.

Support for QSFP-40/100G-SRBD optic is available on following hardware:

- Cisco ASR 9901 Router

**Note** Configuring breakout on port 20 will shut down ports 10 and 11. Configuring breakout on port 21 will shut down ports 24 and 25. Use the **no shutdown** command to reenable these ports.

- Cisco ASR 9000 12-Port 100GE line card (A99-12x100GE)
- Cisco ASR 9000 4-Port 100GE line card (A9K-4x100GE)

### Configuration

This task enables the user to change the dual-mode optic speed option:

```
RP/0/RSP0/CPU0:router (config)# hw-module location 0/0/CPU0 port 21 breakout 1xFortyGigE
RP/0/RSP0/CPU0:router (config)# commit
```

### Verification

Use the `show controller` command to verify the configuration:

```
RP/0/RSP0/CPU0:router# show controllers FortyGigE 0/0/0/21/0 internal
Wed Nov 11 06:34:26.861 UTC

Internal data for interface: FortyGigE0/0/0/21/0
Subport Number : 0
Port Number : 21
Bay Number : 0
Ifinst : 6
Ifinst Subport : 21
Board Type : 0x003d1013

Bandwidth(Kbps) : 40000000
Transport mode : LAN
BIA MAC addr : badb.ad03.a84d
Oper. MAC addr : badb.ad03.a84d
Egress MAC addr : badb.ad03.a84d
Port Available : true
Status polling is : enabled
Status events are : enabled
I/F Handle : 0x04001300
```

```
Cfg Link Enabled : tx/rx enabled
H/W Tx Enable : yes
MTU : 1514
H/W Speed : 40 Gbps
H/W Loopback Type : None
FEC : Disable
H/W FlowCtrl Type : None
H/W AutoNeg Enable : Off
Rx OPD : Not Supported
H/W Link Defects : (0x0000000000000000) none
H/W Raw Link Defects : (0x0000000000000000) none
Link Up : yes
Link Led Status : Link up -- Green/Amber
Serdes fw version : 100.0
Pluggable Present : yes
Pluggable Type : 100/40G SRBD
Pluggable PID : QSFP-40/100-SRBD
Pluggable Compl. : Compliant
Pluggable Type Supp.: Supported
Pluggable PID Supp. : Supported
```

# Configuring MTU

Use the following commands to configure the MTU size on an interface.

```
Router(config)#interface interface name
Router(config-if)#mtu size in bytes
Router(config-if)commit
```

### Example

```
Router(config)#interface Bundle-Ether1
Router(config-if)#mtu 9646
Router(config-if)#commit
Wed Feb  2 07:50:50.275 UTC
```

*Bundle-Ether1* is the interface name.

### Verification

Use the **show** command to verify the MTU size.

```
Router#show ipv4 interface br
Wed Feb  2 07:51:08.732 UTC

Interface               IP-Address      Status     Protocol  Vrf-Name
Bundle-Ether1           10.0.0.2        Up          Up       default
Bundle-Ether2           10.1.1.2        Up          Up       default
MgmtEth0/RSP0/CPU0/0    10.3.3.238      Up          Up       default
MgmtEth0/RSP1/CPU0/0    unassigned      Shutdown    Down     default
HundredGigE0/1/0/40     unassigned      Up          Up       default
HundredGigE0/1/0/41     unassigned      Up          Up       default
HundredGigE0/1/0/42     unassigned      Shutdown    Down     default
HundredGigE0/1/0/43     unassigned      Shutdown    Down     default
Router#show in
infra-sec-keys  inline-service  install  interfaces
inventory

Router#show interfaces Bundle-Ether 1
Wed Feb  2 07:52:55.249 UTC
Bundle-Ether1 is up, line protocol is up
  Interface state transitions: 1
```

```
Hardware is Aggregated Ethernet interface(s), address is 10f3.1176.f6a4
Internet address is 10.0.0.2/24
MTU 9646 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
   reliability 250/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 100000Mb/s
```

# Creating Slices on a Router

This section describes the procedures to configure slices and port-groups on the Cisco ASR 9000 Series 5th Generation Line Cards, and includes the following topics:

## Overview

Slicing is a way to transform a shared network into a set of logical networks. A network slice is a logical group of components, such as Optics (QSFP/SFP), PHY, Network Processor, and Fabric Cards. The system already has the default slicing configured when you enable it. However, you can configure the slices as per your network or service requirement.

You can enable slice configuration on the following:

Routers:

- ASR 9902
- ASR 9903
    - A9903-8HG-PEC

Line cards:

- A9K-4HG-FLEX-SE/-TR
- A99-4HG-FLEX-SE/-TR

### Advantages and Benefits

- Each slice enables you to deliver traffic of up to 400 Gbps, which is the maximum capacity that an NPU can handle.
- Provides the capability to the users to choose a speed from 400GE to 10GE.

### Restrictions on Slice Configurations

- You must configure port speed on the slices in the same or decreasing order. Beginning from the highest port speed to the lowest port speed.
- For 10x1GE or 5x1GE-5x10GE breakout configuration modes with SFP-1G-SX, SFP-1G-LH, SFP-1G-T-X optics, the flow-control configuration must be added under the GigabitEthernet Interface settings to prevent packet drops.
- The second breakout configuration on a port cannot proceed until the first breakout configuration is completed and verified by matching the interface to the first breakout configuration during back-to-back breakout/mode/commit replace–rollback operations.

- The maximum delay duration between breakout configuration changes on the same port or slice is 1 minute 30 seconds for a standalone and up to 5 minutes for bulk commit.

## Slice and Port Numbering

The following tables lists the possible group combinations on the line cards and routers:

***Table 6: Possible Configurations for Cisco ASR 9000 Series Routers and 5th Generation Line Cards***

| Router/Line card | Release | Supported Configurations |
|---|---|---|
| ASR 9902 | Release 7.4.1 | `Default: 1x100GE,1x100GE,10x10GE,10x10GE`<br>`1x100GE,1x100GE,4x25GE,10x10GE`<br>`1x100GE,4x25GE,4x25GE,1x100GE`<br>`1x100GE,1x100GE,1x100GE,1x100GE` |
| | Release 7.5.1 | `Default: 1x100GE,1x100GE,10x10GE,10x10GE`<br>`1x100GE,1x100GE,4x25GE,10x10GE`<br>`1x100GE,4x25GE,4x25GE,1x100GE`<br>`1x100GE,1x100GE,1x100GE,1x100GE`<br>**Note** The following configuration modes are not supported:<br>• 1x40GE/4x10GE<br>• 5x1GE_5x10GE |
| | Release 7.5.2 | Added support for the following configuration mode(s):<br>• 5x1GE_5x10GE |

| Router/Line card | Release | Supported Configurations |
|---|---|---|
| ASR 9903 | Release 7.4.1 | `Default: 10x10GE,10x10GE,4x10GE,UNUSED`<br>`4x25GE,4x25GE,4x25GE,4x25GE` |
| | Release 7.5.1 | `Default: 10x10GE,10x10GE,4x10GE,UNUSED`<br>`4x25GE,4x25GE,4x25GE,4x25GE` |
| | Release 7.9.1 | Added support for the following configuration mode(s):<br>• 5x1GE_5x10GE |
| Cisco ASR 9903 800G Multirate Port Expansion Card (A9903-8HG-PEC) | Release 7.4.1 | `Default: 4x25GE,4x25GE,4x25GE,4x25GE`<br>`1x100GE,1x100GE,4x25GE,10x10GE` |
| | Release 7.9.1 | Added support for the following configuration mode(s):<br>• 5x1GE_5x10GE |
| A9K-4HG-FLEX-SE<br>A9K-4HG-FLEX-TR<br>A99-4HG-FLEX-SE<br>A99-4HG-FLEX-TR | | `Default: 10x10GE,10x10GE,4x10GE,UNUSED`<br>`4x25GE,4x25GE,4x25GE,4x25GE` |
| | Release 7.9.1 | Added support for the following configuration mode(s):<br>• 5x1GE_5x10GE |

## Configure Slices

All configurations can be accomplished by using appropriate values for the **hw-module** command.

**Note** Slice configuration is available on A9903-8HG-PEC. For the other ports on PECs, fixed ports, or slices, use the breakout commands.

To configure the slice, use the following command:

```
configurehw-module location location slice [slice_number ] config-mode
[1x100GE,1x100GE,4x25GE,10x10GE]
```

Consider a scenario, where, you as a user or a customer wants to effectively utilize the bandwidth of the router. In this scenario, we are going to configure a slice on a ASR 9000 Series Router that has a A9903-8HG-PEC line card installed. With A9903-8HG-PEC line card, you can configure slice 4 and 5. The rest of the slices are reserved for other line cards and fixed boards.

To configure the slices on the router or the line card, perform the following steps:

1. Identify the router capability, the line cards installed on the router, and the maximum bandwidth supported. In this scenario, the maximum supported bandwidth is 400GE per Network Processor.
2. Configure slices on the line card or router by using the **configure hw-module location** *location* **slice** [*slice_number* ] config-mode command.
   a. Specify the slices, such as slice 4.
   b. Specify the speed on the ports, such as 4x25GE,4x25GE,4x25GE,4x25GE or 1x100GE,1x100GE,4x25GE,10x10GE.
   c. Save the configuration.

Now, you have successfully configured the slices on the router, which will allow you to use the network bandwidth as per your requirement, without letting the bandwidth getting wasted, which wasn't possible earlier with the port breakout functionality.

For example, you can use the 4x25GE bandwidth for a lower bandwidth consumption service, such as voice calling. And use the 1x100GE bandwidth for video calling service.

### Configuration Steps

```
router# configure terminal
router(config)# hw-module location 0/0/CPU0 slice 4 config-mode  4x25GE,4x25GE,4x25GE,4x25GE
  /* Specify the slice where you configure the groups and specify the port speeds for each
 group*/
router(config)# commit
```

### Verify Slice Configuration

Use this procedure to verify whether the slice is correctly configured.

#### Verification on ASR 9903

```
Router: #show controllers np valid-port-groups all location 0/0/CPU0
Tue Dec 21 10:47:37.522 UTC

                Node: 0/0/CPU0:
----------------------------------------------------------------

 NP0 valid-port-groups is not supported

 NP1 valid-port-groups is not supported

 NP2 valid-port-groups is not supported

 NP3 valid-port-groups is not supported

 NP4: Valid Port Groups
----+---------------------+---------------------+---------------------+---------------------
SEL | Group 0             | Group 1             | Group 2             | Group 3
----+---------------------+---------------------+---------------------+---------------------
DFLT 10x10GE               10x10GE               4x10GE                UNUSED

1    4x25GE                4x25GE                4x25GE                4x25GE

Notes:
  DFLT: SFP10g port 0-23 (default)
        DFLT is set by default or by clearing any non-default configuration using "no
hw-module loc <> slice config-mode"
```

```
     1: SFP25g port 0-15

 NP5: Valid Port Groups
----+---------------------+---------------------+---------------------+---------------------
SEL | Group 0             | Group 1             | Group 2             | Group 3
----+---------------------+---------------------+---------------------+---------------------
DFLT 10x10GE              10x10GE               4x10GE                UNUSED

1    4x25GE               4x25GE                4x25GE                4x25GE

Notes:
  DFLT: SFP10g port 24-47 (default)
        DFLT is set by default or by clearing any non-default configuration using "no
hw-module loc <> slice config-mode"
     1: SFP25g port 32-47

 NP6 valid-port-groups is not supported

 NP7 valid-port-groups is not supported

 NP8 valid-port-groups is not supported
Router: #


RP/0/RP0/CPU0:router#sh controllers np ports all location 0/0/CPU0
Tue Jan 11 10:43:35.581 UTC
               Node: 0/0/CPU0:
----------------------------------------------------------------
NP Bridge Fia                      Ports
-- ------ --- -----------------------------------------------
0  --     0   TenGigE0/0/0/0/0 - TenGigE0/0/0/0/3, TenGigE0/0/0/1/0 - TenGigE0/0/0/1/3,
HundredGigE0/0/0/2, FortyGigE0/0/0/3/0
1  --     1   FortyGigE0/0/0/4/0, HundredGigE0/0/0/5, TenGigE0/0/0/6/0 - TenGigE0/0/0/6/3,
 FortyGigE0/0/0/7/0
2  --     2   FortyGigE0/0/0/8/0, FortyGigE0/0/0/9/0, TenGigE0/0/0/26 - TenGigE0/0/0/35,
FortyGigE0/0/0/11/0
3  --     3   FortyGigE0/0/0/12/0, HundredGigE0/0/0/13 - HundredGigE0/0/0/15
4  --     4   TwentyFiveGigE0/0/1/0 - TwentyFiveGigE0/0/1/15
5  --     5   TenGigE0/0/1/24 - TenGigE0/0/1/47
```

### Verification on A9K-4HG-FLEX-SE Line Cards

```
RP/0/RSP0/CPU0:asr-sat-host1(config)#do show controllers np valid-port-groups all location
 0/0/CPU0
Tue Dec 21 05:07:32.819 UTC

               Node: 0/0/CPU0:
----------------------------------------------------------------

 NP0: Valid Port Groups
----+---------------------+---------------------+---------------------+---------------------
SEL | Group 0             | Group 1             | Group 2             | Group 3
----+---------------------+---------------------+---------------------+---------------------
DFLT 1x100GE              1x100GE               1x100GE               1x100GE

1    1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE

2    1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE 10x10GE

3    1x100GE/1x40GE/4x25GE 1x100GE/1x40GE/4x25GE 10x10GE               10x10GE

4    1x100GE/1x40GE/4x25GE 10x10GE               10x10GE               10x10GE

5    10x10GE               10x10GE               10x10GE               10x10GE
```

```
Notes:
  DFLT : QSFP port 40, QSFP port 41, QSFP port 42, QSFP port 43 (default)
  Group 0: QSFP port 40 or SFP10g port 0-9   or SFP25g port 6-9
  Group 1: QSFP port 41 or SFP10g port 10-19 or SFP25g port 16-19
  Group 2: QSFP port 42 or SFP10g port 20-29 or SFP25g port 26-29
  Group 3: QSFP port 43 or SFP10g port 30-39 or SFP25g port 36-39
RP/0/RSP0/CPU0:asr-sat-host1(config)#
```

For information on breakout, see the hw-module location command.

# Configuration Examples for Ethernet

This section provides the following configuration examples:

## Configuring an Ethernet Interface: Example

The following example shows how to configure an interface for a 10-Gigabit Ethernet modular services card:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/1
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# flow-control ingress
RP/0/RSP0/CPU0:router(config-if)# mtu 1448
RP/0/RSP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/0/0/1

TenGigE0/0/0/1 is down, line protocol is down
  Hardware is TenGigE, address is 0001.2468.abcd (bia 0001.81a1.6b23)
  Internet address is 172.18.189.38/27
  MTU 1448 bytes, BW 10000000 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
Encapsulation ARPA,
  Full-duplex, 10000Mb/s, LR
  output flow control is on, input flow control is on
  Encapsulation ARPA,
ARP type ARPA, ARP timeout 01:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

# Configuring MAC-Accounting: Example

This example indicates how to configure MAC-accounting on an Ethernet interface:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# mac-accounting egress
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

# Configuring a Layer 2 VPN AC: Example

The following example indicates how to configure a Layer 2 VPN AC on an Ethernet interface:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol  tunnel
RP/0/RSP0/CPU0:router(config-if-l2)# commit
```

# Configuring LLDP: Examples

The following example shows how to enable LLDP globally on the router and modify the default LLDP operational characteristics:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# lldp
RP/0/RSP0/CPU0:router(config)# lldp holdtime 60
RP/0/RSP0/CPU0:router(config)# lldp reinit 4
RP/0/RSP0/CPU0:router(config)# lldp timer 60
RP/0/RSP0/CPU0:router(config)# commit
```

The following example shows how to disable a specific Gigabit Ethernet interface for LLDP transmission:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0
RP/0/RSP0/CPU0:router(config-if)# lldp
RP/0/RSP0/CPU0:router(config-lldp)# transmit disable
```

### Where to Go Next

When you have configured an Ethernet interface, you can configure individual VLAN subinterfaces on that Ethernet interface.

For information about modifying Ethernet management interfaces for the shelf controller (SC), route processor (RP), and distributed RP, see the Advanced Configuration and Modification of the Management Ethernet Interface on the Cisco ASR 9000 Series RouterAdvanced Configuration and Modification of the Management Ethernet Interface on the Cisco ASR 9000 Series Router module later in this document.

For information about IPv6 see the Implementing Access Lists and Prefix Lists on Cisco IOS XR Software module in the Cisco IOS XR IP Addresses and Services Configuration Guide.

# How to Configure Interfaces in Breakout Mode

*Table 7: Feature History Table*

| Feature name | Release Information | Feature Description |
|---|---|---|
| 1x10GbE Breakout Ports on Cisco ASR 9903 Router and Cisco A9K-20HG-FLEX-SE Line Cards | Release 24.3.1 | You can now configure 1x10GbE breakout mode on the Cisco ASR 9903 Router and Cisco A9K-20HG-FLEX-SE line card using SFP-10G-ER, SFP-10G-ZR, and SFP-10G-LR hot-pluggable transceiver modules.<br>The feature introduces these changes:<br>**CLI**:<br>• The **1xTenGigE** keyword is introduced in the **hw-module location** command. |
| 4x25GbE Breakout Ports on Cisco ASR 9903 Router | Release 7.5.2 | You can now configure 4x25GbE breakout ports on all QSFP28 ports of the Cisco ASR 9903 fixed port router with QSFP-100G-SR4-S hot-pluggable transceiver module. |

## Information About Breakout

The router supports transmission of traffic in the breakout mode. The breakout mode enables a 100GbE or 400GbE port to be split into multiple independent and logical GbE ports.

Breakout mode options:

- 1x400GbE
- 1x100GbE
- 2x100GbE
- 4x100GbE
- 2x40GbE
- 1x40GbE
- 4x25GbE

- 4x10GbE
- 1x10GbE

**Note**

- The supported breakout mode is dependent on the port and optic transceiver module.

### Restrictions on breakout mode configuration

These restrictions outline the hardware-level limitations, port dependencies, and configuration requirements that must be considered when configuring breakout modes:

- **Primary port configuration:** When the primary port in the same bay is configured in a restricted breakout mode, for example, 400G, 8x50G, or 1x40G, breakout configuration on other ports is not allowed.
- **Hardware-Level restriction:** The system enforces hardware-level restrictions that prevent configuring breakout modes on other ports within the same logical group.
- The second breakout configuration on a port cannot proceed until the first breakout configuration is completed and verified by matching the interface to the first breakout configuration during back-to-back breakout/mode/commit replace–rollback operations.
- The maximum delay duration between breakout configuration changes on the same port or slice is 1 minute 30 seconds for a standalone and up to 5 minutes for bulk commit.

See the hw-module location command for more information.

# Breakout Configuration: Examples

These examples show how to configure breakout in a port.

## Configure 4x25GbE Breakout in a Port

This example shows how to configure 4x25GbE breakout in a port:

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# hw-module location 0/0/CPU0 bay 0 port 11 breakout
4xTwentyFiveGigE
RP/0/RP0/CPU0:Router(config)# commit
RP/0/RP0/CPU0:Router(config)# end
RP/0/RP0/CPU0:Router#
```

### Verify a Breakout Configuration

Verifying the 4x25GbE breakout configuration:

```
RP/0/RP0/CPU0:Router# show ipv4 interfaces brief | inc 0/11/
       TwentyFiveGigE0/0/0/11/0       198.127.6.1      Up              Up       vpn24
       TwentyFiveGigE0/0/0/11/1       198.127.7.1      Up              Up       vpn7
       TwentyFiveGigE0/0/0/11/2       198.127.4.1      Up              Up       vpn4
       TwentyFiveGigE0/0/0/11/3       198.127.9.1      Up              Up       vpn21
```

### Remove the Breakout Configuration

Removing the 4x25GbE breakout configuration:

```
RP/0/RP0/CPU0:uut# configure
RP/0/RP0/CPU0:uut(config)# no hw-module location 0/0/CPU0 bay 0 port 11 breakout
4xTwentyFiveGigE
RP/0/RP0/CPU0:uut(config)# commit
RP/0/RP0/CPU0:uut(config)# end
```

## Configure 4x100GbE Breakout in a Port

This example shows how to configure 4x100GbE breakout in a port:

```
RP/0/RP0/CPU0:Router# configure
RP/0/RP0/CPU0:Router(config)# hw-module location 0/0/CPU0 port 0 breakout 4xHundredGigE
RP/0/RP0/CPU0:Router(config)# commit
RP/0/RP0/CPU0:Router(config)# end
RP/0/RP0/CPU0:Router#
```

### Verify a Breakout Configuration

Verifying the 4x100GbE breakout configuration:

```
RP/0/RP0/CPU0:Router# show ipv4 interfaces brief | inc 0/11/
       HundredGigE0/0/0/11/0       198.127.6.1      Up              Up       vpn24
       HundredGigE0/0/0/11/1       198.127.7.1      Up              Up       vpn7
       HundredGigE0/0/0/11/2       198.127.4.1      Up              Up       vpn4
       HundredGigE0/0/0/11/3       198.127.9.1      Up              Up       vpn21
```

### Remove the Breakout Configuration

Removing the 4x100GbE breakout configuration:

```
RP/0/RP0/CPU0:uut# configure
RP/0/RP0/CPU0:uut(config)# no hw-module location 0/0/CPU0 port 0 breakout 4xHundredGigE
RP/0/RP0/CPU0:uut(config)# commit
RP/0/RP0/CPU0:uut(config)# end
```

## Configure 1x10GbE Breakout in a Port

**Note** The 1x10GbE breakout mode is supported on Cisco ASR 9903 router and Cisco A9K-20HG-FLEX-SE line card with SFP-10G-ER, SFP-10G-ZR, and SFP-10G-LR hot-pluggable transceiver modules.

This example shows how to configure 1x10GbE breakout in a port:

```
Router# configure
Router(config)# hw-module location 0/0/CPU0 bay 1 port 9 breakout 1xTenGbE
Router(config)# commit
Router(config)# end
Router#
```

### Verify a Breakout Configuration

Use the **show running configuration** command to verify that breakout is configured on the TenGigE0/0/1/9/0 interface.

```
Router# show running-config TenGigE0/0/1/9/0
hw-module location 0/0/CPU0 bay 1 port 9 breakout 1xTenGbE
interface TenGigE0/0/1/9/0
shutdown
!
```

Use the **show controller optics** command to display status and configuration information about the interfaces configured as optics controller on a specific node. In this example, the router displays that the 0/0/1/9 interface is configured on the SFP-10G-ER optics.

```
Router#show controller optics 0/0/1/9

 Controller State: Up

 Transport Admin State: In Service

 Laser State: On

 Host Squelch Status: Enable

 LED State: Green

 FEC State: FEC DISABLED

 Optics Status

     Optics Type:  10G SFP+ ER
     Wavelength = 1550.00 nm

     Alarm Status:
     -------------
     Detected Alarms:
         LOW-RX1-PWR
         LOW-TX1-PWR

     LOS/LOL/Fault Status:

     Laser Bias Current = 0.0 mA
     Actual TX Power = -40.00 dBm
     RX Power = -40.00 dBm

     Performance Monitoring: Disable

     THRESHOLD VALUES
     ----------------

     Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
     ------------------------  ----------  ---------  ------------  -----------
     Rx Power Threshold(dBm)          2.0      -19.7          -1.0        -15.8
     Tx Power Threshold(dBm)          7.0       -8.6           4.0         -4.7
     LBC Threshold(mA)             114.00      25.50        106.40        30.60
     Temp. Threshold(celsius)       75.00      -5.00         70.00         0.00
     Voltage Threshold(volt)         3.63       2.97          3.46         3.13

     Configured Tx Power = 0.00 dBm
     Configured CD High Threshold = 0 ps/nm
     Configured CD lower Threshold = 0 ps/nm
     Configured OSNR lower Threshold = 0.00 dB
     Configured DGD Higher Threshold = 0.00 ps
     Polarization parameters not supported by optics

     Temperature = 27.12 Celsius
     Voltage = 3.40 V
```

```
 Transceiver Vendor Details

      Form Factor         : SFP+
      Optics type         : SFP-10G-ER
      Name              : CISCO-INNOLIGHT
      OUI Number          : 44.7c.7f
      Part Number         : TR-PX15E-NCA
      Rev Number          : 1A
      Serial Number         : INL2444A0A1
      PID             : SFP-10G-ER
      VID             : V02
      Date Code(yy/mm/dd)     : 20/10/27

RP/0/RP0/CPU0:ios#
```

Use the **show controllers** command to verify that the speed or the breakout mode of 10GbE is configured on the TenGigE0/0/1/9/0 interface.

```
Router#show controllers TenGigE0/0/1/9/0

Operational data for interface TenGigE0/0/1/9/0:

State:
    Administrative state: enabled
    Operational state: Up
    LED state: Green On

Phy:
    Media type: R fiber over 1550nm optics
    Optics:
        Vendor: CISCO-INNOLIGHT
        Part number: TR-PX15E-NCA
        Serial number: INL2444A0A1
        Type: 10G SFP+ ER
        Wavelength: 1550 nm
    Digital Optical Monitoring:
        Transceiver Temp: 31.699 C
        Transceiver Voltage: 3.406

        DOM alarms:
            No alarms

        Alarm                    Alarm     Warning   Warning    Alarm
        Thresholds               High        High     Low       Low
                                 -------    -------  -------   -------
        Transceiver Temp (C):     75.000     70.000    0.000     -5.000
        Transceiver Voltage (V):   3.630      3.465    3.135      2.970
        Laser Bias (mA):         114.000    106.400   30.600     25.500
        Transmit Power (mW):       5.011      2.511    0.388      0.134
        Transmit Power (dBm):      6.999      3.998   -4.711     -8.729
        Receive Power (mW):        1.584      0.794    0.026      0.010
        Receive Power (dBm):       1.998     -1.002  -15.850    -20.000

MAC address information:
    Operational address: 34ed.1b35.8784
    Burnt-in address: 34ed.1b35.8784
    No unicast addresses in filter
    No multicast addresses in filter

Autonegotiation disabled.

Operational values:
    Speed: 10Gbps
    Duplex: Full Duplex
```

```
    Flowcontrol: None
    Loopback: None (or external)
    MTU: 1514
    MRU: 1514
    Inter-packet gap: standard (12)

Router#
```

### Remove the Breakout Configuration

Removing the 1x10GbE breakout configuration:

```
Router# configure
Router(config)# no hw-module location 0/0/CPU0 bay 1 port 9 breakout 1xTenGbE
Router(config)# commit
Router(config)# end
```