Configuring Link Bundling on Cisco IOS XR Software

This module describes the configuration of link bundle interfaces on the Cisco CRS Router.

A link bundle is a group of one or more ports that are aggregated together and treated as a single link.

The different links within a single bundle can have varying speeds, where the fastest link can be a maximum of four times greater than the slowest link.

Each bundle has a single MAC and shares a single Layer 3 configuration set, such as IP address, ACL, Quality of Service (QoS), and so on.

Note Link bundles do not have a one-to-one modular services card association. Member links can terminate on different cards.

Feature History for Configuring Link Bundling

Release	Modification
Release 3.2	This feature was introduced on the Cisco CRS Router.
Release 3.3.0	This feature was updated as follows: • To support the 1:N redundancy feature, users can configure the minimum number of active links using the **bundle minimum-active links** command. • To support the 1:N redundancy feature, users can configure the minimum bandwidth in kbps using the **bundle minimum-active links** command. • Support was added for VLAN subinterfaces on Ethernet link bundles. • Output for **show bundle bundle-Ether** command and **show bundle bundle-POS** command was modified. • The **reasons** keyword was added to the **show bundle bundle-Ether** command and the **show bundle bundle-POS** command. • The **bundle id** command was changed from **bundle-id**. • BFD over bundled VLANs using static routes.
Release 3.4.0	The configuration procedures in this module were modified with enhancements.

Release 3.7.0	Note was added, specifying that link bundling is supported on the multishelf Cisco CRS Router.
Release 3.8.0	This feature was updated as follows: • The **reasons** keyword was removed from the **show bundle bundle-Ether** command and the **show bundle bundle-POS** command. Now, if a port is in a state other than the distributing state, the output of both commands displays the reason. • The **hot-standby** keyword was added to the **bundle maximum-active links** command. • The **lacp fast-switchover** command was added.
Release 3.8.4	Bundle member links are put into new err-disable link interface status and admin-down protocol state when a bundle interface is shut down.
Release 3.9.0	Support for super short LACP was added. Support for load balancing was added. Support for a maximum of 64 member links per bundle was added.
Release 4.0.0	Support for the following physical layer interface modules (PLIMs) was added: • 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (14X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (20X10GBE-WL-XFP) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card)
Release 4.0.1	Support for the following PLIMs was added: • 1-Port 100-Gigabit Ethernet PLIM (1X100GBE) (with the Cisco CRS-3 Modular Services Card or Cisco CRS-3 Forwarding Processor Card) • 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (4-10GBE-WL-XFP) • 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM (8-10GBE-WL-XFP)

Contents

This module includes the following sections:

Prerequisites for Configuring Link Bundling

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.

The prerequisites for link bundling depend on the platform on which you are configuring this feature. This section includes the following information:

- Prerequisites for Configuring Link Bundling on a Cisco CRS Router, page 189

Prerequisites for Configuring Link Bundling on a Cisco CRS Router

Before configuring link bundling on a Cisco IOS XR Router, be sure that the following tasks and conditions are met:

- You know which links should be included in the bundle you are configuring.
- If you are configuring an Ethernet link bundle, you have at least one of the following Ethernet cards installed in the router:
  - 1-Port 10-Gigabit Ethernet SPA (LAN and WAN-PHY)
  - 4-Port 10-Gigabit Ethernet Tunable WDMPHY Physical Layer Interface Module (PLIM)
  - 4-Port 10-Gigabit Ethernet PLIM
  - 5-Port Gigabit Ethernet SPA
  - 8-Port Gigabit Ethernet SPA (versions 1 and 2)
  - 8-Port 10-Gigabit Ethernet PLIM
  - 10-Port Gigabit Ethernet SPA
  - 42-Port Gigabit Ethernet PLIM
  - 4-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 8-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 14-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 20-Port 10-Gigabit Ethernet LAN/WAN-PHY PLIM
  - 1-Port 100-Gigabit Ethernet PLIM
- If you are configuring a POS link bundle, you have a POS line card or SPA installed in a router that is running Cisco IOS XR software.

Note For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco CRS-1 Carrier Routing System 8-Slot Line Card Chassis System Description*.

Information About Configuring Link Bundling

To implement link bundling, you must understand the following concepts:

- Link Bundling Overview, page 190
- Link Aggregation Through LACP, page 192

Link Bundling Overview

The Link Bundling feature allows you to group multiple point-to-point links together into one logical link and provide higher bidirectional bandwidth, redundancy, and load balancing between two routers. A virtual interface is assigned to the bundled link. The component links can be dynamically added and deleted from the virtual interface.

The virtual interface is treated as a single interface on which one can configure an IP address and other software features used by the link bundle. Packets sent to the link bundle are forwarded to one of the links in the bundle.

The advantages of link bundles are as follows:

- Multiple links can span several line cards and SPAs to form a single interface. Thus, the failure of a single link does not cause a loss of connectivity.
- Bundled interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can move onto another link if one of the links within a bundle fails. You can add or remove bandwidth without interrupting packet flow. For example, you can upgrade from an OC-48c PLIM modular services card to an OC-192 PLIM modular services card without interrupting traffic.

All links within a bundle must be of the same type. For example, a bundle can contain all Ethernet interfaces, or it can contain all POS interfaces, but it cannot contain Ethernet and POS interfaces at the same time.

Cisco IOS XR software supports the following methods of forming bundles of Ethernet and POS interfaces:

- IEEE 802.3ad—Standard technology that employs a Link Aggregation Control Protocol (LACP) to ensure that all the member links in a bundle are compatible. Links that are incompatible or have failed are automatically removed from a bundle.
- EtherChannel or POS Channel—Cisco proprietary technology that allows the user to configure links to join a bundle, but has no mechanisms to check whether the links in a bundle are compatible. (EtherChannel applies to Ethernet interfaces, and POS Channel applies to POS interfaces.)

Features and Compatible Characteristics of Link Bundles

Link bundles support the following features:

- ACL
- Basic IP
- Basic MPLS
- MPLS VPN

- Sampled Netflow
- BGP Policy Accounting
- HSRP/VRRP
- VLAN Bundling (Ethernet only)
- Basic IP
- Basic MPLS
- MPLS VPN
- Inter-AS
- WRED/MDRR per member interface.

The following list describes the properties and limitations of link bundles:

- A bundle contains links, each of which has LACP enabled or disabled. If a bundle contains links, some that have LACP enabled and some that have LACP disabled, the links with LACP disabled are not aggregated in the bundle.
- Bundle membership can span across several modular services cards that are installed in a single router and across SPAS in the same service card.
- For Ethernet link bundles and POS bundles, all ports and interfaces added to the bundle should have the same speed and bandwidth.
- Physical layer and link layer configuration are performed on individual member links of a bundle.
- Configuration of network layer protocols and higher layer applications is performed on the bundle itself.
- IPv4 and IPv6 addressing is supported on link bundles.
- A bundle can be administratively enabled or disabled. Beginning in Cisco IOS XR Release 3.8.4, when you shut down a bundle interface, the member links are put into err-disable link interface status and admin-down line protocol state. You can show the status of a bundle interface and its members using the **show interfaces** command.
- Each individual link within a bundle can be administratively enabled or disabled.
- The MAC address that is set on the bundle becomes the MAC address of the links within that bundle.
- MAC address is set on the bundle the address of the If a MAC address is not set on the bundle, the bundle MAC address is obtained from a pool of pre-assigned MAC addresses stored in EEPROM of the chassis midplane.
- Each link within a bundle can be configured to allow different keepalive periods on different members.
- Load balancing (the distribution of data between member links) is done by flow instead of by packet.
- Upper layer protocols, such as routing updates and hellos, are sent over any member link of an interface bundle.
- All links within a single bundle must terminate on the same two systems. Both systems must be directly connected.
- Bundled interfaces are point-to-point.
- A bundle can contain physical links only. Tunnels and VLAN subinterfaces cannot be bundle members. However, you can create VLANs as subinterfaces of bundles.
- An IPv4 address configuration on link bundles is identical to an IPv4 address configuration on regular interfaces.

- Multicast traffic is load balanced over the members of a bundle. For a given flow, internal processes select the member link, and all traffic for that flow is sent over that member.

Characteristics of CRS-1 Series Router Link Bundles

The following list describes additional properties and limitations of link bundles that are specific to CRS-1 Series:

- Link bundling is supported on all multishelf Cisco CRS Routers.
- A bundle can contain all Ethernet interfaces or all POS interfaces, but not a mix of Ethernet and POS interfaces.
- A single bundle supports a maximum of 64 physical links. If you add more than 64 links to a bundle, only 64 of the links function, and the remaining links are automatically disabled.
- A Cisco CRS Router supports a maximum of 32 bundles.
- Ethernet and POS link bundles are created in the same way as Ethernet channels and POS channels, where the user enters the same configuration on both end systems.
- For POS link bundles, different link speeds are allowed within a single bundle, with a maximum of four times the speed difference between the members of the bundle.
- HDLC is the only supported encapsulation type for POS link bundles in Cisco IOS XR software. POS links that are configured with any other encapsulation type cannot join a bundle. Keep in mind that all POS link bundle members must be running HDLC for HDLC to work on a bundle.
- QoS is supported and is applied proportionally on each bundle member.
- Link layer protocols, such as CDP and HDLC keepalives, work independently on each link within a bundle.
- All links within a single bundle must be configured to run either POS Channel or 802.3ad. Mixed bundles are not supported.

Link Aggregation Through LACP

Aggregating interfaces on different modular services cards and on SPAs within the same services cards provides redundancy, allowing traffic to be quickly redirected to other member links when an interface or modular services card failure occurs.

The optional Link Aggregation Control Protocol (LACP) is defined in the IEEE 802 standard. LACP communicates between two directly connected systems (or peers) to verify the compatibility of bundle members. The peer can be either another router or a switch. LACP monitors the operational state of link bundles to ensure the following:

- All links terminate on the same two systems.
- Both systems consider the links to be part of the same bundle.
- All links have the appropriate settings on the peer.

LACP transmits frames containing the local port state and the local view of the partner system's state. These frames are analyzed to ensure both systems are in agreement.

IEEE 802.3ad Standard

The IEEE 802.3ad standard typically defines a method of forming Ethernet link bundles. In Cisco IOS XR software, the IEEE 802.3ad standard is used on both Ethernet and POS link bundles.

For each link configured as bundle member, the following information is exchanged between the systems that host each end of the link bundle:

- A globally unique local system identifier
- An identifier (operational key) for the bundle of which the link is a member
- An identifier (port ID) for the link
- The current aggregation status of the link

This information is used to form the link aggregation group identifier (LAG ID). Links that share a common LAG ID can be aggregated. Individual links have unique LAG IDs.

The system identifier distinguishes one router from another, and its uniqueness is guaranteed through the use of a MAC address from the system. The bundle and link identifiers have significance only to the router assigning them, which must guarantee that no two links have the same identifier, and that no two bundles have the same identifier.

The information from the peer system is combined with the information from the local system to determine the compatibility of the links configured to be members of a bundle.

The MAC address of the first link attached to a bundle becomes the MAC address of the bundle itself. The bundle uses this MAC address until that link (the first link attached to the bundle) is detached from the bundle, or until the user configures a different MAC address. The bundle MAC address is used by all member links when passing bundle traffic. Any unicast or multicast addresses set on the bundle are also set on all the member links.

Note We recommend that you avoid modifying the MAC address, because changes in the MAC address can affect packet forwarding.

LACP Short Period Time Intervals

As packets are exchanged across member links of a bundled interface, some member links may slow down or time-out and fail. LACP packets are exchanged periodically across these links to verify the stability and reliability of the links over which they pass. The configuration of short period time intervals, in which LACP packets are sent, enables faster detection and recovery from link failures.

Short period time intervals are configured as follows:

- In milliseconds
- In increments of 100 milliseconds
- In the range 100 to 1000 milliseconds
- The default is 1000 milliseconds (1 second)
- Up to 64 member links
- Up to 1280 packets per second (pps)

After 6 missed packets, the link is detached from the bundle.

When the short period time interval is *not* configured, LACP packets are transmitted over a member link every 30 seconds by default.

When the short period time interval is configured, LACP packets are transmitted over a member link once every 1000 milliseconds (1 second) by default. Optionally, both the transmit and receive intervals can be configured to less than 1000 milliseconds, independently or together, in increments of 100 milliseconds (100, 200, 300, and so on).

When you configure a custom LACP short period *transmit* interval at one end of a link, you must configure the same time period for the *receive* interval at the other end of the link.

Note You must always configure the *transmit* interval at both ends of the connection before you configure the *receive* interval at either end of the connection. Failure to configure the *transmit* interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the *receive* intervals first and then the *transmit* intervals.

Load Balancing

Load balancing is a forwarding mechanism which distributes traffic over multiple links, based on Layer 3 routing information in the router. Per-flow load balancing is supported on all links in the bundle. This scheme achieves load sharing by allowing the router to distribute packets over one of the links in the bundle, that is determined through a hash calculation. The hash calculation is an algorithm for link selection based on certain parameters.

The standard hash calculation is a 3-tuple hashing, using the following parameters:

- IP source address
- IP destination address
- Router ID

7-tuple hashing can also be configured. based on Layer 3 and Layer 4 parameters:

- IP source address
- IP destination address
- Router ID
- Input interface
- IP protocol
- Layer 4 source port
- Layer 4 destination port

When per-flow load balancing and 3-tuple hashing is enabled, all packets for a certain source-destination pair will go through the same link, though there are multiple links available. Per-flow load balancing ensures that packets for a certain source-destination pair arrive in order.

Note For multicast traffic, ingress forwarding is based on the Fabric Multicast Group Identifier (FGID). Egress forwarding over the bundle is based on the bundle load balancing.

QoS and Link Bundling

On the Cisco CRS Router, QoS is applied to the local instance of a bundle in the ingress direction. Each bundle is associated with a set of queues. QoS is applied to the various network layer protocols that are configured on the bundle. In the egress direction, QoS is applied on the bundle with a reference to the member links. QoS is applied based on the sum of the member bandwidths.

For complete information on configuring QoS on link bundles on the Cisco CRS Router, refer to the *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco CRS Router* and the *Cisco IOS XR Modular Quality of Service Command Reference for the Cisco CRS Router.*

VLANs on an Ethernet Link Bundle

802.1Q VLAN subinterfaces can be configured on 802.3ad Ethernet link bundles. Keep the following information in mind when adding VLANs on an Ethernet link bundle:

- The maximum number of VLANs allowed per bundle is 128.
- The maximum number of bundled VLANs allowed per router is 4000.

Note The memory requirement for bundle VLANs is slightly higher than standard physical interfaces.

To create a VLAN subinterface on a bundle, include the VLAN subinterface instance with the **interface Bundle-Ether** command, as follows:

interface Bundle-Ether *interface-bundle-id.subinterface*

After you create a VLAN on an Ethernet link bundle, all VLAN subinterface configuration is supported on that link bundle.

VLAN subinterfaces can support multiple Layer 2 frame types and services, such as Ethernet Flow Points - EFPs) and Layer 3 services.

Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration. Keep in mind that a link must be cleared of all previous network layer configuration before it can be added to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the **interface Bundle-Ether** command. To create a POS link bundle, enter the **interface Bundle-POS** command.
2. Assign an IP address and subnet mask to the virtual interface using the **ipv4 address** command.
3. Add interfaces to the bundle you created in Step 1 with the **bundle id** command in the interface configuration submode. You can add up to 64 links to a single bundle.
4. On a CRS-1 Series router, optionally implement 1:1 link protection for the bundle by setting the **bundle maximum-active links** command to 1. Performing this configuration causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. (The link priority is based on the value of the **bundle port-priority** command.) If the active link fails, the standby link immediately becomes the active link.

Note A link is configured as a member of a bundle from the interface configuration submode for that link.

Nonstop Forwarding During RP Switchover

Cisco IOS XR software supports nonstop forwarding during switchover between active and standby paired RP cards. Nonstop forwarding ensures that there is no change in the state of the link bundles when a switchover occurs.

For example, if an active RP fails, the standby RP becomes operational. The configuration, node state, and checkpoint data of the failed RP are replicated to the standby RP. The bundled interfaces will all be present when the standby RP becomes the active RP.

Note You do not need to configure anything to guarantee that the standby interface configurations are maintained.

Link Switchover

By default, a maximum of 64 links in a bundle can actively carry traffic on a Cisco CRS Router, If one member link in a bundle fails, traffic is redirected to the remaining operational member links.

On a Cisco CRS Router, you can optionally implement 1:1 link protection for a bundle by setting the **bundle maximum-active links** command to 1. By doing so, you designate one active link and one or more dedicated standby links. If the active link fails, a switchover occurs and a standby link immediately becomes active, thereby ensuring uninterrupted traffic.

If the active and standby links are running LACP, you can choose between an IEEE standard-based switchover (the default) or a faster proprietary optimized switchover. If the active and standby links are not running LACP, the proprietary optimized switchover option is used.

Regardless of the type of switchover you are using, you can disable the wait-while timer, which expedites the state negotiations of the standby link and causes a faster switchover from a failed active link to the standby link. To do so, you can use the **lacp fast-switchover** command.

How to Configure Link Bundling

This section contains the following procedures:

- Configuring Ethernet Link Bundles, page 196
- Configuring EFP Load Balancing on an Ethernet Link Bundle, page 201
- Configuring VLAN Bundles, page 203
- Configuring POS Link Bundles, page 209
- Configuring the Default LACP Short Period Time Interval, page 214
- Configuring Custom LACP Short Period Time Intervals, page 216

Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

Note MAC accounting is not supported on Ethernet link bundles.

Note In order for an Ethernet bundle to be active, you must perform the same configuration on both connection endpoints of the bundle.

SUMMARY STEPS

The creation of an Ethernet link bundle involves creating a bundle and adding member interfaces to that bundle, as shown in the steps that follow.

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface** {**GigabitEthernet | TenGigE**} *interface-path-id*
10. **bundle id** *bundle-id* [**mode** {**active | on | passive**}
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. Repeat Step 8 through Step 11 to add more links to the bundle you created in Step 2.
15. **end**
    or
    commit
16. **exit**
17. **exit**
18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-Ether** *bundle-id*
20. **show lacp Bundle-Ether** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates a new Ethernet link bundle with the specified bundle-id. The range is 1 to 65535. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand. **Note**
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.

	Command or Action	Purpose
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link. **Note**
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet link bundle.
Step 9	**interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the specified interface. Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the *rack/slot/module* format.
Step 10	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds the link to the specified bundle. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).
Step 11	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration submode for the Ethernet interface.

	Command or Action	Purpose

Step 14	**interface** {**GigabitEthernet** | **TenGigE**} *number*
	bundle id *bundle-id* [**mode** {**active** | **passive** | **on**}]
	no shutdown
	exit
	
	Example:
	`RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/1`
	
	`RP/0/RP0/CPU0:router(config-if)# bundle id 3`
	
	`RP/0/RP0/CPU0:router(config-if)# bundle port-priority 2`
	
	`RP/0/RP0/CPU0:router(config-if)# no shutdown`
	
	`RP/0/RP0/CPU0:router(config-if)# exit`
	
	`RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/3`
	
	`RP/0/RP0/CPU0:router(config-if)# bundle id 3`
	
	`RP/0/RP0/CPU0:router(config-if)# no shutdown`
	
	`RP/0/RP0/CPU0:router(config-if)# exit`
		(Optional) Repeat Step 8 through Step 11 to add more links to the bundle.

Step 15	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	Perform Step 1 through Step 15 on the remote end of the connection.	Brings up the other end of the link bundle.
Step 19	**show bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle.
Step 20	**show lacp bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show lacp bundle Bundle-Ether 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring EFP Load Balancing on an Ethernet Link Bundle

This section describes how to configure Ethernet flow point (EFP) Load Balancing on an Ethernet link bundle.

By default, Ethernet flow point (EFP) load balancing is enabled. However, the user can choose to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link. This configuration is available only on an Ethernet Bundle subinterface with Layer 2 transport (**l2transport**) enabled.

Note If the active members of the bundle change, the traffic for the bundle may get mapped to a different physical link that has a hash value that matches the configured value.

SUMMARY STEPS

Perform the following steps to configure EFP Load Balancing on an Ethernet link bundle:

1. **configure**
2. **interface Bundle-Ether** *bundle-id* **l2transport**
3. **bundle load-balance hash** *hash-value* [**auto**]
4. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **l2transport** **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport`	Creates a new Ethernet link bundle with the specified *bundle-id* and with Layer 2 transport enabled. The range is 1 to 65535.

Step 3	**bundle load-balance hash** *hash-value* [**auto**]
	
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1`
	or
	`RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto`
		Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.
		• *hash-value*—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
		• **auto**—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

Step 4	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring VLAN Bundles

This section describes how to configure a VLAN bundle. The creation of a VLAN bundle involves three main tasks:

1. Create an Ethernet bundle
2. Create VLAN subinterfaces and assign them to the Ethernet bundle.
3. Assign Ethernet links to the Ethernet bundle.

These tasks are describe in detail in the procedure that follows.

Note In order for a VLAN bundle to be active, you must perform the same configuration on both ends of the bundle connection.

Restrictions

The Cisco XR 12000 Series Router does not currently support 1:1 link protection. Therefore, the **bundle maximum-active links** and **lacp fast-switchover** commands are not supported on the Cisco XR 12000 Series Router.

SUMMARY STEPS

The creation of a VLAN link bundle is described in the steps that follow.

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface Bundle-Ether** *bundle-id.vlan-id*
10. **dot1q vlan** *vlan-id*
11. **ipv4 address** *ipv4-address mask*
12. **no shutdown**
13. **exit**
14. Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
15. **end**
    or
    commit
16. **exit**
17. **exit**
18. **configure**

19. **interface** {**GigabitEthernet | TenGigE**} *interface-path-id*
20. **bundle id** *bundle-id* [**mode** {**active | on | passive**}
21. **bundle port-priority** *priority*
22. **no shutdown**
23. Repeat Step19 through Step 21 to add more Ethernet Interfaces to the bundle you created in Step 2.
24. **end**
    or
    commit
25. Perform Step 1 through Step 23 on the remote end of the connection.
26. **show bundle Bundle-Ether** *bundle-id*
27. **show vlan interface**
28. **show vlan trunks** [{**GigabitEthernet | TenGigE | Bundle-Ether**} *interface-path-id*] *[***brief | summary**] [**location** *node-id]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3`	Creates and names a new Ethernet link bundle. This **interface Bundle-Ether** command enters you into the interface configuration submode, where you can enter interface-specific configuration commands. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the **ipv4 address** configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.

	Command or Action	Purpose
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link. **Note**
Step 8	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits the interface configuration submode.
Step 9	**interface Bundle-Ether** *bundle-id.vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1`	Creates a new VLAN, and assigns the VLAN to the Ethernet bundle you created in Step 2. Replace the *bundle-id* argument with the *bundle-id* you created in Step 2. Replace the *vlan-id* with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved). **Note** When you include the *.vlan-id* argument with the **interface Bundle-Ether** *bundle-id* command, you enter subinterface configuration mode.
Step 10	**dot1q vlan** *vlan-id* **Example:** `RP/0/RP0/CPU0:router#(config-subif)# dot1q vlan 10`	Assigns a VLAN to the subinterface. Replace the *vlan-id* argument with a subinterface identifier. Range is from 1 to 4094 inclusive (0 and 4095 are reserved).
Step 11	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24`	Assigns an IP address and subnet mask to the subinterface.

	Command or Action	Purpose
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router#(config-subif)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.

Step 14	Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.
	
	interface Bundle-Ether *bundle-id.vlan-id*
	dot1q vlan *vlan-id*
	
	ipv4 address *ipv4-address mask*
	no shutdown
	exit
	
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# interface Bundle-Ether 3.1`
	`RP/0/RP0/CPU0:router(config-subif)# dot1q vlan 20`
	`RP/0/RP0/CPU0:router(config-subif)# ipv4 address 20.2.3.4/24`
	`RP/0/RP0/CPU0:router(config-subif)# no shutdown`
	`exit`
		(Optional) Adds more subinterfaces to the bundle.

Step 15	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• 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 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-subif)# end`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	**configure** **Example:** `RP/0/RP0/CPU0:router # configure`	Enters global configuration mode.
Step 19	**interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0`	Enters interface configuration mode for the Ethernet interface you want to add to the Bundle. Enter the **GigabitEthernet** or **TenGigE** keyword to specify the interface type. Replace the *interface-path-id* argument with the node-id in the rack/slot/module format. **Note** A VLAN bundle is not active until you add an Ethernet interface on both ends of the link bundle.
Step 20	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds an Ethernet interface to the bundle you configured in Step 2 through Step 13. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the interface to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on** (LACP is not run over the port).
Step 21	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.
Step 22	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	(Optional) If a link is in the down state, bring it up. The **no shutdown** command returns the link to an up or down state depending on the configuration and state of the link.
Step 23	—	Repeat Step 19 through Step 21 to add more Ethernet interfaces to the VLAN bundle.

	Command or Action	Purpose

Step 24	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RP0/CPU0:router(config-subif)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 25	Perform Step 1 through Step 23 on the remote end of the VLAN bundle connection.	Brings up the other end of the link bundle.
Step 26	**show bundle Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3`	(Optional) Shows information about the specified Ethernet link bundle. The **show bundle Bundle-Ether** command displays information about the specified bundle. If your bundle has been configured properly and is carrying traffic, the State field in the **show bundle Bundle-Ether** command output shows the number “4,” which means the specified VLAN bundle port is “distributing.”
Step 27	**show vlan interface** **Example:** `RP/0/RP0/CPU0:router # show vlan interface`	Displays the current VLAN interface and status configuration.

	Command or Action	Purpose
Step 28	**show vlan trunks** [{**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id*] [**brief** | **summary**] [**location** *node-id*] **Example:** `RP/0/RP0/CPU0:router# show vlan trunk summary`	(Optional) Displays summary information about each of the VLAN trunk interfaces. • The keywords have the following meanings: – **brief—**Displays a brief summary. – **summary—**Displays a full summary. – **location—**Displays information about the VLAN trunk interface on the given slot. – **interface—**Displays information about the specified interface or subinterface. Use the **show vlan trunks** command to verify that all configured VLAN subinterfaces on an Ethernet bundle are "up."
Step 29	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.

Configuring POS Link Bundles

This section describes how to configure a POS link bundle.

Note In order for a POS bundle to be active, you must perform the same configuration on both connection endpoints of the POS bundle.

Restrictions

The Cisco XR 12000 Series Router currently does not support POS interfaces and POS link bundles.

SUMMARY STEPS

The creation of a bundled POS interface involves configuring both the bundle and the member interfaces, as shown in the following steps.

1. **configure**
2. **interface Bundle-POS** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*

6. **bundle maximum-active links** *links* [**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface POS** *number*
10. **bundle id** *bundle-id* [**mode** {**active | on | passive**}
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. Repeat Step 8 through Step 11 to add more links to the bundle you created in Step 2.
15. **end**
    or
    commit
16. **exit**
17. **exit**
18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-POS** *bundle-id*
20. **show lacp bundle Bundle-POS** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-POS** *bundle-id* **Example:** `RP/0/RP0/CPU0:router#(config)#interface Bundle-POS 2`	Configures and names the new bundled POS interface. This interface command will enter you into the interface configuration submode, from where interface specific configuration commands are entered. Use the **exit** command to exit from the interface configuration submode back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the ip address configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 620000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.

	Command or Action	Purpose
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	(Optional) Sets the number of active links required before you can bring up a specific bundle.
Step 6	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	(Optional) Implements 1:1 link protection for the bundle, which causes the highest-priority link in the bundle to become active and the second-highest-priority link to become the standby. Also, specifies that a switchover between active and standby LACP-enabled links is implemented per a proprietary optimization. **Note** The priority of the active and standby links is based on the value of the **bundle port-priority** command.
Step 7	**lacp fast-switchover** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.
Step 8	**exit**	Exits the interface configuration submode.
Step 9	**interface POS** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface POS 0/1/0/0`	Enters POS interface configuration mode and specifies the POS interface name and interface-path-id notation *rack/slot/module/port*.
Step 10	**bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle-id 3`	Adds the link to the specified bundle. To enable active or passive LACP on the bundle, include the optional **mode active** or **mode passive** keywords in the command string. To add the link to the bundle without LACP support, include the optional **mode on** keywords with the command string. **Note** If you do not specify the **mode** keyword, the default mode is **on (**LACP is not run over the port).
Step 11	**bundle port-priority** *priority* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1`	(Optional) If you set the **bundle maximum-active links** command to 1, you must also set the priority of the active link to the highest priority (lowest value) and the standby link to the second-highest priority (next lowest value). For example, you can set the priority of the active link to 1 and the standby link to 2.

	Command or Action	Purpose
Step 12	**no shutdown** **Example:** `RP/0/RP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration which forces the interface administratively down. The **no shutdown** command then returns the link to an up or down state, depending on the configuration and state of the link.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router# exit`	Exits the interface configuration submode for the POS interface.
Step 14	Repeat Step 8 through Step 11 to add more links to a bundle	(Optional) Adds more links to the bundle you created in Step 2.

Step 15	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 16	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 18	Perform Step 1 through Step 15 on the remote end of the connection.	Brings up the other end of the link bundle.

	Command or Action	Purpose
Step 19	**show bundle Bundle-POS** *number* **Example:** `RP/0/RP0/CPU0:router# show bundle Bundle-POS 1`	(Optional) Shows information about the specified POS link bundle.
Step 20	**show lacp bundle Bundle-POS** *bundle-id* **Example:** `RP/0/RP0/CPU0:router# show lacp bundle Bundle-POS 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring the Default LACP Short Period Time Interval

This section describes how to configure the default short period time interval for sending and receiving LACP packets on a Gigabit Ethernet interface. This procedure also enables the LACP short period.

SUMMARY STEPS

To enable an LACP short period time interval, using the default time of 1 second, perform the following steps.

1. **configure**
2. **interface GigabitEthernet** *interface-path*
3. **bundle id** *number* **mode active**
4. **lacp period short**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode.
Step 3	**bundle id** *number* **mode active** **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active`	Specifies the bundle interface and puts the member interface in active mode.

	Command or Action	Purpose
Step 4	**lacp period short** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short`	Configures a short period time interval for the sending and receiving of LACP packets, using the default time period of 1000 milliseconds or 1 second.

Step 5	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Custom LACP Short Period Time Intervals

This section describes how to configure custom short period time intervals (less than 1000 milliseconds) for sending and receiving LACP packets on a Gigabit Ethernet interface.

Note You must always configure the *transmit* interval at both ends of the connection before you configure the *receive* interval at either end of the connection. Failure to configure the *transmit* interval at both ends first results in route flapping (a route going up and down continuously). When you remove a custom LACP short period, you must do it in reverse order. You must remove the *receive* intervals first and then the *transmit* intervals.

SUMMARY STEPS

To configure custom receive and transmit intervals for LACP packets, perform the following steps.

Router A

1. **configure**
2. **interface GigabitEthernet** *interface-path*
3. **bundle id** *number* **mode active**
4. **lacp period short**
5. **commit**

Router B

6. **configure**
7. **interface GigabitEthernet** *interface-path*
8. **bundle id** *number* **mode active**
9. **lacp period short**
10. **commit**

Router A

11. **configure**
12. **interface GigabitEthernet** *interface-path*
13. **lacp period short transmit** *interval*
14. **commit**

Router B

15. **configure**
16. **interface GigabitEthernet** *interface-path*
17. **lacp period short transmit** *interval*
18. **commit**

Router A

19. **configure**
20. **interface GigabitEthernet** *interface-path*

21. **lacp period short receive** *interval*
22. **commit**

Router B

23. **configure**
24. **interface GigabitEthernet** *interface-path*
25. **lacp period short receive** *interval*
26. **commit** or **end**

DETAILED STEPS

	Command or Action	Purpose
	Router A	
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode.
Step 3	**bundle id** *number* **mode active** **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active`	Specifies the bundle interface and puts the member interface in active mode.
Step 4	**lacp period short** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short`	Enables the short period time interval.
Step 5	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes and exits to EXEC mode.
	Router B	
Step 6	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 7	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode.
Step 8	**bundle id** *number* **mode active** **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle id 1 mode active`	Specifies the bundle interface and puts the member interface in active mode.
Step 9	**lacp period short** **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short`	Enables the short period time interval.
Step 10	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes and exits to EXEC mode.
	Router A	
Step 11	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 12	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.
Step 13	**lacp period short transmit** *interval* **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short transmit 500`	Configures the short period transmit time interval for LACP packets at one end of the connection. Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.
Step 14	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes and exits to EXEC mode.
	Router B	
Step 15	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 16	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.
Step 17	**lacp period short transmit** *interval* **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short transmit 500`	Configures the short period transmit time interval for LACP packets at one end of the connection. Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.
Step 18	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes and exits to EXEC mode.
	Router A	
Step 19	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 20	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.
Step 21	**lacp period short receive** *interval* **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short receive 500`	Configures the short period receive time interval for LACP packets at one end of the connection. Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.
Step 22	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves configuration changes and exits to EXEC mode.
	Router B	
Step 23	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 24	**interface GigabitEthernet** *interface-path* **Example:** `RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/1`	Creates a Gigabit Ethernet interface and enters interface configuration mode at one end of the connection.
Step 25	**lacp period short receive** *interval* **Example:** `RP/0/RP0/CPU0:router(config-if)# lacp period short receive 500`	Configures the short period receive time interval for LACP packets at one end of the connection. Valid values are 100 to 1000 milliseconds in multiples of 100, such as 100, 200, 300, and so on.

Step 26	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Link Bundling

This section contains the following examples:

Configuring an Ethernet Link Bundle: Example

The following example shows how to join two ports to form an EtherChannel bundle running LACP:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# bundle maximum-active links 1 hot-standby
RP/0/RP0/CPU0:Router(config-if)# lacp fast-switchover
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RP0/CPU0:Router(config-if)# bundle port-priority 1
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config)# interface TenGigE 0/3/0/1
RP/0/RP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RP0/CPU0:Router(config-if)# bundle port-priority 2
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

Configuring a VLAN Link Bundle: Example

The following example shows how to create and bring up two VLANS on an Ethernet bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.1
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 20
RP/0/RP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# no shutdown
```

```
RP/0/RP0/CPU0:Router(config-subif)# exit
RP/0/RP0/CPU0:Router(config)#interface gig 0/1/5/7
RP/0/RP0/CPU0:Router(config-if)# bundle-id 1 mode act
RP/0/RP0/CPU0:Router(config-if)# commit
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router # show vlan trunks
```

Configuring a POS Link Bundle: Example

The following example shows how to join two ports to form a Packet-over-SONET (POS) link bundle:

```
RP/0/RP0/CPU0:Router# config
RP/0/RP0/CPU0:Router(config)# interface Bundle-POS 5
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface POS 0/0/1/0
RP/0/RP0/CPU0:Router(config-if)# bundle id 5
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
RP/0/RP0/CPU0:Router(config)# interface POS 0/0/1/1
RP/0/RP0/CPU0:Router(config-if)# bundle id 5
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

Configuring EFP Load Balancing on an Ethernet Link Bundle: Example

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through the same physical member link automatically.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto
RP/0/RP0/CPU0:router(config-subif)#
```

The following example shows how to configure all egressing traffic on the fixed members of a bundle to flow through a specified physical member link.

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1
RP/0/RP0/CPU0:router(config-subif)#
```

Configuring LACP Short Periods: Examples

The following example shows how to configure the LACP short period time interval to the default time of 1000 milliseconds (1 second):

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
   lacp period short
   commit
```

The following example shows how to configure custom LACP short period transmit and receive intervals to *less than* the default of 1000 milliseconds (1 second):

Router A

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
   lacp period short
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
   bundle id 1 mode active
   lacp period short
   commit
```

Router A

```
config
interface gigabitethernet 0/0/0/1
    lacp period short transmit 100
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
    lacp period short transmit 100
   commit
```

Router A

```
config
interface gigabitethernet 0/0/0/1
   lacp period short receive 100
   commit
```

Router B

```
config
interface gigabitethernet 0/0/0/1
   lacp period short receive 100
   commit
```

Additional References

The following sections provide references related to link bundle configuration.

Related Documents

Related Topic	Document Title
Cisco IOS XR master command reference	*Cisco IOS XR Master Commands List*
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*
Initial system bootup and configuration information for a router using the Cisco IOS XR software.	*Cisco IOS XR Getting Started Guide*
Information about user groups and task IDs	*Cisco IOS XR Interface and Hardware Component Command Reference*

Standards

Standards	Title
IEEE 802.3ad (incorporated as Annex 43 into 802.3-2002)	—

MIBs

MIBs	MIBs Link
The IEEE-defined MIB for Link Aggregation (defined in 802.3 Annex 30C)	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport