# Configure Link Bundling

*Table 1: Feature History Table*

| Feature Name | Release | Description |
|---|---|---|
| 1023 Ethernet Bundle Interfaces Support | Release 7.3.15 | With the introduction of this enhancement, the maximum system-wide bundle interface scale has increased from 512 to 1023 bundle interfaces. The default value remains at 64-member links for each bundle. |
| 64-bit Bandwidth Support | Release 7.3.15 | With this release, the Cisco 8000 Series Router supports 64-bit bandwidth field, as opposed to the previous 32-bit bandwidth field. 64-bit bandwidth enables the system to support interface bandwidths greater than 4.2T. |

This module describes the configuration of link bundle interfaces on the Cisco 8000 Series Router.

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

Each bundle has a single MAC, a single IP address, and a single configuration set (such as ACLs).

**Note** The router supports both Layer 2 and Layer 3 Link Bundles. If the Link Bundle is a Layer 3 interface, the system requires an IP address. If the Link Bundle is a Layer 2 interface, the system does not require an IP address. A Link Bundle on the router may contain Layer 2 and Layer 3 subinterfaces within it. In which case, the Layer 3 subinterfaces require IP addresses, but the Link Bundle interface does not require an IP address.

The router supports bundling for these types of interfaces:

- Ethernet interfaces

**Feature History for Configuring Link Bundling**

| Release | Modification |
|---|---|
| Release 7.0.11 | Support for this feature added on the router. |

| Release 7.2.1 | Mixed speed bundle members feature added on the router. |
|---|---|

# Limitations and Compatible Characteristics of Ethernet Link Bundles

This list describes the properties and limitations of ethernet link bundles:

- Starting with Cisco IOS XR Release 7.2.1, the router supports mixed speed bundles, allowing member links with different bandwidths to be included in the same bundle. The traffic distribution across bundle members is based on the bandwidth of each link. Mixed speed bundles are subject to a maximum bandwidth ratio of 10:1 between the fastest and slowest member links.

  For example, you can combine a 10 Gbps and a 100 Gbps link or a 100 Gbps and a 40 0Gbps link in the same bundle; however, a 10 Gbps and a 400 Gbps link cannot be bundled together. Load balancing is performed in proportion to the bandwidth of each member link. Typical valid combinations include:

  - 400G, 100G
  - 400G, 40G
  - 400G, 100G, 40G
  - 100G, 40G
  - 100G, 10G
  - 100G, 40G, 10G
  - 40G and 10G

  Additionally, the total weight of the bundle must not exceed 64.

- The weight of each bundle member is the ratio of its bandwidth to the lowest bandwidth member. Total weight of the bundle is the sum of weights or relative bandwidth of each bundle member. Since the weight for a bundle member is greater than or equal to 1 and less than or equal to 10, the total member of links in a bundle is less than 64 in mixed bundle case.

- Any type of Ethernet interfaces can be bundled, with or without the use of Link Aggregation Control Protocol (LACP).

- With Cisco IOS XR Release 7.3.15, a single router can support up to 1023 bundle interfaces, with each bundle accommodating up to 64 member links.

  If adding a new line card causes these limits to be exceeded, the system will experience continuous Out of Resource (OOR) failures. To resolve these errors, you must either reduce the scale or disable the affected line card.

- 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 ethernet link bundles.

- A bundle can be administratively enabled or disabled.

- Each individual link within a bundle can be administratively enabled or disabled.

- Ethernet link bundles are created in the same way as Ethernet channels, where the user enters the same configuration on both end systems.

- The MAC address that  is set on the bundle becomes the MAC address of the links within that bundle.

- Load balancing (the distribution of data between member links) is done by flow instead of by packet. Data is distributed to a link in proportion to the bandwidth of the link in relation to its bundle.

- QoS is supported and is applied proportionally on each bundle member.

- All links within a single bundle must terminate on the same two systems.

- Bundled interfaces are point-to-point.

- A link must be in the up state before it can be in distributing state in a bundle.

- Only physical links can be bundle members.

# Prerequisites for Configuring Link Bundling on a Router

Before configuring Link Bundling, ensure that you meet the following tasks and conditions:

- You know the interface IP address (Layer 3 only).

- You know the links that you must include in the bundle that you are configuring.

- If you are configuring an Ethernet link bundle, you must install Ethernet line cards on the router.

**Note**    For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco 8000 Series Router Hardware Installation Guide*.

# Information About Configuring Link Bundling

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

## 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. The system assigns a virtual interface to the bundled link. You can dynamically add and delete component links from the virtual interface.

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

A link bundle is a group of ports that the system bundles together and the group then acts as a single link. Following are the advantages of link bundles:

- Multiple links can span several line cards 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 flow on the available links, if one of the links within a bundle fails. You can add bandwidth without interrupting the packet flow.

Prior to Cisco IOS XR Software Release 7.3.15, the interface bandwidth was stored and processed as a 32-bit value, which supported bundles with an aggregate bandwidth of up to 4.2 Gbps (the sum of its members). Starting with Cisco IOS XR Software Release 7.3.15, the interface bandwidth is stored and processed as a 64-bit value. The 64-bit value supports significantly larger aggregate bandwidths, accommodating bundles with high-bandwidth members whose combined bandwidth can exceed 4.2 Tbps.

All the individual links within a single bundle must be of the same type.

Cisco IOS XR software supports the following methods of forming bundles of Ethernet 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. The system automatically removes the links from a bundle that are incompatible or have failed.

# Link Aggregation Through LACP

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. For the router, 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. The system analyzes these frames to ensure that both the systems are in agreement.

## IEEE 802.3ad Standard

The IEEE 802.3ad standard typically defines a method of forming Ethernet link bundles.

For each link configured as a 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 by using 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. This determines the compatibility of the links that are configured to be members of a bundle.

Bundle MAC addresses in the router come from a set of reserved MAC addresses in the backplane. This MAC address stays with the bundle as long as the bundle interface exists. The bundle uses this MAC address until you configure a different MAC address. The member links use the bundle MAC address when passing the 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.

# Configuring LACP Fallback

This section describes how to configure the LACP Fallback feature.

**Procedure**

**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 192.168.1.27 255.0.0.0
```

Specifies a primary IPv4 address for an interface.

**Step 4** **bundle lacp-fallback timeout 4** *number*

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# bundle lacp-fallback timeout 4
```

Enables the LACP Fallback feature.

**Step 5** **end** or **commit**

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# commit
```

Saves configuration changes.

**Step 6** **show bundle infrastructure database ma bdl-info Bundle-e1010 | inc** *text*

**Example:**

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1010 | inc "fallback"
```

(Optional) Shows the MA information of the bundle manager.

**Step 7** **show bundle infrastructure database ma bdl-info Bundle-e1015 | inc** *text*

**Example:**

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1015 | inc "fallback"
```

(Optional) Shows the MA information of the bundle manager.

# 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 that distributes traffic over multiple links that are based on certain parameters. The router support load balancing for all links in a bundle using Layer 2, Layer 3, and Layer 4 routing information. Starting with Cisco IOS XR Software Release 7.2.1, bandwidth based load-balancing is applicable to L3 unicast flows.

This section describes the load balancing support on link bundles.

For more information about other forms of load balancing on the router, see the following:

   • Per-flow load balancing on non-bundle interfaces using Layer 3 and 4 routing information.

   • Pseudowire (PW) Load Balancing beginning in Cisco IOS XR 4.0.1.

# Layer 3 Egress Load Balancing on Link Bundles

Layer 3 load balancing support began on the router in Cisco IOS XR 7.0.11 release.

Layer 3 load balancing for link bundles is enabled globally by default.

The ingress linecard does bundle member selection and forwards the packet to the linecard and network processor (NP) corresponding to the selected bundle member. The same hash value is used for both ingress and egress linecards. Therefore, even though the egress linecard also does bundle member selection, it selects the same bundle member that was selected by the ingress linecard.

### Multicast IPv4 and IPv6 Traffic

For outbound multicast IPv4 or IPv6 traffic, a set of egress linecards is predetermined by the system. If a bundle interface or bundle subinterface is an outgoing interface, the system selects the bundle member for each outgoing interface in a route based on the multicast group address. This helps with load distribution of multicast routed traffic to different bundle members, while providing traffic sequencing within a specific route.

The egress linecard does NP selection using the same approach, when bundle members are spread across multiple NPs within the egress linecard.

When the packet arrives on an egress NP, it uses the 5-tuple hash to select a bundle member within an NP for each packet. This provides better resiliency for bundle member state changes within an NP.

# 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.

**Procedure**

**Step 1** **configure**

**Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

**Step 2** **interface HundredGigE***interface-path*

**Example:**

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/1/0/1
```

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
```

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.

### Example

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

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

# Configuring Custom LACP Short Period Time Intervals

This section describes how to configure custom short period time interval for sending and receiving LACP packets on a Gigabit Ethernet interface.

**Procedure**

**Step 1**    **configure**

**Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

**Step 2**    **interface HundredGigE***interface-path*

**Example:**

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/1/0/1
```

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** *time-interval*

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# lacp period 300
```

Configures a custom period time interval for the sending and receiving of LACP packets. The interval can be in the range 100 to 1000 ms, in multiples of 100.

**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.

### Example

This example shows how to configure the LACP period time interval to the custom time of 300 milliseconds:

```
config
interface HundredGigE 0/1/0/1
   bundle id 1 mode active
   lacp period 300
   commit
```

# QoS and Link Bundling

On the router, when the system applies QoS on the bundle for either the ingress or egress direction, QoS is applied at each member interface. For complete information on configuring QoS on link bundles on the router, refer to the *Cisco 8000 Series Aggregation Services Router Modular Quality of Service Configuration Guide* and the *Cisco 8000 Series Aggregation Services Router Modular Quality of Service Command Reference*.

# Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration process. Ensure that you clear all previous network layer configuration before adding it to a bundle:

1. In global configuration mode, create a link bundle. To create an Ethernet link bundle, enter the **interface Bundle-Ether** command.

2. Assign an IP address and subnet mask to the virtual interface using the **ipv4 address** command.

3. Add interfaces to the bundle that 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.

**Note** The system configures a link as a member of a bundle from the interface configuration submode for that link.

# Nonstop Forwarding During Card Failover

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

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

**Note** Failover is always onto the standby RP.

You do not need to configure anything to guarantee that the system maintains the standby interface configurations.

# Link Failover

When one member link in a bundle fails, the system redirects the traffic to the remaining operational member links and traffic flow remains uninterrupted.

# Link Switchover

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

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 switchover suppress-flaps** command.

# LACP Fallback

The LACP Fallback feature allows an active LACP interface to establish a Link Aggregation Group (LAG) port-channel before the port-channel receives the Link Aggregation and Control Protocol (LACP) protocol data units (PDU) from its peer.

With the LACP Fallback feature configured, the router allows the server to bring up the LAG, before receiving any LACP PDUs from the server, and keeps one port active. This allows the server to establish a connection to PXE server over one Ethernet port, download its boot image and then continue the booting process. When the server boot process is complete, the server fully forms an LACP port-channel.

# Designate a Member Link as Unviable

*Table 2: Feature History Table*

| Feature Name | Release | Description |
| --- | --- | --- |
| Designate a Member Link as Unviable | Release 7.10.1 | Earlier, when a member link is added to an interface link bundle, the peer starts using the link as soon as the LACP communication is up. Sometimes, the hardware programming for the data-path does not get complete in this time resulting in packet loss without any notification to the source. |
| | | You can now mark a member link as unviable to introduce a delay during which the link is treated as standby. By delaying the usage of the member link for data transmission, you can ensure that the link configuration is fully established, which enables successful data transmission. |
| | | This feature introduces these changes: |
| | | • **CLI: forwarding-unviable** |
| | | • **YANG Data Model:** New XPath for `Cisco-IOS-XR-bundlemgr-oper` (see Github, YANG Data Models Navigator). |

Link bundling aggregates multiple physical links to a single logical link. When a member link is added to a link bundle, link aggregation control protocol (LACP) communication gets established with the peer to negotiate and control the link aggregation. LACP doesn't have any provision to incorporate a delay before letting data transmission over the link. Therefore, the peer starts using the link when the LACP communication is up. Occasionally, even though the link status is up, and LACP communication is up, the hardware programming for data-path packet forwarding doesn't get complete. In such scenarios, the transmitted data gets lost without any notification or error message to the source or destination of the traffic.

You can now delay the use of such member links, which aren't fully ready to handle data transmission, using the **forwarding-unviable** command. This command configures the link as forwarding-unviable and the member link is considered "standby" for bundle management. As standby member links of a bundle aren't used for data transmission, the usage of forwarding-unviable member links is delayed. When the member link is fully up, that is, the packet forwarding data-path is also completely programmed, you can disable forwarding-unviability of the link using **no forwarding-unviable** command. This removes the forwarding-unviable configuration of the link. Then, the link is treated as an "active" member of the bundle and is used in data transmission and load balancing.

**Note** It is recommended to wait for a few minutes before running **no forwarding-unviable** command to ensure that the packet forwarding data-path is completely programmed.

# Guidelines and Restrictions for Designating Member Links as Unviable

- Forwarding-unviable is disabled on all Ethernet interfaces by default. Therefore, by default, all member links in a bundle are considered "active".

- A link bundle is considered up, only if at least one member link is active. Only the active member links in the link bundle are used for data transmission, load balancing, and redundancy.

- If a link bundle has only one member link, which is forwarding-unviable, the bundle state is considered "down".

- If all the member links in a bundle are forwarding-unviable, the bundle state is considered "down".

- Other existing threshold parameters such as minimum-active links, maximum-active links, and maximum-active-bandwidth, which are considered to determine the bundle state, continue to function along with forwarding-unviable functionality. For more details on these parameters, see How to Configure Link Bundling, on page 14.

- There is no effect of forwarding-unviable configuration on individual Ethernet interfaces that are not part of a link bundle. That is, irrespective of the configuration, such non-member interfaces continue to attempt data transmission and reception.

# How to Configure Link Bundling

This section contains the following procedures:

## Configuring Ethernet Link Bundles

This section describes how to configure an Ethernet link bundle.

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

**Tip**  You can programmatically perform the configuration using `openconfig-lacp.yang`, `openconfig-if-aggregate.yang` OpenConfig data models, `Cisco-IOS-XR-bundlemgr-oper.yang` Cisco IOS XR native data model or `Cisco-IOS-XR-um-lacp-cfg.yang` Unified data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

**Procedure**

**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**
   • On the router, only a Layer 3 bundle interface requires an IP address.

**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.

**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**

**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.

**Step 14**    **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.

**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 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 described 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.

**Procedure**

**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 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 5**     **exit**

**Example:**

```
RP/0/RP0/CPU0:router(config-if)# exit
```

Exits the interface configuration submode.

**Step 6**     **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 7**  **encapsulation dot1q**

**Example:**

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
```

Sets the Layer 2 encapsulation of an interface.

**Step 8**  **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.

**Step 9**  **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 10**  **exit**

**Example:**

```
RP/0/RP0`/CPU0:router(config-subif)# exit
```

Exits subinterface configuration mode for the VLAN subinterface.

**Step 11**  Repeat Step 9 through Step 12 to add more VLANS to the bundle you created in Step 2.

(Optional) Adds more subinterfaces to the bundle.

**Step 12**  **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 13**     **exit**

**Example:**

```
RP/0/RP0/CPU0:router(config-subif)# end
```

Exits interface configuration mode.

**Step 14**     **exit**

**Example:**

```
RP/0/RP0/CPU0:router(config)# exit
```

Exits global configuration mode.

**Step 15**     **configure**

**Example:**

```
RP/0/RP0/CPU0:router # configure
```

Enters global configuration mode.

**Step 16**     **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 17**     **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.

# VLANs on an Ethernet Link Bundle

You can configure 802.1Q VLAN subinterfaces on 802.3ad Ethernet link bundles.

✎

**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, the system supports all VLAN subinterface configuration on that link bundle.

VLAN subinterfaces can support Ethernet Flow Points (EFPs) and Layer 3 services.

You can configure Layer 3 VLAN subinterfaces as follows:

```
interface bundle-ether instance.subinterface, encapsulation dot1q xxxxx
```

# Configuring a Member Link as Unviable

Perform the following task to designate a member link as unviable.

### Example Configuration

```
RP/0/RP0/CPU0:ios(config)#interface HundredGigE 0/0/0/34
RP/0/RP0/CPU0:ios(config-if)#forwarding-unviable
RP/0/RP0/CPU0:ios(config-if)#commit
RP/0/RP0/CPU0:ios(config-if)#end
```

### Running Configuration

```
RP/0/RP0/CPU0:ios#show running-config interface HundredGigE 0/0/0/34
Thu Apr 20 11:11:55.744 UTC
interface HundredGigE0/0/0/34
 forwarding-unviable
!
```

### Verification

Use **show bundle** command to view the forwarding-viable status of LAG members. Here, the interface HundredGigE 0/0/0/34 is added to ethernet bundle 3.

```
RP/0/RP0/CPU0:ios#show bundle
Thu Apr 20 11:29:42.500 UTC

Bundle-Ether3
  Status:                             Down
  Local links <active/standby/configured>:  0 / 0 / 1
  Local bandwidth <effective/available>:    0 (0) kbps
  MAC address (source):               78c6.9991.3504 (Chassis pool)
  Inter-chassis link:                 No
  Minimum active links / bandwidth:   1 / 1 kbps
  Maximum active links:               64
  Wait while timer:                   2000 ms
```

```
  Load balancing:
    Link order signaling:                    Not configured
    Hash type:                               Default
    Locality threshold:                      None
  LACP:                                      Operational
    Flap suppression timer:                  Off
    Cisco extensions:                        Disabled
    Non-revertive:                           Disabled
  mLACP:                                     Not configured
  IPv4 BFD:                                  Not configured
  IPv6 BFD:                                  Not configured

  Port                 Device          State       Port ID         B/W, kbps
  -------------------- --------------  ----------  --------------  ----------
  Hu0/0/0/34           Local           Standby     0x8000, 0x0001  100000000
      Link is not forwarding viable and in standby state
```

# Configuration Examples for Link Bundling

This section contains the following examples:

## Example: Configuring an Ethernet Link Bundle

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

```
RP/0/RP0/CPU0:Router(config)# config

RP/0/RP0/CPU0:Router(config-if)# 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)# exit
RP/0/RP0/CPU0:Router(config-if)# 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)# no shutdown
RP/0/RP0/CPU0:Router(config)# exit
RP/0/RP0/CPU0:Router(config-if)# 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)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# exit
```

This example shows the configuration in the case of a mixed speed bundle:

```
RP/0/RP0/CPU0:Router(config)# config

RP/0/RP0/CPU0:Router(config-if)# interface bundle-ether 50
RP/0/RP0/CPU0:Router(config-if)# root
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/11
RP/0/RP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/16
RP/0/RP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/27
RP/0/RP0/CPU0:Router(config-if)# bundleid 50 mode active
```

```
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# interface HundredGigE 0/6/0/1
RP/0/RP0/CPU0:Router(config-if)# bundleid 50 mode active
RP/0/RP0/CPU0:Router(config-if)# no shutdown
RP/0/RP0/CPU0:Router(config-if)# root
RP/0/RP0/CPU0:Router(config)# commit
RP/0/RP0/CPU0:Router(config)# end
```

The following output is shown for the **show bundle bundle-ether** command:

### show bundle bundle-ether50

```
Bundle-Ether50
 Status:                   Up
 Local links <active/standby/configured>:  4 / 0 / 4
 Local bandwidth <effective/available>:   130000000 (130000000) kbps
 MAC address (source):        0011.2233.4458 (Chassis pool)
 Inter-chassis link:         No
 Minimum active links / bandwidth:    1 / 1 kbps
 Maximum active links:         64
 Wait while timer:          2000 ms
 Load balancing:           Default
 LACP:                Operational
  Flap suppression timer:       Off
  Cisco extensions:          Disabled
 mLACP:                Not configured
 IPv4 BFD:               Not configured

 Port          Device      State     Port ID     B/W, kbps
 ------------------- --------------- ----------- -------------- ----------
 Te0/0/0/11     Local     Active     0x8000, 0x0002 10000000
   Link is Active
 Te0/0/0/16     Local     Active     0x8000, 0x0003 10000000
   Link is Active
 Te0/0/0/27     Local     Active     0x8000, 0x0004 10000000
   Link is Active
 Hu0/6/0/1      Local     Active     0x8000, 0x0001 100000000
   Link is Active
```

In order to view the weight of a mixed speed bundle, run the **show bundle load-balancing** command. The following is the truncated output of this command.

```
show bundle load-balancing bundle-ether50 location 0/0/cpu0

<snip>

Bundle-Ether50
 Type:        Ether (L3)
 Members <current/max>: 4/64
 Total Weighting:   13
 Load balance:    Default
 Locality threshold: 65
 Avoid rebalancing?  False
 Sub-interfaces:   1

 Member Information:
  Port:        LON ULID BW
  ------------------- --- ---- --
  Hu0/6/0/1       0   0  10
  Te0/0/0/11      1   1  1
  Te0/0/0/16      2   2  1
  Te0/0/0/27      3   3  1
```

```
 Platform Information:
 =====================

          * Bundle Summary Information *
           ------------------------

Interface        : Bundle-Ether50    Ifhandle       : 0x00000ce0
Lag ID         : 1           Virtual Port    : 255
Number of Members  : 4            Local to LC    : Yes
Hash Modulo Index   : 13
MGSCP Operational Mode : No

Member Information:
LON     Interface     ifhandle    SFP   port   slot remote/rack_id
-----   --------------  ----------  ---   ----   ---- --------------
0    Hu0/6/0/1    0x100001c0   648   116   8     0/0
1    Te0/0/0/11   0x04000380   65    9    2     0/0
2    Te0/0/0/16   0x040004c0   67    8    2     0/0
3    Te0/0/0/27   0x04000780   72    4    2     0/0

</snip>
```

# Example: Configuring a VLAN Link Bundle

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

```
RP/0/RP0/CPU0:Router(config-subif)# config
RP/0/RP0/CPU0:Router(config-subif)# interface Bundle-Ether 1
RP/0/RP0/CPU0:Router(config-ifsubif)# ipv4 address 1.2.3.4/24
RP/0/RP0/CPU0:Router(config-ifsubif)# bundle minimum-active bandwidth 620000
RP/0/RP0/CPU0:Router(config-if)# bundle minimum-active links
RP/0/RP0/CPU0:Router(config-ifsubif)# exit
RP/0/RP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RP0/CPU0:Router(config-subif)# interface Bundle-Ether 1.1
RP/0/RP0/CPU0:Router(config-subif)# encapsulation dot1q 100
RP/0/RP0/CPU0:Router(config-subif) # ip addr 10.2.3.4/24
RP/0/RP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RP0/CPU0:Router(config-subifif)# exit
RP/0/RP0/CPU0:Router(config-if)# interface Bundle-Ether 1.2
RP/0/RP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RP0/CPU0:Router(config-subif)Router # ip addr20.2.3.4/24

RP/0/RP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RP0/CPU0:Router(config-subifif)# 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
```