



## Configuring Link Bundling

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.

A link bundle is simply a group of ports that are bundled together and act as a single link. The advantages of link bundles are as follows:

- 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. Bandwidth can be added without interrupting packet flow.

Cisco IOS XR software supports the following method 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. Links that are incompatible or have failed are automatically removed from a bundle.
- [Limitations and Compatible Characteristics of Ethernet Link Bundles, on page 2](#)
- [Configuring Ethernet Link Bundles, on page 3](#)
- [VLANs on an Ethernet Link Bundle, on page 7](#)
- [Configuring VLAN over Bundles, on page 7](#)
- [Configuring Multichassis Link Aggregation Control Protocol Session, on page 12](#)
- [Configuring Multichassis Link Aggregation Control Protocol Bundle, on page 14](#)
- [Configuring One-way Pseudowire Redundancy in MC-LAG, on page 16](#)
- [Configuring VPWS Cross-Connects in MC-LAG, on page 17](#)
- [Configuring ICCP based Service Homing, on page 20](#)
- [Configuring VPLS in MC-LAG, on page 22](#)
- [Configuring Multichassis Link Aggregation: Example, on page 24](#)
- [Information About Configuring Link Bundling, on page 29](#)

# Limitations and Compatible Characteristics of Ethernet Link Bundles

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

- Any type of Ethernet interfaces can be bundled, with or without the use of LACP (Link Aggregation Control Protocol).
  - 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.
  - QoS is supported and is applied proportionally on each bundle member.
  - In case static MAC address is configured on a bundle-ether interface, the following limitations are applied:
    - Locally generated packets, such as ICMP, BGP, and so on, going out from the interface have the source MAC address as the statically configured MAC address.
    - Transit (forwarded) packets going out of the interface do not have the configured static MAC as source MAC address. In such a scenario, the upper 36-bits come from the system MAC address (or the original/dynamic MAC address) and the lower 12-bits come from the MAC address configured on the bundle. To check the dynamic pool of MAC addresses included, use the `show ethernet mac-allocation detail` command.
- For example, if the dynamic MAC address was 008A.9624.48D8 and the configured static MAC address is 0011.2222.ABCD. Then, the source MAC for transit (forwarded) traffic will be 008A.9624.4BCD.



---

**Note** This limitation can cause traffic blackholing for the transit traffic, in case there is L2 ACL applied for security purpose. In such case, it is necessary to add permit statement for both MAC addresses in the L2 ACL.

---

- 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.
- 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.

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

## 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-if-aggregate.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco NCS 5000 Series Routers*.

### SUMMARY STEPS

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. **exit**
8. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
9. **bundle port-priority** *priority*
10. **no shutdown**
11. **exit**
12. **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit**
13. **end** or **commit**
14. **exit**
15. **exit**
16. Perform Step 1 through Step 15 on the remote end of the connection.
17. **show bundle Bundle-Ether** *bundle-id*
18. **show lacp Bundle-Ether** *bundle-id*

### DETAILED STEPS

**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**     • 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**     **exit**

**Example:**

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

Exits interface configuration submode for the Ethernet link bundle.

**Step 8**      **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]**Example:**

```
RP/0/RP0/CPU0:router(config-if)# bundle id 3 mode on
```

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 9**      **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 10**      **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 11**      **exit****Example:**

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

Exits interface configuration submode for the Ethernet interface.

**Step 12**      **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit****Example:**

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

```
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 TenGigE 0/1/0/1
```

```
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 13 **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 14 **exit**

#### **Example:**

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

Exits interface configuration mode.

### Step 15 **exit**

#### **Example:**

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

Exits global configuration mode.

### Step 16 Perform Step 1 through Step 15 on the remote end of the connection. Brings up the other end of the link bundle.

### Step 17 **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 18** `show lacp Bundle-Ether bundle-id`

**Example:**

```
RP/0/RP0/CPU0:router# show lacp Bundle-Ether 3
```

(Optional) Shows detailed information about LACP ports and their peers.

## 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:

- There is no separate limit defined for Layer 3 sub-interfaces on a bundle. However, an overall system limit of 4000 is applicable for NCS5001 and NCS5002, while a limit of 2000 is applicable for NCS5011.



**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.

Layer 2 EFPs are configured as follows:

```
interface bundle-ether instance.subinterface l2transport. encapsulation dot1q xxxxx
```

Layer 3 VLAN subinterfaces are configured as follows:

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



**Note** The difference between the Layer 2 and Layer 3 interfaces is the **l2transport** keyword. Both types of interfaces use **dot1q encapsulation**.

## Configuring VLAN over Bundles

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

## SUMMARY STEPS

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

## DETAILED STEPS

- 
- Step 1** Create an Ethernet bundle.
- Step 2** Create VLAN subinterfaces and assign them to the Ethernet bundle.
- Step 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.

---

## SUMMARY STEPS

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. **exit**
8. **interface Bundle-Ether** *bundle-id.vlan-id*
9. **encapsulation dot1q***vlan-id*
10. **ipv4 address** *ipv4-address mask*
11. **no shutdown**
12. **exit**
13. Repeat Step 9 through Step 12 to add more VLANs to the bundle you created in Step 2.
14. **end** or **commit**
15. **exit**
16. **exit**
17. **configure**
18. **interface** {**TenGigE** | **FortyGigE** | **HundredGigE**}*interface-path-id*

## DETAILED STEPS

- 
- 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.

**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**      **exit**

**Example:**

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

Exits the interface configuration submode.

**Step 8**      **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 9**      **encapsulation dot1q***vlan-id***Example:**

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

Sets the Layer 2 encapsulation of an interface.

**Step 10**    **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 11**    **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 12**    **exit****Example:**

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

Exits subinterface configuration mode for the VLAN subinterface.

**Step 13**    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 14**    **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 15 **exit**

##### **Example:**

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

Exits interface configuration mode.

#### Step 16 **exit**

##### **Example:**

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

Exits global configuration mode.

#### Step 17 **configure**

##### **Example:**

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

Enters global configuration mode.

#### Step 18 **interface {TenGigE | FortyGigE | HundredGigE} interface-path-id**

##### **Example:**

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 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.

# Configuring Multichassis Link Aggregation Control Protocol Session

Perform this task to enable a Multichassis Link Aggregation Control Protocol (mLACP) session.

## SUMMARY STEPS

1. **configure**
2. **redundancy iccp group** *group-id*
3. **mlacp system mac** *mac-id*
4. **mlacp system priority** *priority*
5. **mlacp node** *node-id*
6. **end** or **commit**

## DETAILED STEPS

### Step 1 **configure**

**Example:**

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

Enters global configuration mode.

### Step 2 **redundancy iccp group** *group-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# redundancy iccp group 100
```

Adds an ICCP redundancy group.

### Step 3 **mlacp system mac** *mac-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# mlacp system mac 1.1.1
```

Configures the LACP system ID to be used in this ICCP Group.

**Note** • The *mac-id* is a user configured value for the LACP system LAG-ID to be used by the POAs. It is highly recommended that the *mac-ids* have the same value on both POAs. You can have different LAG-IDs for different groups.

### Step 4 **mlacp system priority** *priority*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# mlacp system priority 10
```

Sets the LACP system priority to be used in this ICCP Group.

- Note**
- It is recommended that system priority of the POAs be configured to a lower numerical value (higher priority) than the LACP LAG ID of the DHD. If the DHD has higher system priority then dynamic priority management cannot work and brute force switchover is automatically used.

#### Step 5 **mlacp node** *node-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# mlacp node 1
```

Sets the LACP system priority to be used in this ICCP Group.

- Note**
- The *node-id* must be unique for each POA.

#### Step 6 **end** or **commit**

**Example:**

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

or

```
RP/0/RSP0/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 Multichassis Link Aggregation Control Protocol Bundle

Perform this task to configure a Multichassis Link Aggregation Control Protocol (mLACP) bundle.

## SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **mac-address** *mac-id*
4. **bundle wait-while** *milliseconds*
5. **lACP switchover suppress-flaps** *milliseconds*
6. **mlACP iccp-group** *group-id*
7. **mlACP port-priority** *priority*
8. **end** or **commit**

## DETAILED STEPS

---

### Step 1 **configure**

#### Example:

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

Enters global configuration mode.

### Step 2 **interface Bundle-Ether** *bundle-id*

#### Example:

```
RP/0/RSP0/CPU0:router#(config)# interface Bundle-Ether 3
```

Creates and names a new Ethernet link bundle.

### Step 3 **mac-address** *mac-id*

#### Example:

```
RP/0/RSP0/CPU0:router#(config-if)# mac-address 1.1.1
```

Sets the MAC address on the interface.

**Note** • Configuring the same MAC address on both POAs is highly recommended.

### Step 4 **bundle wait-while** *milliseconds*

#### Example:

```
RP/0/RSP0/CPU0:router#(config-if)# bundle wait-while 100
```

Sets the wait-while timeout for members of this bundle.

**Step 5** **lACP switchover suppress-flaps** *milliseconds***Example:**

```
RP/0/RSP0/CPU0:router#(config-if)# lacp switchover suppress-flaps 300
```

Sets the time for which to suppress flaps during a LACP switchover.

- Note**
- It is recommended that the value used for the *milliseconds* argument is greater than that for the wait-while timer of the local device (and DHD).

**Step 6** **mLACP iccp-group** *group-id***Example:**

```
RP/0/RSP0/CPU0:router#(config-if)# mlacp iccp-group 10
```

Configures the ICCP redundancy group in which this bundle should operate.

**Step 7** **mLACP port-priority** *priority***Example:**

```
RP/0/RSP0/CPU0:router#(config-if)# mlacp port-priority 10
```

Sets the starting priority for all member links on this device when running mLACP.

- Note**
- Lower value indicates higher priority. If you are using dynamic priority management the priority of the links change when switchovers occur.

**Step 8** **end** or **commit****Example:**

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

or

```
RP/0/RSP0/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)?`
- 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 One-way Pseudowire Redundancy in MC-LAG

Perform this task to allow one-way pseudowire redundancy behavior when the redundancy group is configured.

### SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **pw-class** {*class-name*}
4. **encapsulation mpls**
5. **redundancy one-way**
6. **end** or **commit**

### DETAILED STEPS

---

#### Step 1 **configure**

##### **Example:**

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

Enters global configuration mode.

#### Step 2 **l2vpn**

##### **Example:**

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

```
RP/0/RSP0/CPU0:router(config-l2vpn)#
```

Enters L2VPN configuration mode.

#### Step 3 **pw-class** {*class-name*}

##### **Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class class1
```

Configures the pseudowire class template name to use for the pseudowire.

#### Step 4 **encapsulation mpls**

##### **Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls
```

Configures the pseudowire encapsulation to MPLS.

**Step 5**      **redundancy one-way****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-pwc-mpls)# redundancy one-way
```

Configures one-way PW redundancy behavior.

**Note**      • The **redundancy one-way** command is effective only if the redundancy group is configured.

**Step 6**      **end or commit****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# end
```

or

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# 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 VPWS Cross-Connects in MC-LAG

Perform this task to configure VPWS cross-connects in MC-LAG.

**SUMMARY STEPS**

1. **configure**
2. **l2vpn**
3. **pw-status**
4. **xconnect group** *group-name*
5. **p2p** *xconnect-name*
6. **interface type** *interface-path-id*

7. **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
8. **pw-class** *{class-name}*
9. **backup neighbor** *A.B.C.D* **pw-id** *pseudowire-id*
10. **pw-class** *{class-name}*
11. **end** or **commit**

## DETAILED STEPS

### Step 1 **configure**

#### Example:

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

Enters global configuration mode.

### Step 2 **l2vpn**

#### Example:

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters L2VPN configuration mode.

### Step 3 **pw-status**

#### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-status
```

Enables pseudowire status.

- Note**
- When the attachment circuit changes redundancy state to Active, Active pw-status is sent over the primary and backup pseudowires.
  - When the attachment circuit changes redundancy state to Standby, Standby pw-status is sent over the primary and backup pseudowires.

### Step 4 **xconnect group** *group-name*

#### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group grp_1
```

Enters the name of the cross-connect group.

### Step 5 **p2p** *xconnect-name*

#### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p p1
```

Enters a name for the point-to-point cross-connect.

### Step 6 **interface** *type interface-path-id*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
```

Specifies the interface type ID.

**Step 7** **neighbor A.B.C.D pw-id pseudowire-id****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000
```

Configures the pseudowire segment for the cross-connect.

Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN.

**Step 8** **pw-class {class-name}****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class c1
```

Configures the pseudowire class template name to use for the pseudowire.

**Step 9** **backup neighbor A.B.C.D pw-id pseudowire-id****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 2000
```

Adds a backup pseudowire.

**Step 10** **pw-class {class-name}****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# pw-class c2
```

Configures the pseudowire class template name to use for the backup pseudowire.

**Step 11** **end or commit****Example:**

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# end
```

or

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# 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 ICCP based Service Homing

Perform this task to configure ICCP-SM.

### Before you begin

You must have configured ICCP as shown in the procedure *Configuring Interchassis Communication Protocol*.

### SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **redundancy iccp group** *group-id*
4. **multi-homing node-id** *node-id*
5. **mac-flush** *type*
6. **interface** *type interface-path-id*
7. **primary vlan** *{vlan range}*
8. **secondary vlan** *{vlan range}*
9. **end** or **commit**

### DETAILED STEPS

---

#### Step 1 **configure**

##### Example:

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

Enters global configuration mode.

#### Step 2 **l2vpn**

##### Example:

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

```
RP/0/RSP0/CPU0:router(config-l2vpn)#
```

Enters L2VPN configuration mode.

#### Step 3 **redundancy iccp group** *group-id*

**Example:**

```
RP/0/RSP0/CPU0:router#(config-l2vpn)# redundancy iccp group 100
```

Enables L2VPN redundancy mode and enters redundancy configuration submode. Adds an ICCP redundancy group.

**Step 4**    **multi-homing node-id** *node-id***Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-red-grp)# multi-homing node-id 1
```

Enter the pseudo MLACP node ID. Enables the ICCP based multi-homing service. The node-ID is used for ICCP signaling arbitration.

**Step 5**    **mac-flush** *type***Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-red-grp)# mac-flush stp-tcn
```

Specifies the type of MAC flush, either stp tcn or mvrp (default).

**Step 6**    **interface** *type interface-path-id***Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-reg-grp)# interface Bundle-Ether 1
```

Specifies the interface type ID. It can be a physical port name or the main bundle name (sub-port is not allowed). It can be any physical Ethernet or bundle Ethernet interface connecting to a dual homed CE device.

Only bundle-ethernet main ports are allowed for ICCP-SM. If you want to use this feature on a single ethernet link, then you must configure a bundle with that link.

**Step 7**    **primary vlan** *{vlan range}***Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-reg-grp)# primary vlan 1-10
```

Configures the list of VLANs under the main port, which default to active (forwarding) when there are no faults detected.

Specify the list of of comma separated VLAN ranges or individual VLANs.

**Step 8**    **secondary vlan** *{vlan range}***Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-reg-grp)# secondary vlan 11-20
```

Configures the list of VLANs under the main port, which default to standby (blocked) when there are no faults detected.

Specify the list of of comma separated VLAN ranges or individual VLANs.

**Step 9**    **end** or **commit****Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# end
```

or

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# 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)?`
- 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.

**Note** You can use the **show iccp group**, **show l2vpn iccp-sm** and **show lacp bundle-ether** commands to monitor ICCP-SM.

## Configuring VPLS in MC-LAG

Perform this task to configure VPLS in MC-LAG.

### SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **pw-status**
4. **bridge group** *bridge-group-name*
5. **bridge-domain** *bridge-domain-name*
6. **interface** *type interface-path-id*
7. **vfi** *{vfi-name}*
8. **neighbor** *A.B.C.D pw-id pseudowire-id*
9. **pw-class** *{class-name}*
10. **end** or **commit**

### DETAILED STEPS

**Step 1** **configure**

**Example:**

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

Enters global configuration mode.

## Step 2 **l2vpn**

### Example:

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters L2VPN configuration mode.

## Step 3 **pw-status**

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-status
```

(Optional) Enables pseudowire status.

All the pseudowires in the VFI are always active, independent of the attachment circuit redundancy state.

## Step 4 **bridge group** *bridge-group-name*

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group cisco
RP/0/RSP0/CPU0:router(config-l2vpn-bg)#
```

Creates a bridge group so that it can contain bridge domains and then assigns network interfaces to the bridge domain.

## Step 5 **bridge-domain** *bridge-domain-name*

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain abc
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)#
```

Establishes a bridge domain and enters L2VPN bridge group bridge domain configuration mode.

## Step 6 **interface** *type interface-path-id*

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface Bundle-Ether 1.1
```

Specifies the interface type ID.

## Step 7 **vfi** *{vfi-name}*

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# vfi vfi-east
```

Enters virtual forwarding instance (VFI) configuration mode.

## Step 8 **neighbor** *A.B.C.D pw-id pseudowire-id*

### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi)# neighbor 10.2.2.2 pw-id 2000
```

Configures the pseudowire segment for the cross-connect.

Optionally, you can disable the control word or set the transport-type to Ethernet or VLAN.

### Step 9 **pw-class** {*class-name*}

#### Example:

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# pw-class canada
```

Configures the pseudowire class template name to use for the pseudowire.

### Step 10 **end** or **commit**

#### Example:

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# end
```

or

```
RP/0/RSP0RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# 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 Multichassis Link Aggregation: Example

This example shows how to configure POAs:

### Active PoA

```
l2vpn bridge group bg1
bridge-domain bd1
neighbor 101.101.101.101 pw-id 5000
pw-class class1
backup neighbor 102.102.102.102 pw-id 3000
pw-class class1
!
```

!

### Standby PoA

```
interface Bundle-Ether10
mlacp iccp-group 1
mlacp port-priority 20
```

This example shows how to configure ICCP:

```
redundancy iccp group
member neighbor 1.2.3.4
backbone interface TenGigE 0/0/0/0
isolation recovery-delay 30
```

This example shows how to configure mLACP:

```
configure
redundancy iccp group 100
mlacp system mac 1.1.1
mlacp system priority 10
mlacp node 1
interface Bundle-Ether 3
mac-address 1.1.1
bundle wait-while 100
lacp switchover suppress-flaps 300
mlacp iccp-group 100
```

This example illustrates a switchover:

```
RP/0/0/CPU0:router# show bundle
```

```
Bundle-Ether1
Status: Up
Local links <active/standby/configured>: 1 / 0 / 1
Local bandwidth <effective/available>: 1000000 (1000000) kbps
MAC address (source): 0000.deaf.0000 (Configured)
Minimum active links / bandwidth: 1 / 1 kbps
Maximum active links: 64
Wait while timer: 100 ms
LACP: Operational
Flap suppression timer: 300 ms
mLACP: Operational
ICCP Group: 1
Role: Active
Foreign links <active/configured>: 0 / 1
Switchover type: Non-revertive
Recovery delay: 300 s
Maximize threshold: Not configured
IPv4 BFD: Not configured
```

```
Port Device State Port ID B/W, kbps
```

```
-----
Te0/0/0/0 Local Active 0x8001, 0x9001 1000000
Link is Active
Te0/0/0/0 5.4.3.2 Standby 0x8002, 0xa001 1000000
Link is marked as Standby by mLACP peer
```

```
RP/0/0/CPU0:router#mlacp switchover Bundle-Ether 1
```

This will trigger the peer device (Node 5.4.3.2 in IG 1) to become active for Bundle-Ether1.  
This may result in packet loss on the specified bundle.

```
Proceed with switch over? [confirm]
```

```
RP/0/0/CPU0:Jan 31 23:46:44.666 : BM-DISTRIB[282]: %L2-BM-5-MLACP_BUNDLE_ACTIVE : This
device is no longer the active device for Bundle-Ether1
RP/0/0/CPU0:Jan 31 23:46:44.668 : BM-DISTRIB[282]: %L2-BM-6-ACTIVE : TenGigE0/0/0/0 is no
longer Active as part of Bundle-Ether1
(Not enough links available to meet minimum-active threshold)
```

```
RP/0/0/CPU0:router#show bundle
Mon Jun 7 06:04:17.778 PDT
```

```
Bundle-Ether1
Status: mLACP hot standby
Local links <active/standby/configured>: 0 / 1 / 1
Local bandwidth <effective/available>: 0 (0) kbps
MAC address (source): 0000.deaf.0000 (Configured)
Minimum active links / bandwidth: 1 / 1 kbps
Maximum active links: 64
Wait while timer: 100 ms
LACP: Operational
Flap suppression timer: 300 ms
mLACP: Operational
ICCP Group: 1
Role: Standby
Foreign links <active/configured>: 1 / 1
Switchover type: Non-revertive
Recovery delay: 300 s
Maximize threshold: Not configured
IPv4 BFD: Not configured
```

```
Port Device State Port ID B/W, kbps
-----
Te0/0/0/0 Local Standby 0x8003, 0x9001 1000000
mLACP peer is active
Te0/0/0/0 5.4.3.2 Active 0x8002, 0xa001 1000000
Link is Active
RP/0/0/CPU0:router#
```

This example shows how to add a backup pseudowire to a VPLS access pseudowire:

```
l2vpn bridge group bg1
bridge-domain bdl
neighbor 101.101.101.101 pw-id 5000
pw-class class1
backup neighbor 102.102.102.102 pw-id 3000
pw-class class1
!
!
!
!
```

This example shows how to configure one-way pseudowire redundancy behavior when redundancy group is configured:

This example illustrates an overall MC-LAG configuration:

Topology:

DHD POA 1 POA 2

```

Te0/0/0/0 ----- Te0/0/0/0
Te0/0/0/1 ----- Te0/0/0/1
Te0/0/0/2
Te0/0/0/3 ----- Te0/0/0/0
Te0/0/0/4 ----- Te0/0/0/1
Te0/0/0/2 Te0/0/0/2
Te0/0/0/3 ----- Te0/0/0/3
Te0/0/0/4 ----- Te0/0/0/4

```

On POA 1:

```

redundancy
iccp
group 1
mlacp node 1
mlacp system mac 000d.000e.000f
mlacp system priority 1
member
neighbor 5.4.3.2
!
!
!
!
interface Bundle-Ether1
lacp switchover suppress-flaps 300
mlacp iccp-group 1
mac-address 0.deaf.0
bundle wait-while 100
!
interface Loopback0
ipv4 address 5.4.3.1 255.255.255.255
!
interface TenGigE0/0/0/0
description Connected to DHD Te0/0/0/0
bundle id 1 mode active
lacp period short
no shutdown
!
interface tenGigE0/0/0/3
description Connected to POA2 Te0/0/0/3
ipv4 address 1.2.3.1 255.255.255.0
proxy-arp
no shutdown
!
router static
address-family ipv4 unicast
5.4.3.2/32 1.2.3.2
!
!
mpls ldp
router-id 5.4.3.1
discovery targeted-hello accept
log
neighbor
!

```

```

interface TenGigE0/0/0/3
!
!
On POA 2:

redundancy
iccp
group 1
mlacp node 2
mlacp system mac 000d.000e.000f
mlacp system priority 1
member
neighbor 5.4.3.1
!
!
!
!
interface Bundle-Ether1
lacp switchover suppress-flaps 300
mlacp iccp-group 1
mac-address 0.deaf.0
bundle wait-while 100
!
interface Loopback0
ipv4 address 5.4.3.2 255.255.255.255
!
interface TenGigE0/0/0/0
description Connected to DHD Te0/0/0/3
bundle id 1 mode active
lacp period short
no shutdown
!
interface TenGigE0/0/0/3
description Connected to POA1 Te0/0/0/3
ipv4 address 1.2.3.2 255.255.255.0
proxy-arp
no shutdown
!
router static
address-family ipv4 unicast
5.4.3.1/32 1.2.3.1
!
!
mpls ldp
router-id 5.4.3.2
discovery targeted-hello accept
log
neighbor
!
interface TenGigE0/0/0/3
!
!
On the DHD:

interface Bundle-Ether1
lacp switchover suppress-flaps 300
bundle wait-while 100
!
interface TenGigE0/0/0/0
description Connected to POA1 Te0/0/0/0
bundle id 1 mode active
lacp period short

```

```
no shutdown
!  
interface TenGigE0/0/0/3  
description Connected to POA2 Te0/0/0/0  
bundle id 1 mode active  
lacp period short  
no shutdown  
!
```

## Information About Configuring Link Bundling

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

### IEEE 802.3ad Standard

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

---

## 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.
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 32 links to a single bundle.

4. You can 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.

---

## 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 fast-switchover** command.

## Multichassis Link Aggregation

The Multichassis Link Aggregation (MC-LAG) feature provides an end to end interchassis redundancy solution for the Carrier Ethernet Networks. MC-LAG involves two devices collaborating to act as a single LAG from the perspective of a (third) connected device, thus providing device-level as well as link-level redundancy.

To achieve this, two devices co-ordinate with each other to present a single LACP bundle (spanning the two devices) to a partner device. Only one of the devices forwards traffic at any one time, eliminating the risk of

forwarding loops. When a failure occurs, these devices coordinate to perform a switchover, changing the device on which traffic is being forwarded by manipulating the link LACP states.

The existing pseudowire redundancy in the core network coordinates with the redundancy in the access network based on:

- Multichassis Link Aggregation Control Protocol (mLACP)
- Interchassis Communication Protocol (ICCP)

The mLACP protocol defines the expected behavior between the two devices and uses the Interchassis Control Protocol (ICCP) to exchange TLVs and identify peer devices to operate with. At the edge of a provider's network, a simple customer edge (CE) device that only supports standard LACP is connected to two provider edge (PE) devices. Thus the CE device is dual-homed, providing better L2 redundancy from the provider's side. In mLACP terminology, the CE device is referred to as a dual-homed device (DHD) and each PE device is known as a point of attachment (POA). The POA forwarding traffic for the bundle is the active device for that bundle, while the other POA is the standby device.

For information on MC-LAG Active/Active, refer to the *L2VPN and Ethernet Services Configuration Guide*.

## Failure Cases

MC-LAG provides redundancy, switching traffic to the unaffected POA while presenting an unchanged bundle interface to the DHD, for these failure events:

- Link failure: A port or link between the DHD and one of the POAs fails.
- Device failure: Meltdown or reload of one of the POAs, with total loss of connectivity (to the DHD, the core and the other POA).
- Core isolation: A POA loses its connectivity to the core network, and therefore is of no value, being unable to forward traffic to or from the DHD.

A loss of connectivity between the POAs leads both devices to assume that the other has experienced device failure, causing them to attempt to take on the Active role. This is known as a split brain scenario and can happen in either of the following cases:

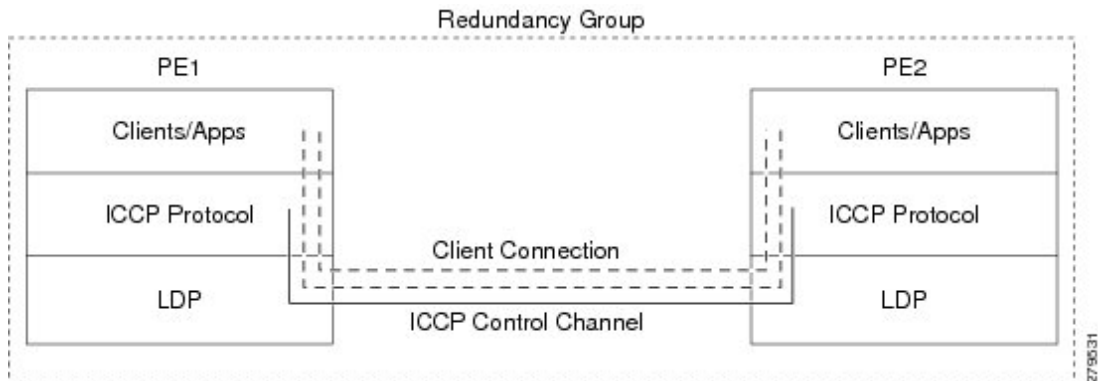
- All other connectivity remains; only the link between POAs is lost.
- One POA is isolated from the core network (i.e. a core isolation scenario where the connection between the two POAs was over the core network).

MC-LAG by itself does not provide a means to avoid this situation; resiliency in the connection between the POAs is a requirement. The DHD is given the responsibility of mitigating the problem by setting a limit on the number of links, within the bundle, that can be active. As such only the links connected to one of the POAs can be active at any one point of time.

## Interchassis Communication Protocol

This figure shows the graphical representation of the Interchassis Communication Protocol (ICCP).

Figure 1: ICCP Protocol



Two POAs communicate with each other over an LDP link using the Interchassis Communication Protocol (ICCP). ICCP is an LDP based protocol wherein an LDP session is created between the POAs in a redundancy group, and the ICCP messages are carried over that LDP session. The PE routers in a redundancy group may be a single-hop (directly connected) or a multi-hop away from one another. The ICCP protocol manages the setup and controls the redundancy groups. It also establishes, maintains, and tears down ICCP connections. The ICCP protocol uses route-watch to monitor the connectivity to the PEs in a given redundancy group. It is also responsible for tracking core isolation failures. It notifies all client applications of failure (core isolation and active PE failure).

To operate ICCP, the devices are configured as members of redundancy groups (RGs).



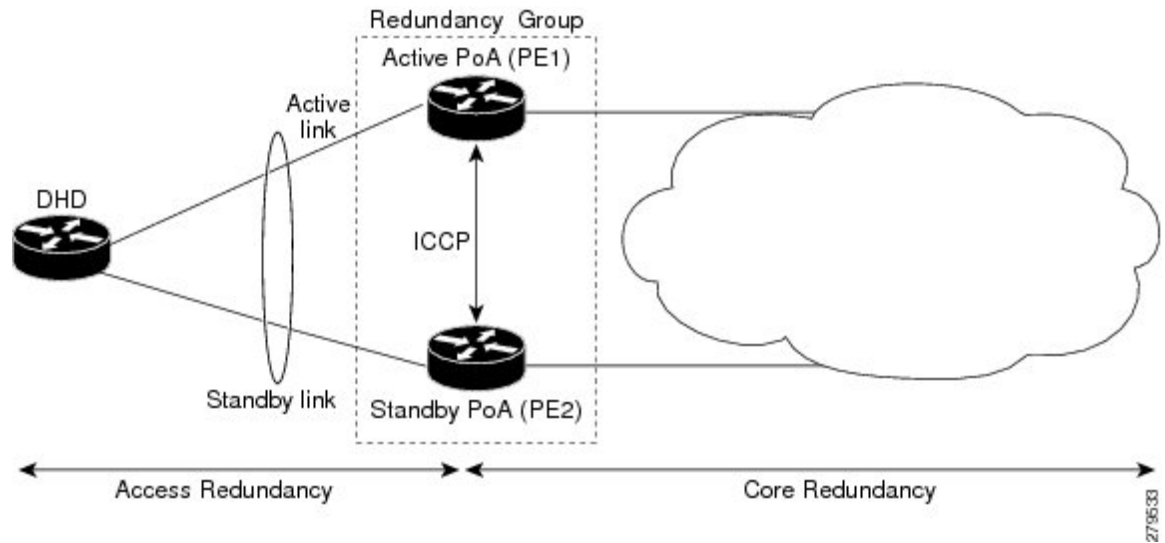
**Note** In the mLACP configuration, two devices are configured to be members of each RG (until a device-level failure occurs leaving only a single member). However, each device can be a member of more than one RG.

In each redundancy group, a POA's mLACP peer is the other POA in that group, with which it communicates using mLACP over ICCP. For each bundle, the POA and DHD at each end are LACP partners, communicating using the standard LACP protocol.

## Access Network Redundancy Model

The Multichassis Link Aggregation Control Protocol (mLACP) based redundancy between the customer edge device (CE) or access network and the provider edge (PE) device is achieved by allowing the CE to be connected to two PE routers. The two PE routers synchronize the data through ICCP; therefore they appear as a single device to the CE.

Figure 2: mLACP/ICCP Redundancy Model



The CE is also called dual-homed device (DHD) and the PE is also called point of attachment (POA). The pair of POAs that is connected to the single DHD forms a redundancy group (RG).

At any given time, only one POA is active for a bundle. Only the set of links between the DHD and the active POA actively sends traffic. The set of links between the DHD and the standby POA does not forward traffic. When the multichassis link bundle software detects that the connection to the active POA has failed, the software triggers the standby POA to become the active POA, and the traffic flows using the links between the DHD and newly active POA.

The ICCP protocol operates between the active and the standby POAs, and allows the POAs to coordinate their configuration, determine which POA is active, and trigger a POA to become active. Applications running on the two POAs (mLACP, IGMP snooping, DHCP snooping or ANCP) synchronize their state using ICCP.

## ICCP Based Service Multihoming

In the case of ICCP based Service Multihoming (ICCP-SM), the CE device uses two independent bundle interfaces to connect to the PoAs. Although bundle interfaces are used, they are not aggregated across the two chassis, and mLACP is not involved in the communication. The CE device configures the bundle interfaces in such a manner that all VLANs are allowed on both bundles. You can manually configure the PoAs to distribute the VLANs across the two bundles in order that individual VLANs are active(forwarding) on one bundle or PoA, and standby (blocked) on the other. The CE device initially floods a traffic flow on both bundles and learns the MAC address on the interface where it receives the response.

With ICCP-SM, you are not limited to a dual homed device. The access links can connect to a dual homed network (DHN) that are separate devices in the access network. The two bundles on the DHD or the DHN must be in a bridge domain so that L2 learning selects the link with the active set of VLANs.

Figure 3: ICCP Based Service Multihoming

If a bundle interface between the CE and the PoA fails, ICCP-SM on the PoA with the failed bundle communicates through ICCP to the other PoA's ICCP-SM. This activates the standby VLANs on the remaining bundle. A MAC flush is sent to the CE so that packets destined to hosts on the failed bundle are again flooded, in order to be learned on the newly activated bundle. The MAC flush is required because it is possible that the bundle interface failure is not detected by the CE.

In ICCP Based Service Multihoming, the total set of VLANs are split into a primary set and a secondary set and are configured on each PoA such that the primary set on one PoA is configured as secondary on the other. On each PoA, the VLANs are associated with ACs. If the VLANs are primary on a PoA and there are no faults, the associated ACs are set to forwarding. If the VLANs are secondary on a PoA, the associated ACs are blocked. ICCP-SM is only supported in VPLS cores.

## Advantages of Pseudo mLACP:

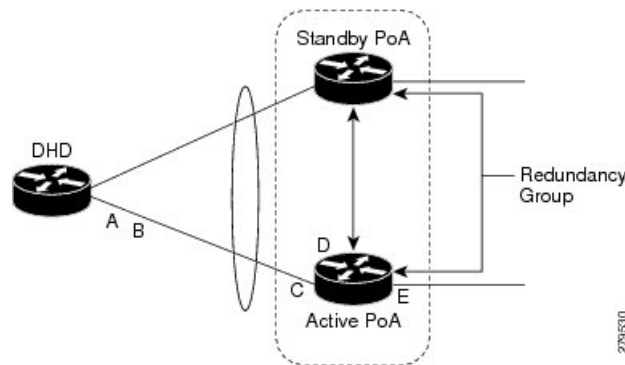
Pseudo mLACP has these three major advantages over mLACP:

- Pseudo mLACP can support a Dual Homed Network (DHN), while mLACP can only support a Dual Homed Device (DHD).
- Pseudo mLACP supports per-VLAN active/active redundancy without any load-balancing requirements on the CE.
- Pseudo mLACP does not require LACP support from the DHD, or DHN. It is independent of the access redundancy mechanism; therefore, it provides a network based redundancy solution. It allows maximum flexibility for the PE-CE interoperability in terms of dual-homing redundancy and recovery.

## Failure Modes

The mLACP feature provides network resiliency by protecting against port, link, and node failures. This figure depicts the various failure modes.

**Figure 4: Failure Modes**



These are the failure categories:

- A—DHD uplink port failure. The port on the DHD that is connected to the POA fails.
- B—DHD uplink failure. The connection between the DHD and the POA fails.
- C—Active POA downlink port failure.
- D—Active POA node failure.
- E—Active POA uplink failure (network isolation). The links between the active POA and the core network fails



**Note** ICCP Based Service Multihoming is similar to MC-LAG in the case of core network failures. It is revertive in nature. In the case of a failure, the PoA whose link has been restored activates the VLANs that are configured as primary.

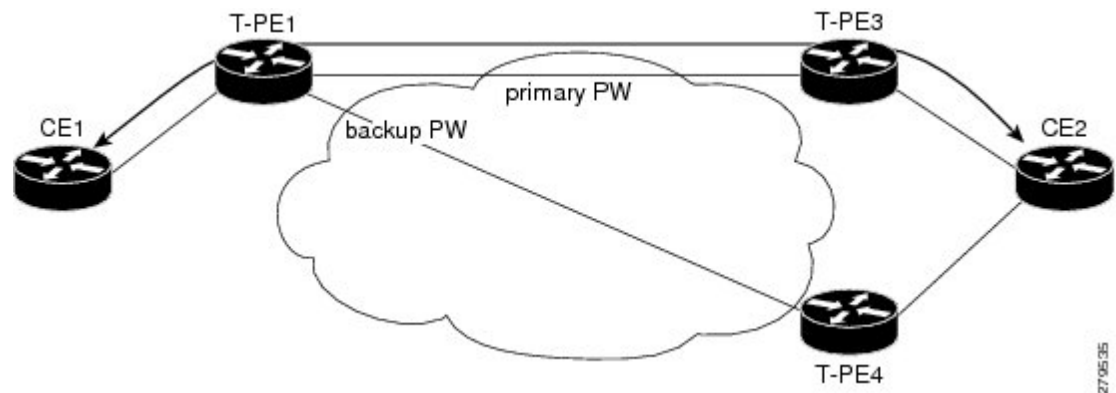
## Core Network Redundancy Model

This section explains:

### One-way Pseudowire Redundancy

This figure shows the VPWS one-way pseudowire redundancy model. Only one end of the pseudowire is protected by a backup pseudowire.

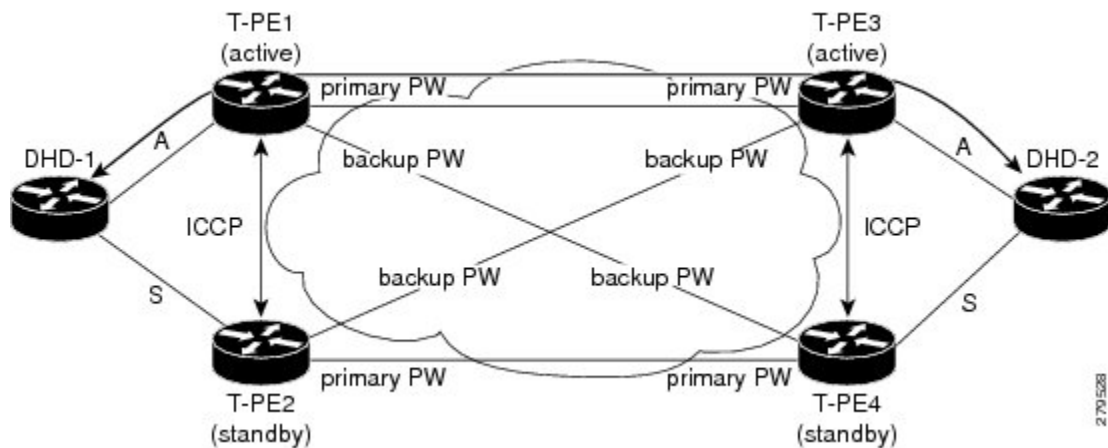
**Figure 5: VPWS one-way Pseudowire Redundancy**



### Two-way Pseudowire Redundancy

This figure shows the VPWS two-way pseudowire redundancy model. In this topology, each T-PE at the end of a PW has a primary and a backup PW. The state of the PW is coordinated with the state of the mLACP link between the DHD and the PE.

Figure 6: VPWS two-way Pseudowire Redundancy



## Switchovers

Switchovers, which is changing the Active/Standby roles of the POAs, are performed using dynamic priority management or brute force behavior.

## Dynamic Priority Management

Dynamic Priority Management involves co-ordination between the POAs to manipulate the LACP port priorities of their member links. Two priority values are tracked for each links:

- A configured priority which can either be configured explicitly, or defaults to 32768
- An operational priority used in LACP negotiations, which may differ from the configured priority if switchovers have occurred.

Higher priority LACP links are always selected ahead of lower priority LACP links. This means the operational priorities can be manipulated to force the standard LACP Selection Logic (on the POAs and on the DHD) to select desired links on both ends.

For example, consider a case where the DHD has two links to each POA, and each POA is configured with minimum-active links is 2. (This means the bundle goes down on the POA if the number of active links falls below 2.) The operational priorities for the member links are 1 on POA-1 and 2 on POA-2. This means that POA-1 is active (being higher priority) and the links on POA-2 are held in Standby state. The sequence of events in a switchover is as follows:

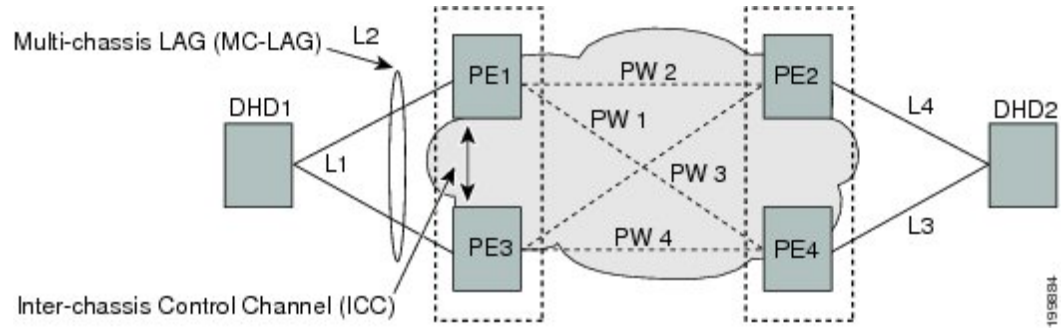
1. A link fails on POA-1, causing the number of active links to fall below the minimum of 2.
2. POA-1 changes the operational priority of both its links to 3, so the links on POA 2 are now higher priority.
3. POA-1 sends a LACP message to the DHD and an mLACP message to POA-2, informing both devices of the change.
4. The DHD tries to activate the links connected to POA-2 as these now have the highest priority.
5. POA-2 also ensures that its links have the highest priority and activates its links to the DHD.

At this point the switchover is complete.

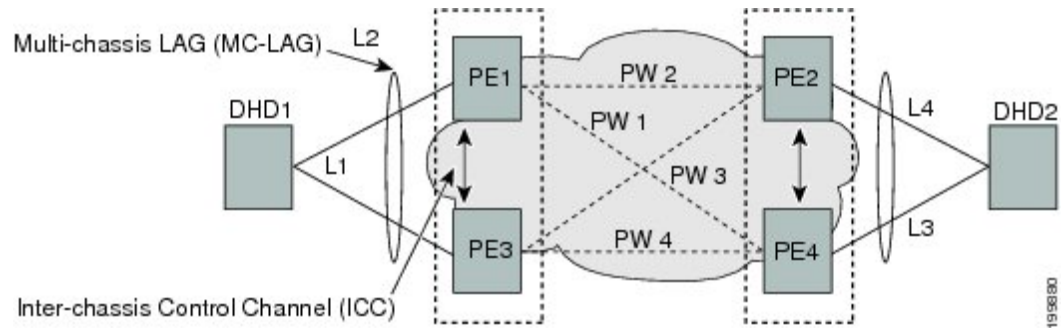
## MC-LAG Topologies

This section illustrates the supported MC-LAG topologies.

**Figure 7: VPWS One-way Pseudowire Redundancy in Redundancy Group**



**Figure 8: VPWS Two-way Pseudowire Redundancy**



**Figure 9: VPLS Pseudowires in One Redundancy Group**

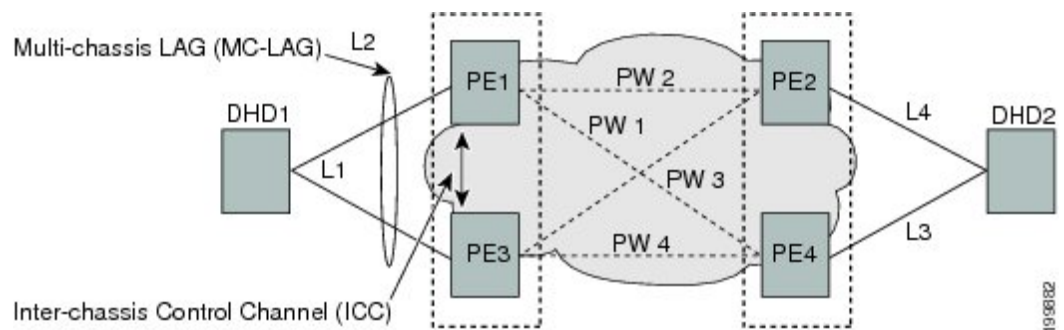


Figure 10: VPLS Pseudowires in Two Redundancy

