Configuring Link Bundling

This module describes the configuration of link bundle interfaces on the Cisco ASR 9000 Series Aggregation Services Routers.

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). POS link bundles do not have mac address, only ethernet link bundles have mac address.

Note The Cisco ASR 9000 Series Router supports both Layer 2 and Layer 3 Link Bundles. If the Link Bundle is a Layer 3 interface, an IP address is required. If the Link Bundle is a Layer 2 interface, an IP address is not required. A Link Bundle on the Cisco ASR 9000 Series 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. POS Link bundling is supported only on Layer 3 link bundles.

The Cisco ASR 9000 Series Router supports bundling for these types of interfaces:

- Ethernet interfaces and

Feature History for Configuring Link Bundling

Release	Modification
Release 3.7.2	This feature was introduced on the Cisco ASR 9000 Series Router.
Release 3.9.0	Support for load balancing was added. Bundle member links are put into new err-disable link interface status and admin-down protocol state when a bundle interface is shut down.
Release 3.9.1	Support for Layer 3 load balancing on Layer 2 link bundles was added.
Release 4.0.0	The following support was added: • Up to a maximum of 64 member links per bundle. • IPv6 addressing. • Multichassis Link Aggregation.

Release 4.0.1	Support for Dynamic Load Balancing for Link Aggregation (LAG) members was added. The **hw-module load-balance bundle l2-service l3-params** command is replaced by the **load-balancing flow** command in L2VPN configuration mode. For more information see the Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide and Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference.
Release 4.1.0	Support for Multi-Gigabit Service Control Point was added.
Release 4.2.0	Support for Link bundling for POS interfaces was added.
Release 4.3.1	Support for ICCP Based Service Multihoming was included.
Release 5.1.1	Support for Multichassis Link Aggregation on IPv6 traffic was included.
Release 5.1.2	Support for mixed speed member links in a bundle interface was included.
Release 6.0.0	Bundle scale support increases to 1600 bundles on A9K-RSP880-SE, A99-RP2-SE and the third generation of ASR 9000 Ethernet line card.
Release 6.2.2	Support for Layer 3 Multicast traffic over mixed speed bundles was added.

- Prerequisites for Configuring Link Bundling, on page 2
- Information About Configuring Link Bundling, on page 3
- How to Configure Link Bundling, on page 20
- Configuring Multichassis Link Aggregation, on page 33
- How to Configure MGSCP, on page 49
- Configuration Examples for Link Bundling, on page 57
- Configuration Examples for MGSCP, on page 64

Prerequisites for Configuring Link Bundling

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

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

Prerequisites for Configuring Link Bundling on a Cisco ASR 9000 Series Router

Before configuring Link Bundling, be sure that the following tasks and conditions are met:

- You know the interface IP address (Layer 3 only).
- You know which links should be included in the bundle you are configuring.
- If you are configuring an Ethernet link bundle, you must have Ethernet line cards installed in the router.

Note Link bundling is supported on all flavours of ASR 9000 line cards.

- The POS link bundling feature is supported on the following shared port adaptors (SPA):
  - 2-port OC-48 POS/SDH SPA
  - 4-port OC-48 POS/SDH SPA
  - 1-port OC-192 POS/XFP SPA
  - 4-port OC-3 POS-V2 SPA
  - 8-port OC-3 POS/SDH SPA
  - 8-port OC-12 POS/SDH SPA

Note For more information about physical interfaces, PLIMs, and modular services cards, refer to the *Cisco ASR 9000 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. 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.

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

For example, a bundle can contain all Ethernet interfaces, or it can contain all POS interfaces, but it cannot contain Ethernet and POS interfaces at the same time.

Cisco IOS XR software supports the following methods of forming bundles of Ethernet 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.
- Ether Channel or POS Channel—Cisco proprietary technology that allows the user to configure links to join a bundle, but has no mechanisms to check whether the links in a bundle are compatible.(EtherChannel applies to Ethernet interfaces, and POS Channel applies to POS interfaces.)

Features and Compatible Characteristics of 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).
- Bundle membership can span across several line cards that are installed in a single router or multiple routers in the case of MC-LAG.
- An ethernet link bundle can support a maximum of 64 physical links. If you add more than 64 links to a bundle, only 64 of the links are in distributing state, and the remaining links are in waiting state.
- A single Cisco ASR 9000 Series Router supports a maximum of 1600 bundles.
- Cisco ASR 9000 Series Router supports mixed speed bundles. Mixed speed bundles allow member links of different bandwidth to be configured as active members in a single bundle. The ratio of the bandwidth for bundle members must not exceed 10. Also, the total weight of the bundle must not exceed 64. For example, 100Gbps link and 10Gbps links can be active members in a bundle. Mixed speed bundles also allow:
  - Load-balancing on member links based on bandwidth weightage.
  - Support of proportional load-balancing for all unicast flows.
  - Support of load-balancing based on bandwidth weightage for all VPLS flooding and Layer 2 multicast flows.
- Multi-Gigabit Service Control Point (MGSCP) is not supported for mixed speed bundles.
- With mixed speed bundles, the total weight of a bundle is greater than the number of members in the bundle, as the weight represents the smallest active number bandwidth.
- 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.
- Mixed speed bundles are supported in MC-LAG.
- 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. Beginning in Cisco IOS XR Release 3.9.0, when you shut down a bundle interface, the member links are put into err-disable link interface status and

admin-down line protocol state. You can show the status of a bundle interface and its members using the **show interfaces** command.

- Each individual link within a bundle can be administratively enabled or disabled.
- 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.
- When LACP configured, each link within a bundle can be configured to allow different keepalive periods on different members.
- Load balancing (the distribution of data between member links) is done by flow instead of by packet. 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.
- Link layer protocols, such as CDP and HDLC keepalives, work independently on each link within a bundle.
- Upper layer protocols, such as routing updates and hellos, are sent over any member link of an ethernet interface bundle.
- All links within a single bundle must terminate on the same two systems. Both systems must be directly connected except in the case of MC-LAG.
- Bundled interfaces are point-to-point.
- A link must be in the up state before it can be in distributing state in a bundle.
- All links within a single bundle must be configured either to run 802.3ad (LACP) or Etherchannel (non-LACP). Mixed links within a single bundle are not supported.
- A bundle interface can contain physical links and VLAN subinterfaces only. Tunnels cannot be bundle members.
- Access Control List (ACL) configuration on link bundles is identical to ACL configuration on regular interfaces.
- Multicast traffic is load balanced over the members of a bundle. For a given flow, the control plane selects the member link, and all traffic for that flow is sent over that member. The system supports Layer 3 Multicast traffic over mixed speed bundles.
- When the LACP system receives priority value as zero (0) from a remote device, bundle does not come up.

Characteristics of POS Link Bundles in Cisco ASR 9000 Series Router

This section lists the properties of POS link bundles that are specific to Cisco ASR 9000 Series Router:

- Each bundle has to be configured between a pair of directly connected systems.
- All members of a bundle must be POS.
- POS link bundling can support up to 32 physical links if they are in the same speed. If links are in different speed, it cannot reach 32 physical links.
- Only physical interfaces can become bundle members.

- All bundles must be statically configured.
- Only cHDLC encapsulation type is currently supported on POS Link Bundle.
- Only POS SPA is supported for POS Link Bundling and not channelized SPA.
- Upper layer protocols, such as routing updates and hellos, are sent over through the bundle interface.
- Bandwidths for policers and queues must be in percentage and not in absolute values.
- Queue-limit must be in time unit and not in bytes.
- For POS link bundles, different link speeds are allowed within a single bundle, with a maximum of four times the speed difference between the members of the bundle. This means that only up to 4 times the bandwidth ratio is supported.

Restrictions of POS Link Bundles in Cisco ASR 9000 Series Router

This section lists the limitations of POS link bundles that are specific to Cisco ASR 9000 Series Router:

- LACP is not supported for POS link bundles in Cisco IOS XR Release 4.2.0 and later releases.
- IPv6 and ACL are not supported for POS link bundels in Cisco IOS XR Release 4.2.0.
- Multicast routing is not supported for POS link bundles in Cisco IOS XR Release 4.2.0.

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 Cisco ASR 9000 Series 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. These frames are analyzed to ensure both systems are in agreement.

IEEE 802.3ad Standard

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

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.

Bundle MAC addresses in the Cisco ASR 9000 Series 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 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.

Non Revertive Behavior for LACP Bundle Interface

In LACP, by default, a higher priority port would become the active port after it becomes operational again. To avoid this reversion, you can run the **lacp non-revertive** command. This configures the lower priority port to continue as the active port even after the higher priority port is capable of being operational. This avoids the traffic disruption that may happen in putting the currently active but lower priority port into standby and diverting traffic through the higher priority port that is now capable of being operational.

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.

Note MC-LAG in EVPN Multihoming-ELAN is not supported and alternative EVPN port-active should be used.

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:

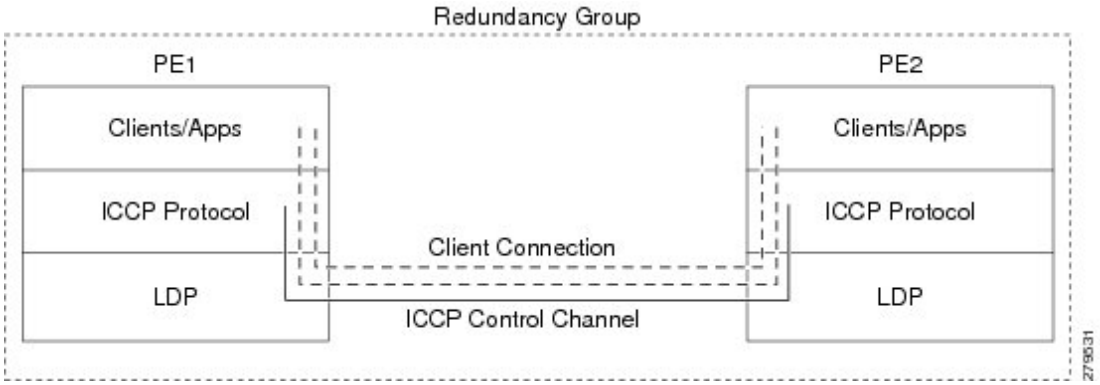
- All other connectivity remains; only the link between POAs is lost.
- One POA is isolated from the core network (i.e. a core isolation scenario where the connection between the two POAs was over the core network).

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).

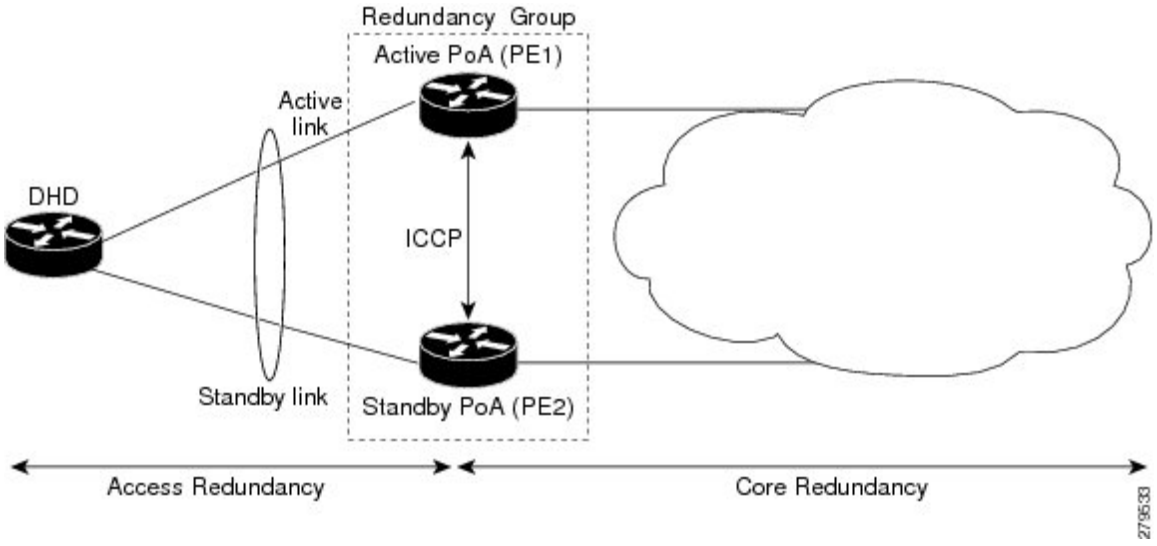
Note In the mLACP configuration, two devices are configured to be members of each RG (until a device-level failure occurs leaving only a single member). However, each device can be a member of more than one RG.

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.

Note When the connection to the active POA fails and the active POA is in the Negotiation state, the standby POA does not take over.

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.

Note While upgrading to a newer version of Cisco ASR 9000 router, ensure that you upgrade the active and standby nodes to the same Cisco IOS XR version.

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.

ICCP-SM Redundancy Group

Prior to Release 6.4.1, ICCP-SM configurations were required to match on both peers for traffic flow. Primary VLANs on one PoA were configured as secondary VLANs on the other. If there was a mismatch in VLAN configuration, all primary and secondary VLANs were blocked. Also, when a user configured a new VLAN, there was a temporary mismatch which blocked other services until the VLAN configuration on both sides matched.

Starting from Release 6.4.1, when there is a mismatch in VLAN configuration on PoAs, instead of blocking all primary and secondary VLANs, only unmatched VLANs are blocked. That is, if at least one primary VLAN on one PoA matches the secondary VLAN on the other PoA, traffic flows. If misconfiguration is detected while there is no port failure, each PoA re-evaluates the configuration. The PoAs move misconfigured VLANs to the secondary VLAN list while the primary VLANs forward traffic. The misconfigured VLAN list also includes those VLANs that are only configured remotely. For example, if VLAN 1 is only configured on remote PoA and is not configured locally, VLAN 1 will still be in the secondary list locally.

When two PoAs are connected, if one PoA has been configured with Release 6.4.1 image and the other PoA has been configured with an image earlier to Release 6.4.1, the behavior of the PoAs with respect to forwarding and blocking the traffic is different from each other. The PoA with an image earlier to Release 6.4.1 blocks all VLANs when there is a mismatch in VLAN configuration. The PoA with Release 6.4.1 image forwards primary VLANs that are matched in secondary list on the peer PoA.

Advantages of Pseudo mLACP:

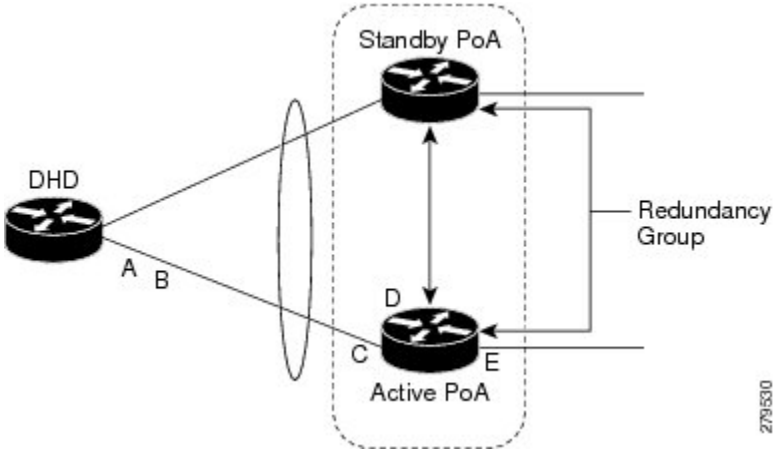
Pseudo mLACP has these three major advantages over mLACP:

- 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.

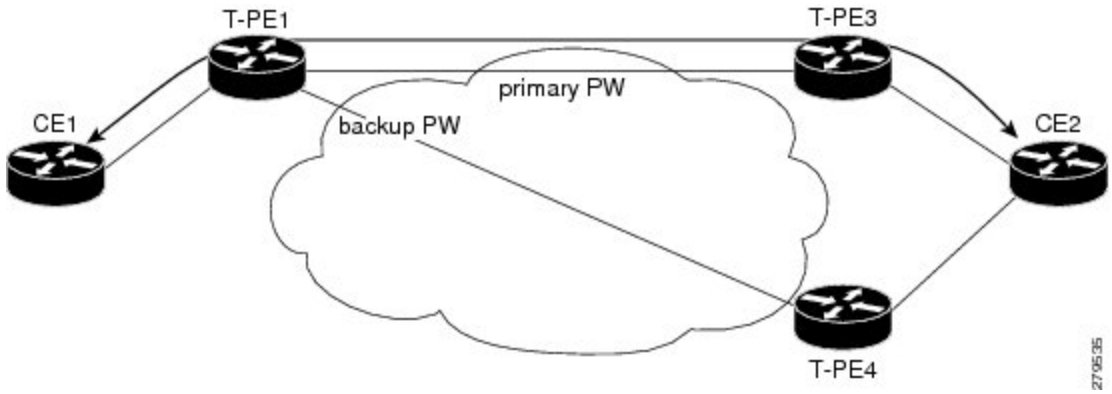
Core Network Redundancy Model

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.

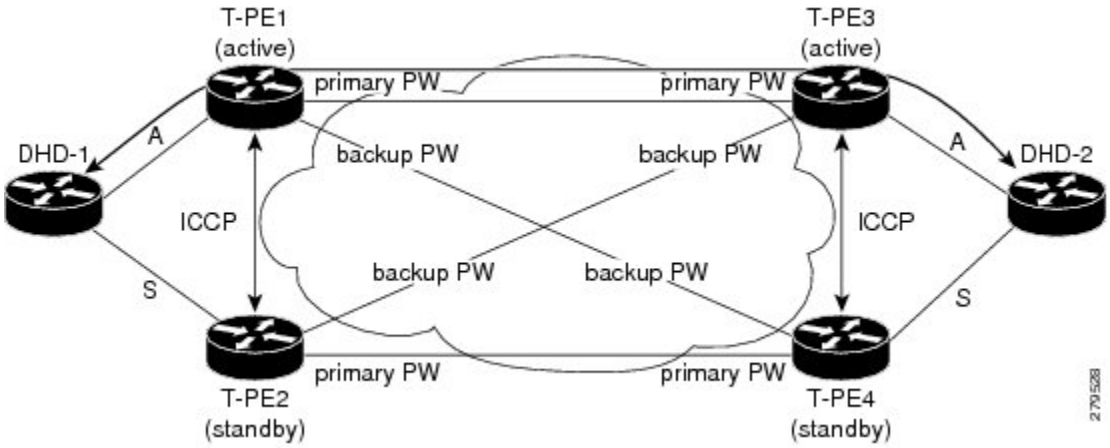
Figure 5: VPWS one-way Pseudowire Redundancy



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.

Brute Force Behavior

In a brute force switchover, port priorities are not modified. Instead the failing POA sends a single *dying gasp* to the DHD over LACP, forcing it to deselect the link. It then terminates LACP communications on that link. This only leaves links between the DHD and POA-2, as links that can be selected. So, both ends select those links.

MC-LAG Topologies

This section illustrates the supported MC-LAG topologies.

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

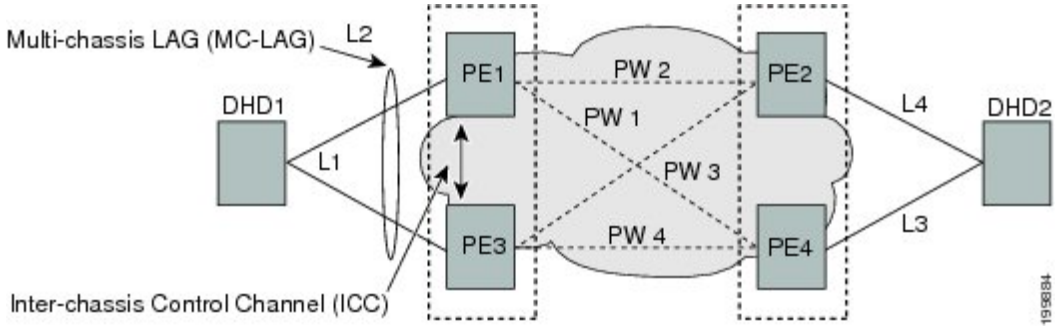


Figure 8: VPWS Two-way Pseudowire Redundancy

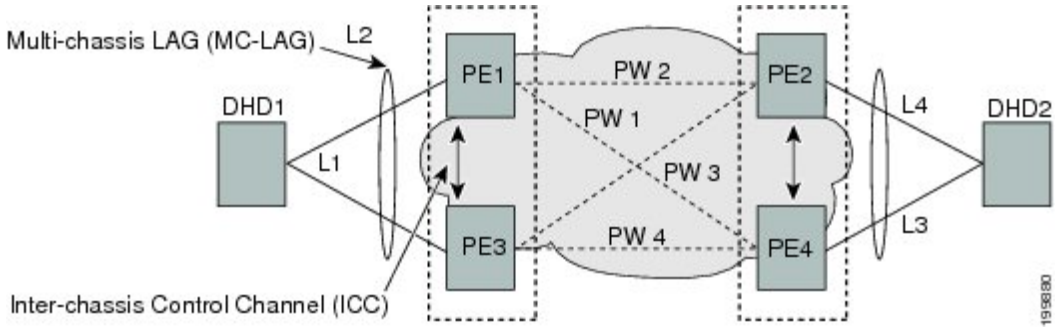


Figure 9: VPLS Pseudowires in One Redundancy Group

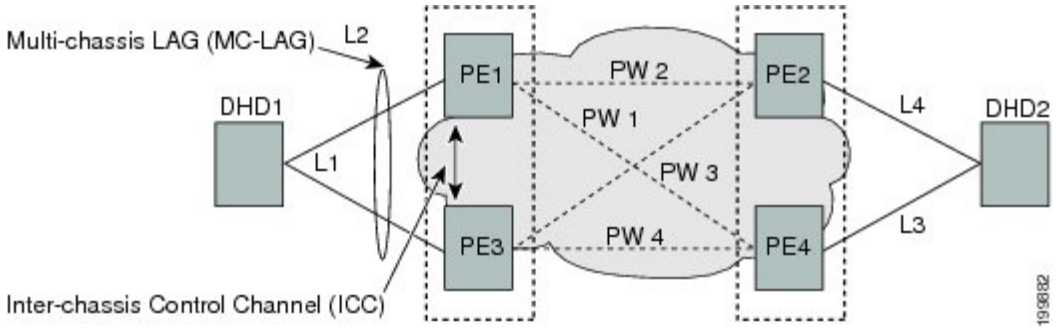
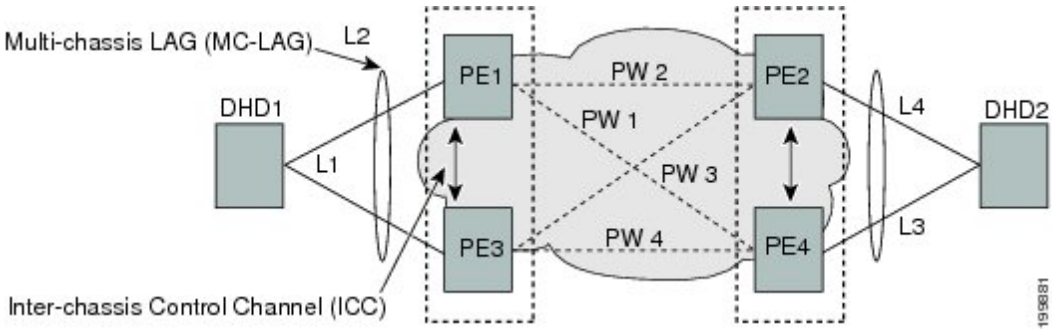


Figure 10: VPLS Pseudowires in Two Redundancy Groups



LACP Short Period Time Intervals

Load Balancing

Load balancing is a forwarding mechanism that distributes traffic over multiple links based on certain parameters. The Cisco ASR 9000 Series Router supports load balancing for all links in a bundle using Layer 2, Layer 3, and Layer 4 routing information.

This section describes load balancing support on link bundles.

For more information about other forms of load balancing on the Cisco ASR 9000 Series Router, see the following references:

- Per-flow load balancing on non-bundle interfaces using Layer 3 and 4 routing information— See the Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide.
- Pseudowire (PW) Load Balancing beginning in Cisco IOS XR 4.0.1—See the Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide.

Layer 2 Ingress Load Balancing on Link Bundles

By default, load balancing on Layer 2 link bundles is done based on the MAC source and destination address (SA/DA) fields in the incoming packet header. Table 1 shows a summary of the parameters used for load balancing of incoming traffic at Layer 2 based on whether the default mode, EFP-based, or flow-based load balancing is in use.

Per-flow load balancing is supported on all links in the bundle. This scheme achieves load sharing by allowing the router to distribute packets over one of the links in the bundle, that is determined through a hash calculation. The hash calculation is an algorithm for link selection based on certain parameters.

The standard hash calculation is a 5-tuple hashing, using the following parameters:

- IP source address
- IP destination address
- Router ID
- Layer 4 source port
- Layer 4 destination port

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

Note Load balancing for multicast traffic applies only when outgoing interfaces are link bundle interfaces or subinterfaces.

Table 1: Bundle Load Balancing for Incoming Traffic

Ingress Unicast, Flood, or Multicast Traffic	Parameters	Configuration
Default	• Source MAC address • Destination MAC address	n/a
EFP-based auto mode	XID of the xconnect	Auto mode is enabled using the **bundle load-balancing hash auto** command.
EFP-based with user hash	User hash	A user hash is configured using the **bundl load-balancing** *hash-value* command.
Flow-based with IP source and destination	• Source IP address • Destination IP address	Enabled using the L2VPN **load-balancing src-dst-ip** command.
Flow-based with MAC source and destination	• Source MAC address • Destination MAC address	Enabled using the L2VPN **load-balancing src-dst-mac** command.

Layer 3 Egress Load Balancing on Link Bundles

Layer 3 load balancing support began on the Cisco ASR 9000 Series Router in Cisco IOS XR 3.9.1, with changes introduced in Cisco IOS XR Release 4.0.1.

Layer 3 Load Balancing Before Cisco IOS XR Release 4.0.1

In Cisco IOS XR 3.9.1 through Cisco IOS XR 4.0, Layer 3 load balancing for link bundles is done on Ethernet Flow Points (EFPs) and is based on the IPv4 source and destination addresses in the packet. When Layer 3 service-specific load balancing is configured, all egressing bundles are load balanced based on the IPv4 source and destination addresses. When packets do not have IPv4 addresses, default load-balancing is used.

Layer 3 load balancing for link bundles is enabled globally, using the following command:

hw-module load-balance bundle l2-service l3-params

Layer 3 Load Balancing From Cisco IOS XR Release 4.0.1

Layer 3 load balancing for link bundles is done when outgoing interfaces are either bundles or bundle subinterfaces. 5-tuple hashing is used for load balancing among bundle member links, using the following parameters:

- IP source address
- IP destination address
- Router ID
- Layer 4 source port
- Layer 4 destination port

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.

Note In the case of in-transit IP-in-IP GRE traffic, prior to Release 5.3.1, third-generation line cards used only the outer IP header for hashing. Starting with Release 5.3.1, these line cards use both the inner and outer IP headers for hashing. In contrast, fourth and fifth-generation line cards consistently use only the outer IP header for hashing.

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.

Dynamic Load Balancing for LAG

Beginning in Cisco IOS XR Release 4.0.1, the Cisco ASR 9000 Series Router supports a method of dynamic load balancing among link aggregation (LAG) members. With dynamic load balancing, the hash algorithms for link selection include up to a maximum of 64 links, and are based on the current number of active members in the bundle.

QoS and Link Bundling

On the Cisco ASR 9000 Series Router, when QoS is applied 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 Cisco ASR 9000 Series Router, refer to the *Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide* and the *Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*.

VLANs on an Ethernet Link Bundle

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

- The maximum number of VLANs allowed per bundle is 4096.
- The maximum number of bundled VLANs allowed per router is 16384.

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**.

Link Bundle Configuration Overview

The following steps provide a general overview of the link bundle configuration process. 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 64 links to a single bundle.

Note A link is configured 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 failover between active and standby paired RSP 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 RSP fails, the standby RSP becomes operational. The configuration, node state, and checkpoint data of the failed RSP are replicated to the standby RSP. The bundled interfaces will all be present when the standby RSP becomes the active RSP.

Note Failover is always onto the standby RSP.

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

Link Failover

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

Multi-Gigabit Service Control Point

Multi-Gigabit Service Control Point (MGSCP) is a deployment model that uses certain link bundling and forwarding features on the Cisco ASR 9000 Series Aggregation Services Routers to support load balancing, clustering, and redundancy for broadband subscriber traffic on Cisco Service Control Engine (SCE) devices.

The Cisco SCE platform is used to provide many services such as user authorization, reporting, and application bandwidth metering for broadband subscribers. It manages IP traffic using a stateful processing mechanism based on application and subscriber awareness. Maintaining this statefulness requires that the SCE platform captures both the upstream and downstream flows of a session to classify it and provide Layer 7 processing at the application level.

To process an application that is implemented with a bundle of flows, such as FTP or Session Initiation Protocol (SIP), the SCE platform needs to process all the flows that comprise a session of this application. In addition, when the SCE platform is configured to implement per subscriber reporting or control (sometimes referred to as *subscriber awareness*), it must process all traffic flows that a given subscriber generates.

Because of this stateful processing to the subscriber level, the SCE platform is implemented in a network with a “bump-in-the-wire” topology for Layer 2 and Layer 3 transparency. However, as the number of broadband subscribers increases along with the bandwidth that an SCE platform must support, scaling the solution presents certain challenges when inserted into a typical network environment where asymmetric routing is often implemented and the two directions of a single session, or the many flows of a specific subscriber, could be split between different links.

The MGSCP solution on the Cisco ASR 9000 Series Router satisfies these requirements by providing a topology to scale multiple SCE devices in a cluster that are connected to the router using link bundling, where

all subscriber traffic can be directed through the same bundle member link. In addition, MGSCP also provides the benefits of load balancing and redundancy.

This figure shows a basic network topology for MGSCP with a Cisco ASR 9000 Series Router connected between the subscriber and core networks, and acting as a dispatcher for the attached SCE cluster. The N+1 notation indicates one backup (or protect) link for the other active links on either side of the SCEs.

Figure 11: Basic MGSCP Network Topology

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.

YANG Data Model You can programmatically perform the configuration using `openconfig-lacp.yang` OpenConfig 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 ASR 9000 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. **lacp fast-switchover**
8. **exit**
9. **interface** {**GigabitEthernet** | **TenGigE**} interface-path-id
10. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
11. **bundle port-priority** *priority*
12. **no shutdown**
13. **exit**
14. **bundle id** *bundle-id* `[` **mode** `{` **active** `|` **passive** `|` **on** `}]` **no shutdown exit**
15. **end** or **commit**
16. **exit**
17. **exit**

18. Perform Step 1 through Step 15 on the remote end of the connection.
19. **show bundle Bundle-Ether** *bundle-id*
20. **show lacp bundle Bundle-Ether** *bundle-id*

DETAILED STEPS

Procedure

	Command or Action	Purpose
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 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/RSP0/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 Cisco ASR 9000 Series Router, only a Layer 3 bundle interface requires an IP address.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RSP0/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/RSP0/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.

	Command or Action	Purpose
		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/RSP0/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/RSP0/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/RSP0/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/RSP0/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/RSP0/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/RSP0/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.

	Command or Action	Purpose
Step 13	**exit** **Example:** `RP/0/RSP0/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/RSP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/1`
	`RP/0/RSP0/CPU0:router(config-if)# bundle id 3`
	`RP/0/RSP0/CPU0:router(config-if)# bundle port-priority 2`
	`RP/0/RSP0/CPU0:router(config-if)# no shutdown`
	`RP/0/RSP0/CPU0:router(config-if)# exit`
	`RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/3`
	`RP/0/RSP0/CPU0:router(config-if)# bundle id 3`
	`RP/0/RSP0/CPU0:router(config-if)# no shutdown`
	`RP/0/RSP0/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/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.

	Command or Action	Purpose
Step 16	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 17	**exit** **Example:** `RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router# show lacp bundle Bundle-Ether 3`	(Optional) Shows detailed information about LACP ports and their peers.

Configuring EFP Load Balancing on an Ethernet Link Bundle

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

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

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

SUMMARY STEPS

1. **configure**
2. **hw-module load-balance bundle l2-service l3-params**
3. **interface Bundle-Ether** *bundle-id* **l2transport**
4. **bundle load-balance hash** *hash-value* [**auto**]
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module load-balance bundle l2-service l3-params** **Example:** `RP/0/RSP0/CPU0:router(config)# hw-module load-balance bundle l2-service l3-params`	(Optional) Enables Layer 3 load balancing on Layer 2 link bundles.
Step 3	**interface Bundle-Ether** *bundle-id* **l2transport** **Example:** `RP/0/RSP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport`	Creates a new Ethernet link bundle with the specified *bundle-id* and with Layer 2 transport enabled. The range is 1 to 65535.

Step 4	**bundle load-balance hash** *hash-value* [**auto**]
	Example:
	`RP/0/RSP0# bundle load-balancing hash 1`
	or
	`RP/0/RSP0# bundle load-balancing hash auto`
		Configures all egressing traffic on the fixed members of a bundle to flow through the same physical member link.
		• *hash-value*—Numeric value that specifies the physical member link through which all egressing traffic in this bundle will flow. The values are 1 through 8.
		• **auto**—The physical member link through which all egressing traffic on this bundle will flow is automatically chosen.

Step 5	**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.

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

Configuring VLAN Bundles

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

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

Procedure

	Command or Action	Purpose
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**
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** {**GigabitEthernet** | **TenGigE**}*interface-path-id*
19. **lacp fast-switchover**

DETAILED STEPS

Procedure

	Command or Action	Purpose
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. 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/RSP0/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/RSP0/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/RSP0/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:**	(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

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby`	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/RSP0/CPU0:router(config-if)# exit`	Exits the interface configuration submode.
Step 8	**interface Bundle-Ether** *bundle-id.vlan-id* **Example:** `RP/0/RSP0/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** **Example:** `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged`	Sets the Layer 2 encapsulation of an interface. **Note** The **dot1q vlan** command is replaced by the **encapsulation dot1q** command on the Cisco ASR 9000 Series Router. It is still available for backward-compatibility, but only for Layer 3 interfaces.
Step 10	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router(config-subif)# exit`	Exits subinterface configuration mode for the VLAN subinterface.

	Command or Action	Purpose
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/RSP0/CPU0:router(config-subif)# end`
	or
	`RP/0/RSP0/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/RSP0/CPU0:router(config-subif)# end`	Exits interface configuration mode.
Step 16	**exit** **Example:** `RP/0/RSP0/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** {**GigabitEthernet** | **TenGigE**}*interface-path-id* **Example:** `RP/0/RSP0/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.

	Command or Action	Purpose
Step 19	**lacp fast-switchover** **Example:** `RP/0/RSP0/CPU0:router(config-if)# lacp fast-switchover`	(Optional) If you enabled 1:1 link protection (you set the value of the **bundle maximum-active links** command to 1) on a bundle with member links running LACP, you can optionally disable the wait-while timer in the LACP state machine. Disabling this timer causes a bundle member link in standby mode to expedite its normal state negotiations, thereby enabling a faster switchover from a failed active link to the standby link.

Configuring POS Link Bundles

This section describes how to configure a POS link bundle.

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

SUMMARY STEPS

1. **configure**
2. **interface Bundle-POS** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **exit**
8. **interface POS** *interface-path-id*
9. **bundle id** *bundle-id*
10. **bundle port-priority** *priority*
11. **no shutdown**
12. **exit**
13. Repeat Step 19 through Step 21 to add more links to a bundle
14. **end** or **commit**
15. **exit**
16. **exit**
17. Perform Step 1 through Step 23 on the remote end of the connection.
18. **show bundle Bundle-POS** *number*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-POS** *bundle-id* **Example:** `RP/0/RSP0/CPU0:router#(config)#interface Bundle-POS 2`	Configures and names the new bundled POS interface. Enters the interface configuration submode, from where interface specific configuration commands are executed. Use the **exit** command to exit from the interface configuration submode, and get back to the normal global configuration mode.
Step 3	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0`	Assigns an IP address and subnet mask to the virtual interface using the ip address configuration subcommand.
Step 4	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle minimum-active bandwidth 620000`	(Optional) Sets the minimum amount of bandwidth required before a user can bring up a bundle.
Step 5	**bundle minimum-active links** *links* **Example:** `RP/0/RSP0/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/RSP0/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 according to 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**	Exits the interface configuration submode.

	Command or Action	Purpose
Step 8	**interface POS** *interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface POS 0/1/0/0`	Enters POS interface configuration mode and specifies the POS interface name and interface-path-id notation *rack/slot/module/port*.
Step 9	**bundle id** *bundle-id* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle-id 3`	Adds the link to the specified bundle. 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 10	**bundle port-priority** *priority* **Example:** `RP/0/RSP0/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 11	**no shutdown** **Example:** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	Removes the shutdown configuration which forces the interface administratively down. The **no shutdown** command then returns the link to an up or down state, depending on the configuration and state of the link.
Step 12	**exit** **Example:** `RP/0/RSP0/CPU0:router# exit`	Exits the interface configuration submode for the POS interface.
Step 13	Repeat Step 19 through Step 21 to add more links to a bundle	(Optional) Adds more links to the bundle you created in Step 2.

Step 14	**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.

	Command or Action	Purpose
		• 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/RSP0/CPU0:router(config-if)# exit`	Exits interface configuration mode.
Step 16	**exit** **Example:** `RP/0/RSP0/CPU0:router(config)# exit`	Exits global configuration mode.
Step 17	Perform Step 1 through Step 23 on the remote end of the connection.	Brings up the other end of the link bundle.
Step 18	**show bundle Bundle-POS** *number* **Example:** `RP/0/RSP0/CPU0:router# show bundle Bundle-POS 1`	(Optional) Shows information about the specified POS link bundle.

Configuring Multichassis Link Aggregation

Perform these tasks to configure Multichassis Link Aggregation (MC-LAG):

Configuring Interchassis Communication Protocol

Perform this task to configure Interchassis Communication Protocol (ICCP).

SUMMARY STEPS

1. **configure**
2. **redundancy iccp group** *group-id*
3. **member neighbor** *neighbor-ip-address*
4. **backbone interface** *interface-type-id*
5. **isolation recovery-delay** *delay*
6. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
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	**member neighbor** *neighbor-ip-address* **Example:** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# member neighbor 10.1.1.1`	Configures ICCP members. This is the ICCP peer for this redundancy group. Only one neighbor can be configured per redundancy group. The IP address is the LDP router-ID of the neighbor. This configuration is required for ICCP to function.
Step 4	**backbone interface** *interface-type-id* **Example:** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# backbone interface GigabitEthernet0/1/0/2`	Configures ICCP backbone interfaces. This is an optional configuration to detect isolation from the network core, and triggers switchover to the peer POA if the POA on which the failure is occurring is active. Multiple backbone interfaces can be configured for each redundancy group. When all backbone interfaces are not UP, this is an indication of core isolation. When one or more backbone interfaces are UP, then the POA is not isolated from the network core. Backbone interfaces are typically the interfaces which L2VPN pseudowires can use.
Step 5	**isolation recovery-delay** *delay* **Example:** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# isolation recovery-delay 30`	Configures the isolation parameters and specifies delay before clearing isolation condition after recovery from failure. Isolation recovery delay timer is started once the core isolation condition has cleared. When the timer expires, the POA can take over as the active POA (depending on other conditions like bundle recovery delay timer). This allows: • the network core to reconverge after the backbone interfaces have come up • ICCP state to be exchanged in order for POAs to know what state they are supposed to be in so that MCLAG bundles do not flap excessively. This is an optional configuration; if not configured, the delay is set to 180 seconds, by default.

	Command or Action	Purpose

Step 6	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-redundancy-iccp-group)# end`
	or
	`RP/0/RSP0/CPU0:router(config-redundancy-iccp-group)# 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 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

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**redundancy iccp group** *group-id* **Example:**	Adds an ICCP redundancy group.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)# redundancy iccp group 100`	
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.

	Command or Action	Purpose
		• 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

Procedure

	Command or Action	Purpose
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:**	Sets the wait-while timeout for members of this bundle.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router#(config-if)# bundle wait-while 100`	
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 Dual-Homed Device

Perform this task to configure the dual-homed device (DHD).

Note If an ASR 9000 Series Router is to be used as a DHD, it is recommended that you configure the **bundle maximum-active links** *links* command where *links* is the number of links connecting the DHD to one of the POAs.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bundle wait-while** *milliseconds*
4. **lacp switchover suppress-flaps** *milliseconds*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
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-if)# interface Bundle-Ether 3`	Creates and names a new Ethernet link bundle.
Step 3	**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 4	**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.
Step 5	**end** or **commit** **Example:**	Saves configuration changes.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-if)# end`
	or
	`RP/0/RSP0/CPU0:router(config-if)# commit`
		• 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.
		
		The members added to the bundle on one POA go *Active*, and the members on the other POA are in *Standby* state. This can be verified by using the **show bundle** command on either POA to display the membership information for correctly configured members on both the POAs:
		
		`RP/0/RSP0/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`
		` ---------- ---------- ---------- ---------- ----------`
		` Gi0/0/0/0 Local Active 0x8001, 0x9001 1000000`
		` Link is Active`

	Command or Action	Purpose
		Gi0/0/0/0 5.4.3.2 Standby 0x8002, 0xa001 1000000 Link is marked as Standby by mLACP peer

Note To switch to an active POA, use the **mlacp switchover Bundle-Ether** command on the currently active router.

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

Procedure

	Command or Action	Purpose
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:**	Configures the pseudowire encapsulation to MPLS.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# encapsulation 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. • The **redundancy one-way** command is mandatory for one-way pseudowire redundancy in MC-LAG to avoid inconsistent PW states on remote PEs for the MC-LAG point of attachments.

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

Procedure

	Command or Action	Purpose
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:**	Specifies the interface type ID.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1`	
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/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# end`
	or
	`RP/0/RSP0/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

Procedure

	Command or Action	Purpose
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.

	Command or Action	Purpose
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

	Command or Action	Purpose
		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

Procedure

	Command or Action	Purpose
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.

	Command or Action	Purpose

Step 4	**bridge group** *bridge-group-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group csco`
	`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-ac)# 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/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)? [cancel]:`
		- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		- 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.

How to Configure MGSCP

Perform these tasks to configure MGSCP.

Prerequisites for Configuring MGSCP

Before configuring MGSCP, be sure that the following prerequisites are met:

- You have Gigabit Ethernet or 10-Gigabit Ethernet line cards installed in the Cisco ASR 9000 Series Router.
- You understand how to configure your cluster of Service Control Engine (SCE) devices and configure them according to the desired requirements of your network, including the following requirements for MGSCP support:
  - When you connect the SCE devices to the Cisco ASR 9000 Series Router, be sure that each SCE device has two separate physical links connecting to two different bundle interfaces on the Cisco ASR 9000 Series Router as follows:

    — One link from each SCE device is connected to a link on the bundle interface that is routed to the access (or subscriber) side of the network.

    — The second link from each SCE device is connected to a link on another bundle interface that is routed to the core side of the network.
  - On the SCE device, you configure the SCE ports for link failure reflection (using the **link failure-reflection** command) to ensure that if a link on one side of the SCE goes down, then the link on the other side is automatically shut down. For more information, see the "Configuring the Connection" chapter in the Cisco SCE software configuration guide for your device and release at:

    http://www.cisco.com/en/US/products/ps6134/products_installation_and_configuration_guides_list.html
- For your bundle configuration on the Cisco ASR 9000 Series Router, determine the following information:
  - The maximum number of active links that you will support.
  - The bundle links that will be protect (backup) links. You can configure a maximum of 4 protect links.

- To maintain the statefulness of the connected SCEs, all subscriber flows must pass through the same SCE. Therefore, before you configure MGSCP, you need to determine how you want to configure the router to redirect subscriber traffic to ensure that it passes through the appropriate bundle interfaces connected to that SCE.

  You can use one of the following methods:

  - ACL-Based Forwarding (ABF)—Supports only IP addresses for the next hop, and can be complex to configure. For more information about ABF, see the "Implementing Access Lists and Prefix Lists" chapter of the Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide.
  - Virtual Routing and Forwarding (VRF)—Recommended. Uses VRF instances for the access and network bundles, which can then be routed using static or dynamic routing with OSPF and BGP.

Restrictions for Configuring MGSCP

Before configuring MGSCP, consider these restrictions:

- You can configure up to a maximum of 4 protect links on a bundle.
- You can configure up to a maximum of 8 member links on a bundle.
- Packets received on the ingress interface must not be tagged with MPLS for MGSCP to effectively do load balancing at the egress interface.

Configuring the Access Bundle for the Subscriber-Facing Side

The configuration of the access bundle facing the subscriber side of the network is similar to the core bundle configuration, with the following guidelines:

- If using VRFs to route subscriber traffic on the same SCE to the bundle (recommended), then a separate VRF is used for the subscriber-facing side.
- Link-order signaling is required to enable LACP processing of link ordering numbers (LONs) for load balancing tables.
- Bundle load balancing is configured based on source IP address.
- The maximum number of active links must be configured to match the maximum number of active links on the core bundle.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *ipv4-address mask*
5. **lacp cisco enable link-order signaled**
6. **bundle load-balancing hash src-ip**
7. **bundle maximum-active links** *links* [**hot-standby**]
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
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 100`	Specifies or creates an Ethernet bundle interface for the subscriber-facing side of the network, where *bundle-id* is a number from 1 to 65535, and enters interface configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# vrf access`	(Optional—Recommended) Specifies the VRF instance for the subscriber-facing side of the network in which this Ethernet bundle participates.
Step 4	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0`	Specifies an IPv4 address and mask that is part of the specified VRF for this interface, where *ipv4-address* is the 32-bit IP address wtih corresponding mask in dotted-decimal format (A.B.C.D). **Note** • This command must be specified after the **vrf** command to be sure that the IP address is part of the VRF instance.
Step 5	**lacp cisco enable link-order signaled** **Example:** `RP/0/RSP0/CPU0:router(config-if)# lacp cisco enable link-order signaled`	Enables the use of Cisco TLVs to include link order numbering as part of the LACP processing on this bundle.
Step 6	**bundle load-balancing hash src-ip** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle load-balancing hash src-ip`	Specifies that the hash used for load balancing on the subscriber bundle interface is based on source IP address.
Step 7	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 2`	Specifies the maximum number of active links allowed for the bundle, and sets the upper bound on the link ordering numbers in use for load balancing tables. **Note** To support MGSCP, this command must also be configured with the same value on the core bundle.

	Command or Action	Purpose

Step 8	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router (config-bfd-if)# end`
	or
	`RP/0/RSP0/CPU0:router(config-bfd-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 the Network Bundle for the Core-Facing Side

The configuration of the bundle facing the core side of the network is similar to the access bundle configuration, with the following guidelines:

- If using VRFs to route subscriber traffic on the same SCE to the bundle (recommended), then a separate VRF is used for the core-facing side.
- Link-order signaling is required to enable LACP processing of LONs for load balancing tables.
- Bundle load balancing is configured based on destination IP address.
- The maximum number of active links must be configured to match the maximum number of active links on the access bundle.

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *ipv4-address mask*
5. **lacp cisco enable link-order signaled**
6. **bundle load-balancing hash dst-ip**
7. **bundle maximum-active links** *links* [**hot-standby**]
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
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 100`	Specifies or creates an Ethernet bundle interface for the subscriber-facing side of the network, where *bundle-id* is a number from 1 to 65535, and enters interface configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-if)# vrf access`	(Optional—Recommended) Specifies the VRF instance for the core-facing side of the network in which this Ethernet bundle participates.
Step 4	**ipv4 address** *ipv4-address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0`	Specifies an IPv4 address and mask that is part of the specified VRF for this interface, where *ipv4-address* is the 32-bit IP address wtih corresponding mask in dotted-decimal format (A.B.C.D). **Note** • This command must be specified after the **vrf** command to be sure that the IP address is part of the VRF instance.
Step 5	**lacp cisco enable link-order signaled** **Example:** `RP/0/RSP0/CPU0:router(config-if)# lacp cisco enable link-order signaled`	Enables the use of Cisco TLVs to include link order numbering as part of the LACP processing on this bundle.
Step 6	**bundle load-balancing hash dst-ip** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle load-balancing hash dst-ip`	Specifies that the hash used for load balancing on the subscriber bundle interface is based on destination IP address.
Step 7	**bundle maximum-active links** *links* [**hot-standby**] **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 2`	Specifies the maximum number of active links allowed for the bundle, and sets the upper bound on the link ordering numbers in use for load balancing tables. **Note** • To support MGSCP, this command must also be configured with the same value on the access bundle.

	Command or Action	Purpose

Step 8	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router (config-bfd-if)# end`
	or
	`RP/0/RSP0/CPU0:router(config-bfd-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 the Bundle Member Interfaces

When the access and core bundles have been configured, bundle interfaces must be configured as the active and protect links on those bundles, with the following guidelines:

- A link becomes a bundle member using the **bundle id** command and specifying the ID of the corresponding bundle interface. For MGSCP, there are two different bundles: one for the access side traffic, and one for the core side traffic. These bundles each have a link connecting to either side of an SCE. Be sure to carefully map your interfaces to the appropriate bundle.
- LACP is required for MGSCP, so the link must be configured with **mode active** on the bundle.
- Active and backup (protect) links are configured using the **bundle port-priority** command:
  - To configure a working (active) link, use a priority of 1. The maximum number of active links that you can configure is determined by the value of the **bundle maximum-active links** command on the bundle.
  - Any priority other than 1 designates the link as a protect link. You can configure a maximum of 4 protect links.

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **bundle id** *bundle-id* **mode active**

4. **bundle port-priority** *priority*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0`	Specifies or creates a Gigabit Ethernet or 10-Gigabit Ethernet interface, where *interface-path-id* is the physical location of the interface using *rack*/*slot*/*module*/*port* notation, and enters interface configuration mode.
Step 3	**bundle id** *bundle-id* **mode active** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle id 100 mode active`	Adds the interface as a member of the specified bundle, and runs LACP in active mode on the interface to exchange LACP packets for MGSCP.
Step 4	**bundle port-priority** *priority* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bundle port-priority 1`	Specifies the LACP priority for the interface and determines if a bundle interface is an active or protect link for MGSCP: • Value of 1—Specifies the link is an active interface. • Value other than 1—Specfies the link is a protect interface. The default is 32768.

Step 5	**end** or **commit**
	Example:
	`RP/0/RSP0/CPU0:router (config-bfd-if)# end`
	or
	`RP/0/RSP0/CPU0:router(config-bfd-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.

	Command or Action	Purpose
		- 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 VRFs to Route Traffic to the Bundles

VRFs are the recommended way to route subscriber traffic to the bundles to be sure that all subscriber traffic remains with the same SCE device for statefulness. To configure VRFs for MGSCP, complete one of the following tasks:

Configuring VRFs with Static Routing

These steps summarize the tasks required to configure VRFs using static routing:

1. Configure two VRFs in global configuration—one each for the access and core sides of the network. Be sure to specify the IPv4 unicast address family.
2. Configure IPv4 addresses at each of the bundle interfaces and associate those addresses with the corresponding VRF that you configured in global configuration for the access and core side of the network.
3. Configure IPv4 addresses at the Gigabit Ethernet physical interfaces and associate those addresses with the corresponding VRF that you configured in global configuration for the access and core side of the network.
4. Configure static routing using the **router static** command to map the access and core VRFs to their corresponding bundle interfaces.

For a sample configuration, see the “Example: Configuring VRFs with Static Routing” section.

Configuring VRFs with Dynamic Routing

VRFs for MGSCP are supported for both OSPF and BGP routing protocols. The general configuration of the VRFs in global configuration and at the bundle and physical interfaces is the same as for static routing.

These steps summarize the tasks required to configure VRFs using OSPF routing:

1. Configure two VRFs in global configuration—one each for the access and core sides of the network. Be sure to specify the IPv4 unicast address family.
2. Configure IPv4 addresses at each of the bundle interfaces and associate those addresses with the corresponding VRF that you configured in global configuration for the access and core side of the network.
3. Configure IPv4 addresses at the Gigabit Ethernet physical interfaces and associate those addresses with the corresponding VRF that you configured in global configuration for the access and core side of the network.
4. Configure a dynamic routing protocol, such as OSPF, using the **router ospf** command to define the VRFs and associate the bundle and physical interfaces to the OSPF areas.

For a sample configuration, see the "Example: Configuring VRFs with OSPF Routing" section on page 310.

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 running LACP:

```
RP/0/RSP0/CPU0:Router(config)# config

RP/0/RSP0/CPU0:Router(config-if)# interface Bundle-Ether 3
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config-if)# interface TenGigE 0/3/0/0
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router(config-if)# interface TenGigE 0/3/0/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

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

```
RP/0/RSP0/CPU0:Router(config)# config

RP/0/RSP0/CPU0:Router(config-if)# interface bundle-ether 50
RP/0/RSP0/CPU0:Router(config-if)# root
RP/0/RSP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/11
RP/0/RSP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/16
RP/0/RSP0/CPU0:Router(config-if)# bundle id 50 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# interface TenGigE 0/0/0/27
RP/0/RSP0/CPU0:Router(config-if)# bundleid 50 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# interface HundredGigE 0/6/0/1
RP/0/RSP0/CPU0:Router(config-if)# bundleid 50 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# root
RP/0/RSP0/CPU0:Router(config)# commit
RP/0/RSP0/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/RSP0/CPU0:Router(config-subif)# config
RP/0/RSP0/CPU0:Router(config-subif)# interface Bundle-Ether 1
RP/0/RSP0/CPU0:Router(config-ifsubif)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-ifsubif)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links
RP/0/RSP0/CPU0:Router(config-ifsubif)# exit
RP/0/RP0/CPU0:Router(config-subif)# ip addr 20.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# interface Bundle-Ether 1.1
RP/0/RSP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RSP0/CPU0:Router(config-subif) # ip addr 10.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subifif)# exit
RP/0/RSP0/CPU0:Router(config-if)# interface Bundle-Ether 1.2
RP/0/RSP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RSP0/CPU0:Router(config-subif)Router # ip addr20.2.3.4/24

RP/0/RSP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subifif)# exit
RP/0/RSP0/CPU0:Router(config)# interface gig 0/1/5/7
RP/0/RSP0/CPU0:Router(config-if)# bundle-id 1 mode act
RP/0/RSP0/CPU0:Router(config-if)# commit
RP/0/RSP0/CPU0:Router(config-if)# exit
```

Example: Configuring a POS Link Bundle

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

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-POS 5
RP/0/RSP0/CPU0:Router(config-if)#ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface POS 0/0/1/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 5
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

Example: Configuring EFP Load Balancing on an Ethernet Link Bundle

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

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

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

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

Example: Configuring Multichassis Link Aggregation

This example shows how to configure POAs:

Active POA

```
interface Bundle-Ether10
  mlacp iccp-group 1
  mlacp port-priority 10
```

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 GigabitEthernet0/0/0/0
   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
 -------------------- -------------- ----------- -------------- ----------
 Gi0/0/0/0       Local      Active    0x8001, 0x9001   1000000
   Link is Active
 Gi0/0/0/0       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 : GigabitEthernet0/0/0/0
 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
 -------------------- -------------- ----------- -------------- ----------
```

```
 Gi0/0/0/0       Local      Standby   0x8003, 0x9001   1000000
   mLACP peer is active
 Gi0/0/0/0       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 bd1
  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:

```
l2vpn pw-class class_mpls
  encapsulation mpls
    redundancy one-way
  !
!
```

This example illustrates an overall MC-LAG configuration:

Topology:

```
DHD           POA 1           POA 2
Gi0/0/0/0 --------------- Gi0/0/0/0
Gi0/0/0/1 --------------- Gi0/0/0/1
Gi0/0/0/2
Gi0/0/0/3 ----------------------------------------- Gi0/0/0/0
Gi0/0/0/4 ----------------------------------------- Gi0/0/0/1
             Gi0/0/0/2         Gi0/0/0/2
             Gi0/0/0/3 --------------- Gi0/0/0/3
             Gi0/0/0/4 --------------- Gi0/0/0/4
```

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 GigabitEthernet0/0/0/0
 description Connected to DHD Gi0/0/0/0
 bundle id 1 mode active
 lacp period short
 no shutdown
!
interface GigabitEthernet0/0/0/3
 description Connected to POA2 Gi0/0/0/3
 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 GigabitEthernet0/0/0/3
 !
!
```

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 GigabitEthernet0/0/0/0
 description Connected to DHD Gi0/0/0/3
 bundle id 1 mode active
 lacp period short
 no shutdown
!
```

```
interface GigabitEthernet0/0/0/3
 description Connected to POA1 Gi0/0/0/3
 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 GigabitEthernet0/0/0/3
 !
!
```

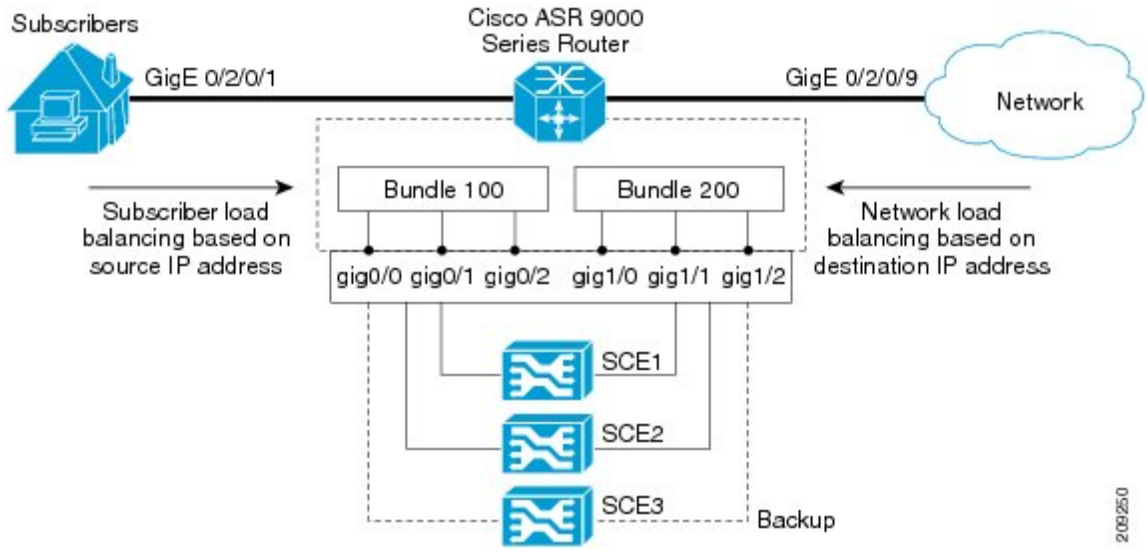
On the DHD:

```
interface Bundle-Ether1
 lacp switchover suppress-flaps 300
 bundle wait-while 100
!
interface GigabitEthernet0/0/0/0
 description Connected to POA1 Gi0/0/0/0
 bundle id 1 mode active
 lacp period short
 no shutdown
!
interface GigabitEthernet0/0/0/3
 description Connected to POA2 Gi0/0/0/0
 bundle id 1 mode active
 lacp period short
 no shutdown
!
```

Configuration Examples for MGSCP

This figure illustrates a sample network with a single Cisco ASR 9000 Series Router as a dispatcher for a cluster of SCE devices that is used as an example for the sample configurations.

Figure 12: Cisco ASR 9000 Series Router as Dispatcher for SCE Cluster



This section includes the following examples:

Example: Configuring Bundle Interfaces and Member Links

This example shows how to configure the two bundles on the Cisco ASR 9000 Series Router shown in the figure above. Each bundle supports a maximum of two active links (configurations for both bundles must match), with one backup protect link.

The bundle interface members in Ethernet bundle 100 connect the SCE device links for the subscriber side of the network using load balancing based on source IP address. The bundle interface members in Ethernet bundle 200 connect the SCE device links for the core side of the network using load balancing based on destination IP address.

Subscriber-Facing Access Bundle Configuration

```
interface Bundle-Ether 100
 description Faces-SCE-Subscriber-Side
 vrf access
 ipv4 address 10.0.0.1 255.255.255.0 <<-- Same subnet as Bundle-Ether 200
 lacp cisco enable link-order signaled <<-- Enables Cisco LACP extensions, required for
MGSCP
 bundle load-balancing hash src-ip <<-- Hashes traffic based on source (Subscriber) IP
 bundle maximum-active links 2
!
interface GigabitEthernet 0/0/0/0
 description to SCE1
 bundle id 100 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/0/1
 description to SCE2
 bundle id 100 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/0/3
```

```
 description to SCE3 (backup)
 bundle id 100 mode active
```

Core-Facing Bundle Configuration

```
interface Bundle-Ether 200
 description Faces-SCE-Network-Side
 vrf core
 ipv4 address 10.0.0.2 255.255.255.0 <<-- Same subnet as Bundle-Ether100
 lacp cisco enable link-order signaled <<-- Enables Cisco LACP extensions, required for
MGSCP
 bundle load-balancing hash dst-ip <<-- Hashes traffic based on destination (Subscriber)
IP
 bundle maximum active links 2
!
interface GigabitEthernet 0/0/1/0
 description from SCE1
 bundle id 200 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/1/1
 description from SCE2
 bundle id 200 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/1/2
 description from SCE3 (standby)
 bundle id 200 mode active
```

Examples: Configuring VRFs to Route Traffic to the Bundles

To ensure that the traffic to and from the same subscriber is going through the same port of the SCE, VRFs are recommended. You need to configure two VRFs for MGSCP: One for the access traffic, and one for the core traffic.

The examples in this section also show two different ways that you can route using VRFs with either static or dynamic (OSPF) routing for the bundle interface at the VRF:

Example: Configuring VRFs with Static Routing

In the following configuration examples, VRFs are established for the core and access sides of the network using IPv4. From there, the bundle interface addresses for each side are each configured as part of the VRF, as well as two physical interfaces. The final piece of the configuration shows how to configure a static route to each VRF using the bundle interfaces.

VRF Global Configuration

```
vrf core
 address-family ipv4 unicast
 import route-target
  1:1
 !
 export route-target
  1:1
 !
vrf access
 address-family ipv4 unicast
```

```
 import route-target
  1:1
 !
 export route-target
  1:1
 !
```

VRF Configuration on Bundle Interfaces

```
interface Bundle-Ether100
 vrf access
 ipv4 address 10.0.0.1 255.255.255.0
!
interface Bundle-Ether200
 vrf core
 ipv4 address 10.0.0.2 255.255.255.0
```

VRF Configuration on Physical Interfaces

```
interface GigabitEthernet0/2/0/1
 desc Subscriber-facing
 vrf access
 ipv4 address 10.10.1.2 255.255.255.0

interface GigabitEthernet0/2/0/9
 desc Network-facing
 vrf core
 ipv4 address 10.20.1.2 255.255.255.0
 negotiation auto
```

Static Routing Configuration for the VRFs to the Bundle Interfaces

```
router static
 vrf core
 address-family ipv4 unicast
  0.0.0.0/0 10.0.0.1
 !
 !
 vrf access
 address-family ipv4 unicast
  0.0.0.0/0 10.0.0.2
 !
 !
```

Example: Configuring VRFs with OSPF Routing

In the following configuration examples, VRFs are established for the core and access sides of the network using IPv4. From there, you configure an OSPF routing instance and area to include the VRFs and associate the bundle and physical interfaces.

Global VRF Configuration

```
vrf core
 address-family ipv4 unicast
 import route-target
  1:1
 export route-target
```

```
  1:1

vrf access
 address-family ipv4 unicast
 import route-target
  1:1
 export route-target
  1:1
```

VRF Configuration on Physical Interfaces

```
interface GigabitEthernet0/2/0/1
 vrf access
 ipv4 address 10.10.1.4 255.255.255.0

_
interface GigabitEthernet0/2/0/9
 vrf core
 ipv4 address 10.20.1.4 255.255.255.0
```

OSPF Routing Configuration for the VRFs and the Bundle and Physical Interfaces

```
router ospf 100
 vrf core
 router-id 10.20.1.2
 area 0
  interface Bundle-Ether200
  interface GigabitEthernet0/2/0/9

 vrf access
 router-id 10.10.1.2
 area 0
  interface Bundle-Ether100
  interface GigabitEthernet0/2/0/1
```

Example: Configuring MGSCP with ABF to Route Traffic to the Bundles

The following example routes traffic to the bundles using access lists to forward the traffic.

```
ipv4 access-list inbound
!
! Set the nexthop address to be a virtual IP address on the same network
! as the access bundle.
!
 10 permit ipv4 any any nexthop 10.10.1.5
!
ipv4 access-list outbound
!
! Set the nexthop address to be a virtual IP address on the same network
! as the core bundle.
!
10 permit ipv4 any any nexthop 10.20.1.5
!
! Configure static ARP for the virtual IP addresses
!
arp vrf default 10.10.1.5 0024.98eb.bf8a ARPA
arp vrf default 10.20.1.5 0024.98eb.bf8b ARPA

interface Bundle-Ether100
 ipv4 address 10.10.1.2 255.255.255.0
```

```
!
interface Bundle-Ether200
 ipv4 address 10.20.1.2 255.255.255.0
!
interface GigabitEthernet0/2/0/1
 ipv4 address 10.10.1.3 255.255.255.0
 ipv4 access-group inbound
!
interface GigabitEthernet0/2/0/9
ipv4 address 10.20.1.3 255.255.255.0
 ipv4 access-group outbound
!
```