# Configuring PPP

This module describes the configuration of Point-to-Point Protocol (PPP) on POS and serial interfaces on the Cisco ASR 9000 Series Router.

### Feature History for Configuring PPP Interfaces

| Release | Modification |
|---|---|
| Release 3.9.0 | PPP and ICSSO for PPP and MLPPP were introduced on the Cisco ASR 9000 Series Router. |
| Release 3.9.1 | Support for T3 Channelized SONET was added. |
| Release 4.0.0 | Support for the following features was added for the 2-Port Channelized OC-12c/DS0 SPA:<br>• IPHC over PPP, MLPPP, and MLPPP/LFI<br>• NxDS0 serial interfaces<br>Support for PPP was introduced on the following SPAs:<br>• 1-Port Channelized OC-48/STM-16 SPA<br>• 1-Port OC-192c/STM-64 POS/RPR XFP SPA<br>• 2-Port OC-48c/STM-16 POS/RPR SPA<br>• 8-Port OC-12c/STM-4 POS SPA |
| Release 4.0.1 | Support for PPP was added for the following SPAs on the Cisco ASR 9000 Series Router:<br>• Cisco 1-Port Channelized OC-3/STM-1 SPA (also supports MLPPP)<br>• Cisco 2-Port and 4-Port Clear Channel T3/E3 SPA<br>• Cisco 4-Port OC-3c/STM-1 SPA<br>• Cisco 8-Port OC-3c/STM-1 SPA |

| | |
|---|---|
| Release 4.1.0 | Support for the Noise Attribute was added for PPP to remove links on MLPPP bundles when Link Noise Monitoring (LNM) thresholds are crossed on a link.<br>Support for PPP, including MLPPP support on T1/E1 channels, was introduced on the following SPAs:<br>• Cisco 4-Port Channelized T3 SPA<br>• Cisco 8-Port Channelized T1/E1 SPA |



# Prerequisites for Configuring PPP

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

Before you can configure PPP authentication on a POS or serial interface, be sure that the following tasks and conditions are met:

- Your hardware must support POS or serial interfaces.
- You have enabled PPP encapsulation on your interface with the **encap ppp** command, as described in the appropriate module:
  - To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces module in this manual.
  - To enable PPP encapsulation on a serial interface, see the Configuring Serial Interfaces module in this manual.

# Information About PPP

To configure PPP and related features, you should understand the information in this section:

## PPP Authentication

When PPP authentication is configured on an interface, a host requires that the other host uniquely identify itself with a secure password before establishing a PPP connection. The password is unique and is known to both hosts.

PPP supports the following authentication protocols:

- Challenge-Handshake Authentication Protocol (CHAP)

- Microsoft extension to the CHAP protocol (MS-CHAP)
- Password Authentication Protocol (PAP).

When you first enable PPP on a POS or serial interface, no authentication is enabled on the interface until you configure a CHAP, MS-CHAP, or PAP secret password under that interface. Keep the following information in mind when configuring PPP on an interface:

- CHAP, MS-CHAP, and PAP can be configured on a single interface; however, only one authentication method is used at any one time. The order in which the authentication protocols are used is determined by the peer during the LCP negotiations. The first authentication method used is the one that is also supported by the peer.
- PAP is the least secure authentication protocol available on POS and serial interfaces. To ensure higher security for information that is sent over POS and serial interfaces, we recommend configuring CHAP or MS-CHAP authentication in addition to PAP authentication.
- Enabling or disabling PPP authentication does not effect the local router's willingness to authenticate itself to the remote device.
- The **ppp authentication** command is also used to specify the order in which CHAP, MS-CHAP, and PAP authentication is selected on the interface. You can enable CHAP, MS-CHAP, or PAP in any order. If you enable all three methods, the first method specified is requested during link negotiation. If the peer suggests using the second method, or refuses the first method, the second method is tried. Some remote devices support only one method. Base the order in which you specify methods on the remote device's ability to correctly negotiate the appropriate method and on the level of data line security you require. PAP usernames and passwords are sent as clear text strings, which can be intercepted and reused.



**Caution** If you use a *list-name* value that was not configured with the **aaa authentication ppp** command, your interface cannot authenticate the peer. For details on implementing the **aaa authentication** command with the **ppp** keyword, see the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of *Cisco IOS XR System Security Command Reference* and *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.

## PAP Authentication

PAP provides a simple method for a remote node to establish its identity using a two-way handshake. After a PPP link is established between two hosts, a username and password pair is repeatedly sent by the remote node across the link (in clear text) until authentication is acknowledged, or until the connection is terminated.

PAP is not a secure authentication protocol. Passwords are sent across the link in clear text and there is no protection from playback or trial-and-error attacks. The remote node is in control of the frequency and timing of the login attempts.

## CHAP Authentication

CHAP is defined in RFC 1994, and it verifies the identity of the peer by means of a three-way handshake. The steps that follow provide a general overview of the CHAP process:

**SUMMARY STEPS**

1. The CHAP authenticator sends a challenge message to the peer.
2. The peer responds with a value calculated through a one-way hash function.
3. The authenticator checks the response against its own calculation of the expected hash value. If the values match, then the authentication is successful. If the values do not match, then the connection is terminated.

**DETAILED STEPS**

**Step 1** The CHAP authenticator sends a challenge message to the peer.

**Step 2** The peer responds with a value calculated through a one-way hash function.

**Step 3** The authenticator checks the response against its own calculation of the expected hash value. If the values match, then the authentication is successful. If the values do not match, then the connection is terminated.

This authentication method depends on a CHAP password known only to the authenticator and the peer. The CHAP password is not sent over the link. Although the authentication is only one-way, you can negotiate CHAP in both directions, with the help of the same CHAP password set for mutual authentication.

**Note** For CHAP authentication to be valid, the CHAP password must be identical on both hosts.

## MS-CHAP Authentication

Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) is the Microsoft version of CHAP and is an extension to RFC 1994. MS-CHAP follows the same authentication process used by CHAP. In this case, however, authentication occurs between a PC using Microsoft Windows NT or Microsoft Windows 95 and a Cisco router or access server acting as a network access server (NAS).

**Note** For MS-CHAP authentication to be valid, the MS-CHAP password must be identical on both hosts.

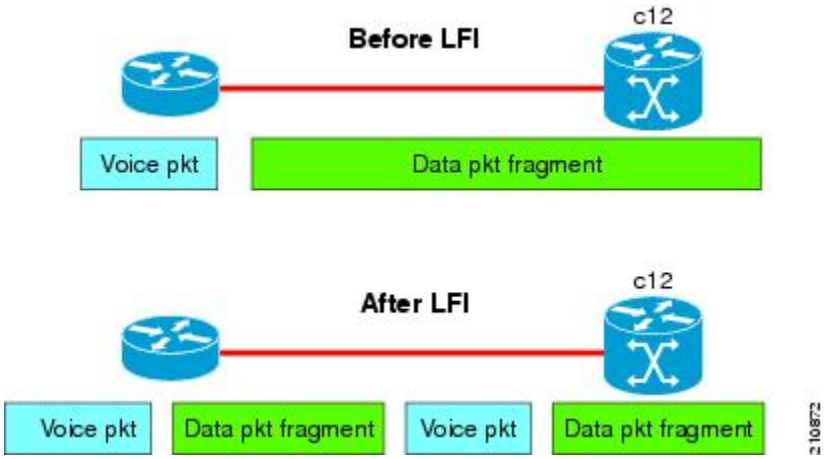
# Multilink PPP

Multilink Point-to-Point Protocol (MLPPP) provides a method for combining multiple physical links into one logical link. The implementation combines multiple PPP interfaces into one multilink interface. MLPPP performs the fragmenting, reassembling, and sequencing of datagrams across multiple PPP links.

Link Fragmentation and Interleaving (LFI) is designed for MLPPP interfaces and is required when integrating voice and data on low-speed interfaces.

Link Fragmentation and Interleaving (LFI) provides stability for delay-sensitive traffic, such as voice or video, traveling on the same circuit as data. Voice is susceptible to increased latency and jitter when the network processes large packets on low-speed interfaces. LFI reduces delay and jitter by fragmenting large datagrams and interleaving them with low-delay traffic packets.

*Figure 1: Link Fragmentation Interleave*




## MLPPP Feature Summary

MLPPP in Cisco IOS XR provides the same features that are supported on PPP Serial interfaces with the exception of QoS. It also provides the following additional features:

- Long sequence numbers (24-bit).
- Lost fragment detection timeout period of 1 second.
- Minimum-active-links configuration option.
- LCP echo request/reply support over multilink interface.
- Full T1 and E1 framed and unframed links.
- Support for the Cisco 2-Port Channelized OC-12c/DS0 SPA to set thresholds for noise errors on T1/E1 links that are used to signal the Noise Attribute to PPP for removal of an MLPPP bundle link. For more information about LNM, see the "Configuring Clear Channel T3/E3 Controllers and Channelized T3 Controllers on the Cisco ASR 9000 Series Router" module in the *Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide*.

## IPHC Over MLPPP

The 2-Port Channelized OC-12c/DS0 SPA supports IPHC over PPP, MLPPP, and MLPPP/LFI. For more information about IPHC and how to configure it, see the "Configuring Serial Interfaces on the Cisco ASR 9000 Series Router" module in the *Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide*.

# ICSSO for PPP and MLPPP

**Note** SR- and MR-APS is not supported on the Cisco 1-Port Channelized OC-48/STM-16 SPA.

Inter-Chassis Stateful Switchover (ICSSO) on the Cisco ASR 9000 Series Router provides features that maintain Point-to-Point Protocol (PPP) and Multilink PPP (MLPPP) sessions during a Multi-Router Automatic Protection Switching (MR-APS) switchover from the MR-APS Working router to the MR-APS Protect router.

ICSSO allows an MR-APS switchover to occur without the need for Link Control Protocol (LCP) or IP Control Protocol (IPCP) renegotiation between the new MR-APS active router and the remote PPP/MLPPP peer devices. The primary purpose of ICSSO is to minimize subscriber session and data loss during an MR-APS switchover.

ICSSO synchronizes the PPP and MLPPP state information on the active router with the state information on the backup router, and ensures that the backup router is ready to forward traffic immediately after an MR-APS switchover.

ICSSO works in conjunction with the following other software components:

## Multi-Router Automatic Protection Switching (MR-APS)

Multi-Router Automatic Protection Switching (MR-APS) is a Cisco feature that provides Layer 1 protection against facility and equipment failures through the configuration of a protection pair of SONET controllers located on two different routers. The redundant backup router is configured identically to the active router and is ready to forward traffic immediately upon an MR-APS switchover.

The protection pair communicates using Layer 1 (k1/k2) signalling bytes from the SONET downstream connection (as per Bellcore specification GR-253-CORE) and Layer 3 signaling messages using Protect Group Protocol (PGP). MR-APS detects many of the sources of failures that indirectly trigger an IP-FRR update to use backup routes.

In an MR-APS configuration, two interfaces, on different routers, are assigned the roles of Working interface or Protect interface. These roles are configured by the operator. Under normal conditions, the Working interface carries active traffic. If the Working interface fails, the Protect interface takes over the active traffic immediately with no loss of PPP traffic.

## Session State Redundancy Protocol (SSRP)

A pair of SONET controllers configured for MR-APS are part of a Session State Redundancy Protocol (SSRP) protection group. SSRP communicates interface and system state information between the Active and Standby routers. SSRP also serves as the keepalive protocol.

SSRP configuration associates a SONET controller with an inter-chassis redundancy group and enables MR-APS peer routers to synchronize PPP session states on each Active SONET controller.

PPP sessions can have one of three states:

- Active–A PPP session is in the Active state when the PPP session negotiation is complete, the associated route is installed, and the associated adjacency is created. PPP sessions in the Active state replicate data to their peers on the Standby router.
- Standby Up–A PPP session on the Standby router is in the Standby Up state when replicated state information is received from the Active router, the associated PPP route is installed, and the associated adjacency is created. PPP sessions in the Standby Up state are ready to forward traffic immediately after an MR-APS switchover.
- Standby Down–A PPP session on the Standby router is in the Standby Down state when the associated route is not installed and the adjacency is not created.

SSRP runs between the MR-APS peer routers and uses TCP/IP. One SSRP session runs on each pair of redundant SONET controllers, meaning multiple SSRP sessions can be running on a pair of MR-APS-redundant routers.

**Note** SSRP is not a redundancy control protocol, but is a state information synchronization protocol.

## Redundancy Group Manager (RG-MGR)

The Redundancy Group Manager (RG-MGR) configures the backup routes for the protected interface. The RG-MGR registers events on protected SONET controllers and provides the Routing Information Base (RIB) component with IP Fast Reroute (IP-FRR) updates.

## IP Fast Reroute (IP-FRR)

**Note** IP-FRR, when used with IC-SSO, is only supported with PPP encapsulation. It is not supported with HDLC encapsulation.

IP Fast Reroute (IP-FRR) provides extremely fast rerouting of PPP/MLPPP traffic after an MR-APS switchover.

IP-FRR controls the primary and backup routes. Each route is mapped in the Routing Information Base (RIB), and IP-FRR controls which backup path is used to forward traffic after an MR-APS switchover.

An MR-APS switchover triggers an IP-FRR update, which activates the backup routes on the protection SONET controller. When the working SONET controller is restored, another IP-FRR update is triggered, and traffic is rerouted to the primary route.

For more information about IP-FRR, refer to the “Implementing MPLS Traffic Engineering on Cisco IOS XR Software” module in the *Cisco IOS XR MPLS Configuration Guide*.

## VPN Routing And Forwarding (VRF)

ICSSO can be used with VPN routing and forwarding (VRF). Customers who wish to isolate traffic streams with different service types can do so using VRF technology. VRF allows the user to create and maintain separate routing and forwarding databases. See VRF on Multilink Configuration for Use with ICSSO: Example and VRF on Ethernet Configuration for Use with ICSSO: Example. For more information on configuring VRF, refer to the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

## Open Shortest Path First (OSPF)

Aggregation routers that terminate PPP sessions to a set of remote peers, must advertise their availability on the network using Open Shortest Path First (OSPF). OSPF is required to advertise the availability of remote PPP peers to the ICSSO peer router. See OSPF Configuration for Use with ICSSO: Example. For more information on configuring OSPF, refer to the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

## ICSSO Configuration Overview

ICSSO is configured as follows:

- Configure MR-APS
- Configure SSRP profile
- Configure SSRP groups
- Configure serial interfaces with PPP encapsulation
- Configure multilink interfaces
- Verify ICSSO configuration

The Configuring ICSSO for PPP and MLPPP of this module provides step procedures for configuring ICSSO.

The ICSSO for PPP and MLPPP Configuration: Examples gives specific examples for configuring ICSSO and related components.

# Multiclass MLPPP with QoS

Multiclass Multilink Point-to-Point Protocol (MLPPP) can be utilized with Quality of Service (QoS) and configured using the **encap-sequence** command under a class in a policy map.

The **encap-sequence** command specifies the MLPPP MCMP class ID for the packets in an MQC defined class.

The valid values for the **encap-sequence** ID number are **none**, 0, 1, 2, or 3. The **none** value is applicable only when the **priority level** is 1 and indicates that there is no MLPPP encapsulation. The values 1, 2, or 3 can be used with priority 1 or 2 classes or other classes with queuing actions. An **encap-sequence** ID number of zero (0) is reserved for the default class and cannot be specified in any other classes.

**Note** The **encap-sequence** ID numbers must be configured in numeric order. For example, you cannot assign an ID number of 3 unless you have already assigned 1 and 2.

The number of **encap-sequence** ID numbers must be less than the number of MLPPP classes that are negotiated between the peers via the Multilink header. The user must ensure that the configuration is consistent as the system does not verify this.

The **ppp multilink multiclass remote apply** command provides a way to ensure this. You can ensure that the number of classes using an **encap-sequence** ID number (including the default of 0) is less than the *min-number* value in the **ppp multilink multiclass remote apply** command. For example, if the min-number value in the ppp multilink multiclass remote apply command is 4, you can only have 3 or less classes with encap-sequence ID numbers

The QoS policy validates the following conditions. If these conditions are not met, the policy is rejected:

- The **encap-sequence** ID number is within the allowed values of 1 to 3.
- When **encap-sequence** is configured for any class in a policy map, all classes in that policy map with **priority level 1** must also contain an **encap-sequence** ID number.
- The **encap-sequence none** configuration is restricted to classes with **priority level** 1.
- The class-default does not contain an **encap-sequence** configuration.
- Only classes containing a queuing action have the **encap-sequence** configuration.

**Note** Classes that share the same **encap-sequence** ID number must have the same priority.

A QoS policy map is configured as follows:

```
config
 policy-map type qos policy-name
  class class-name
   action
   action
   action
. . .
```

The following example shows how to configure a policy map for MLPPP:

```
config
 policy-map foo
  class ip-prec-1
   encap-sequence none
   police rate percent 10
   priority level 1
 !
  class ip-prec-2
   encap-sequence 1
   shape average percent 80
 !
  class ip-prec-3
   encap-sequence 1
   bandwidth percent 10
 !
  class class-default
 !
 end-policy-map
 !
```

For complete information on configuring QoS and QoS commands, refer to the *Cisco ASR 9000 Series Aggregation Services Routers Modular Quality of Service Configuration Guide* and the *Cisco ASR 9000 Series Aggregation Services Routers Modular Quality of Service Command Reference*.

# T3 SONET Channels

The Cisco ASR 9000 Series Router supports T3 channelized SONET on the following hardware:

- SIP 700 SPA Interface Processor
- 1-Port Channelized OC-3/STM-1 SPA
- 2-Port Channelized OC-12c/DS0 SPA
- 1-Port Channelized OC-48/STM-16 SPA

Channelized SONET provides the ability to transport multiple T3 channels over the same physical link.

For more detailed information about configuring channelized SONET, T3 and T1 controllers, serial interfaces, and SONET APS, see the following related modules:

- Configuring Channelized SONET/SDH
- Configuring Clear Channel SONET Controllers
- Configuring Clear Channel T3/E3 Controllers and Channelized T3 and T1/E1 Controllers
- Configuring Serial Interfaces

# How to Configure PPP

This section includes the following procedures:

## Modifying the Default PPP Configuration

When you first enable PPP on an interface, the following default configuration applies:

- The interface resets itself immediately after an authentication failure.
- The maximum number of configuration requests without response permitted before all requests are stopped is 10.
- The maximum number of consecutive Configure Negative Acknowledgments (CONFNAKs) permitted before terminating a negotiation is 5.
- The maximum number of terminate requests (TermReqs) without response permitted before the Link Control Protocol (LCP) or Network Control Protocol (NCP) is closed is 2.
- Maximum time to wait for a response to an authentication packet is 10 seconds.
- Maximum time to wait for a response during PPP negotiation is 3 seconds.

This task explains how to modify the basic PPP configuration on serial and POS interfaces that have PPP encapsulation enabled. The commands in this task apply to all authentication types supported by PPP (CHAP, MS-CHAP, and PAP).

### Before you begin

You must enable PPP encapsulation on the interface with the **encapsulation ppp** command.

- To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces module in this manual.
- To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces module in this manual.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp max-bad-auth** *retries*
4. **ppp max-configure** *retries*
5. **ppp max-failure** *retries*
6. **ppp max-terminate** *number*

7. **ppp timeout authentication** *seconds*
8. **ppp timeout retry** *seconds*
9. **end** or **commit**
10. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp max-bad-auth** *retries*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp max-bad-auth 3` | (Optional) Configures the number of authentication retries allowed on an interface after a PPP authentication failure.<br>• If you do not specify the number of authentication retries allowed, the router resets itself immediately after an authentication failure.<br>• Replace the *retries* argument with number of retries after which the interface is to reset itself, in the range from 0 through 10.<br>• The default is 0 retries.<br>• The **ppp max-bad-auth** command applies to any interface on which PPP encapsulation is enabled. |
| **Step 4** | **ppp max-configure** *retries*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp max-configure 4` | (Optional) Specifies the maximum number of configure requests to attempt (without response) before the requests are stopped.<br>• Replace the *retries* argument with the maximum number of configure requests retries, in the range from 4 through 20.<br>• The default maximum number of configure requests is 10.<br>• If a configure request message receives a reply before the maximum number of configure requests are sent, further configure requests are abandoned. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **ppp max-failure** *retries*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp max-failure 3` | (Optional) Configures the maximum number of consecutive Configure Negative Acknowledgments (CONFNAKs) permitted before a negotiation is terminated.<br>• Replace the *retries* argument with the maximum number of CONFNAKs to permit before terminating a negotiation, in the range from 2 through 10.<br>• The default maximum number of CONFNAKs is 5. |
| **Step 6** | **ppp max-terminate** *number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp max-terminate 5` | (Optional) Configures the maximum number of terminate requests (TermReqs) to send without reply before the Link Control Protocol (LCP) or Network Control Protocol (NCP) is closed.<br>• Replace the *number* argument with the maximum number of TermReqs to send without reply before closing down the LCP or NCP. Range is from 2 to 10.<br>• The default maximum number of TermReqs is 2. |
| **Step 7** | **ppp timeout authentication** *seconds*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp timeout authentication 20` | (Optional) Sets PPP authentication timeout parameters.<br>• Replace the *seconds* argument with the maximum time, in seconds, to wait for a response to an authentication packet. Range is from 3 to 30 seconds.<br>• The default authentication time is 10 seconds, which should allow time for a remote router to authenticate and authorize the connection and provide a response. However, it is also possible that it will take much less time than 10 seconds. In such cases, use the **ppp timeout authentication** command to lower the timeout period to improve connection times in the event that an authentication response is lost. |
| **Step 8** | **ppp timeout retry** *seconds*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp timeout retry 8` | (Optional) Sets PPP timeout retry parameters.<br>• Replace the *seconds* argument with the maximum time, in seconds, to wait for a response during PPP negotiation. Range is from 1 to 10 seconds.<br>• The default is 3 seconds. |
| **Step 9** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)?`<br>`[cancel]:` |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config-if)# commit` | - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 10** | **show ppp interfaces** {*type interface-path-id* \| **all** \| **brief** {*type interface-path-id* \| **all** \| **location** *node-id*} \| **detail** {*type interface-path-id* \| **all** \| **location** *node-id*} \| **location** *node-id*}<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show ppp interfaces serial 0/2/0/0` | Verifies the PPP configuration for an interface or for all interfaces that have PPP encapsulation enabled. |

# Configuring PPP Authentication

This section contains the following procedures:

## Enabling PAP, CHAP, and MS-CHAP Authentication

This task explains how to enable PAP, CHAP, and MS-CHAP authentication on a serial or POS interface.

### Before you begin

You must enable PPP encapsulation on the interface with the **encapsulation ppp** command, as described in the following modules:

- To enable PPP encapsulation on a POS interface, see the Configuring POS Interfaces module in this manual.
- To enable PPP encapsulation on an interface, see the Configuring Serial Interfaces module in this manual.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**]
4. **end** or **commit**

5. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* \| **default**]<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access` | Enables CHAP, MS-CHAP, or PAP on an interface, and specifies the order in which CHAP, MS-CHAP, and PAP authentication is selected on the interface.<br>• Replace the *protocol* argument with **pap**, **chap**, **or ms-chap**.<br>• Replace the *list name* argument with the name of a list of methods of authentication to use. To create a list, use the **aaa authentication ppp** command, as described in the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Command Reference*.<br>• If no list name is specified, the system uses the default. The default list is designated with the **aaa authentication ppp** command, as described in the *Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Command Reference*. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | | - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show ppp interfaces** {*type interface-path-id* \| **all** \| **brief** {*type interface-path-id* \| **all** \| **location** *node-id*} \| **detail** {*type interface-path-id* \| **all** \| **location** *node-id*} \| **location** *node-id*}<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show ppp interfaces serial 0/2/0/0` | Displays PPP state information for an interface.<br>• Enter the *type interface-path-id* argument to display PPP information for a specific interface.<br>• Enter the **brief** keyword to display brief output for all interfaces on the router, for a specific interface instance, or for all interfaces on a specific node.<br>• Enter the **all** keyword to display detailed PPP information for all nodes installed in the router.<br>• Enter the **location** *node-id* keyword argument to display detailed PPP information for the designated node.<br>There are seven possible PPP states applicable for either the Link Control Protocol (LCP) or the Network Control Protocol (NCP). |

### What to do next

Configure a PAP, CHAP, or MS-CHAP authentication password, as described in the appropriate section:

- If you enabled PAP on an interface, configure a PAP authentication username and password, as described in the “Configuring a PAP Authentication Password” section on page 641.
- If you enabled CHAP on an interface, configure a CHAP authentication password, as described in the “Configuring a CHAP Authentication Password” section on page 643
- If you enabled MS-CHAP on an interface, configure an MS-CHAP authentication password, as described in the “Configuring an MS-CHAP Authentication Password” section on page 645

## Configuring a PAP Authentication Password

This task explains how to enable and configure PAP authentication on a serial or POS interface.

**Note** PAP is the least secure authentication protocol available on POS and interfaces. To ensure higher security for information that is sent over POS and interfaces, we recommend configuring CHAP or MS-CHAP authentication in addition to PAP authentication.

### Before you begin

You must enable PAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

## SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp pap sent-username** *username* **password** [**clear** \| **encrypted**] *password*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp pap sent-username xxxx password notified` | Enables remote Password Authentication Protocol (PAP) support for an interface, and includes the **sent-username** and **password** commands in the PAP authentication request packet to the peer.<br>• Replace the *username* argument with the username sent in the PAP authentication request.<br>• Enter **password clear** to select cleartext encryption for the password, or enter **password encrypted** if the password is already encrypted.<br>• The **ppp pap sent-username** command allows you to replace several username and password configuration commands with a single copy of this command on interfaces. |

| | Command or Action | Purpose |
|---|---|---|
| | | • You must configure the **ppp pap sent-username** command for each interface.<br>• Remote PAP support is disabled by default. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

## Configuring a CHAP Authentication Password

This task explains how to enable CHAP authentication and configure a CHAP password on a serial or POS interface.

### Before you begin

You must enable CHAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

### Restrictions

The same CHAP password must be configured on both host endpoints.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*

3. **ppp chap password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp chap password** [**clear** | **encrypted**] *password*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp chap password clear xxxx` | Enables CHAP authentication on the specified interface, and defines an interface-specific CHAP password.<br>• Enter **clear** to select cleartext encryption, or **encrypted** if the password is already encrypted.<br>• Replace the *password* argument with a cleartext or already-encrypted password. This password is used to authenticate secure communications among a collection of routers.<br>• The **ppp chap password** command is used for remote CHAP authentication only (when routers authenticate to the peer) and does not effect local CHAP authentication.This command is useful when you are trying to authenticate a peer that does not support this command (such as a router running an older Cisco IOS XR software image).<br>• The CHAP secret password is used by the routers in response to challenges from an unknown peer. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | | - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

## Configuring an MS-CHAP Authentication Password

This task explains how to enable MS-CHAP authentication and configure an MS-CHAP password on a serial or POS interface.

### Before you begin

You must enable MS-CHAP authentication on the interface with the **ppp authentication** command, as described in the Enabling PAP, CHAP, and MS-CHAP Authentication.

### Restrictions

The same MS-CHAP password must be configured on both host endpoints.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap password** [**clear** | **encrypted**] *password*
4. **end** or **commit**
5. **show running-config**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:** | Enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | |
| **Step 3** | **ppp ms-chap password** [**clear** \| **encrypted**] *password*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap password clear xxxx` | Enables a router calling a collection of routers to configure a common Microsoft Challenge Handshake Authentication (MS-CHAP) secret password.<br>The MS-CHAP secret password is used by the routers in response to challenges from an unknown peer. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

# Disabling an Authentication Protocol

This section contains the following procedures:

## Disabling PAP Authentication on an Interface

This task explains how to disable PAP authentication on a serial or POS interface.

### SUMMARY STEPS

1. **configure**

2. **interface** *type interface-path-id*
3. **ppp pap refuse**
4. **end** or **commit**
5. **show running-config**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp pap refuse**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp pap refuse` | Refuses Password Authentication Protocol (PAP) authentication from peers requesting it.<br>• If outbound Challenge Handshake Authentication Protocol (CHAP) has been configured (using the **ppp authentication** command), CHAP will be suggested as the authentication method in the refusal packet.<br>• PAP authentication is disabled by default. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

## Disabling CHAP Authentication on an Interface

This task explains how to disable CHAP authentication on a serial or POS interface.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp chap refuse**
4. **end** or **commit**
5. **show running-config**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |
| **Step 3** | **ppp chap refuse**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp chap refuse` | Refuses CHAP authentication from peers requesting it. After you enter the **ppp chap refuse** command under the specified interface, all attempts by the peer to force the user to authenticate with the help of CHAP are refused.<br>• CHAP authentication is disabled by default.<br>• If outbound Password Authentication Protocol (PAP) has been configured (using the **ppp authentication** command), PAP will be suggested as the authentication method in the refusal packet. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before` |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config-if)# commit` | `exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

## Disabling MS-CHAP Authentication on an Interface

This task explains how to disable MS-CHAP authentication on a serial or POS interface.

### SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap refuse**
4. **end** or **commit**
5. **show running-config**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1` | Enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **ppp ms-chap refuse**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap refuse` | Refuses MS-CHAP authentication from peers requesting it. After you enter the **ppp ms-chap refuse** command under the specified interface, all attempts by the peer to force the user to authenticate with the help of MS-CHAP are refused.<br>• MS-CHAP authentication is disabled by default.<br>• If outbound Password Authentication Protocol (PAP) has been configured (using the **ppp authentication** command), PAP will be suggested as the authentication method in the refusal packet. |
| **Step 4** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |
| **Step 5** | **show running-config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running-config` | Verifies PPP authentication information for interfaces that have PPP encapsulation enabled. |

# Configuring Multilink PPP

This section contains the following procedures:

### Before you begin

- MLPPP and LFI are supported on the 1-Port Channelized OC-3/STM-1 SPA and 2-Port Channelized OC-12/DS0 SPA.

### Restrictions

MLPPP for Cisco IOS XR software has the following restrictions:

- Only full rate T1s are supported.
- All links in a bundle must belong to the same SPA.
- All links in a bundle must operate at the same speed.
- A maximum of 10 links per bundle is supported.
- A maximum of 700 bundles per line card is supported.
- A maximum of 2600 bundles per system is supported.
- MLPPP interfaces are not supported with DS0 link members.
- MLPPP interfaces are not be supported with T3 channels as members. Therefore, LFI is also unsupported on T3 channels.
- All serial links in an MLPPP bundle inherit the value of the **mtu** command from the multilink interface. Therefore, you should not configure the **mtu** command on a serial interface before configuring it as a member of an MLPPP bundle. The Cisco IOS XR software blocks the following:
  - Attempts to configure a serial interface as a member of an MLPPP bundle if the interface is configured with a nondefault MTU value.
  - Attempts to change the **mtu** command value for a serial interface that is configured as a member of an MLPPP bundle.

    In Cisco IOS XR software, multilink processing is controlled by a hardware module called the Multilink Controller, which consists of an ASIC, network processor, and CPU working in conjunction. The MgmtMultilink Controller makes the multilink interfaces behave like the serial interfaces of channelized SPAs.

## Configuring the Controller

Perform this task to configure the controller.

### SUMMARY STEPS

1. **configure**
2. **controller** *type interface-path-id*
3. **mode** *type*
4. **clock source** {**internal** | **line**}
5. **exit**
6. **controller t1** *interface-path-id*
7. **channel-group** *channel-group-number*
8. **timeslots** *range*
9. **exit**
10. **exit**
11. **controller mgmtmultilink** *interface-path-id*
12. **bundle** *bundle-id*
13. **end** or **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **controller** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0` | Enters controller configuration submode and specifies the controller name and instance identifier in *rack/slot/module/port* notation. |
| **Step 3** | **mode** *type*<br>**Example:**<br>`RP/0/RSP0/CPU0:router# mode t1` | Configures the type of multilinks to channelize; for example, 28 T1s. |
| **Step 4** | **clock source** {**internal** \| **line**}<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t3)# clock source internal` | (Optional) Configures the clocking for the port.<br>**Note** • The default clock source is **internal**. |
| **Step 5** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t3)# exit` | Exits controller configuration mode. |
| **Step 6** | **controller t1** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/1` | Enters T1 configuration mode. |
| **Step 7** | **channel-group** *channel-group-number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t1)# channel-group 0` | Creates a T1 channel group and enters channel group configuration mode for that channel group. Channel group numbers can range from 0 to 23. |
| **Step 8** | **timeslots** *range*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24` | Associates one or more DS0 time slots to a channel group and creates an associated serial subinterface on that channel group.<br>• Range is from 1 to 24 time slots.<br>• **Note** • The time slot range must be from 1 to 24 for the resulting serial interface to be accepted into a MLPPP bundle. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit` | Exits channel group configuration mode. |
| **Step 10** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t1)# exit` | Exits T1 configuration mode and enters global configuration mode. |
| **Step 11** | **controller mgmtmultilink** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0` | Enters controller configuration submode for the management of multilink interfaces. Specify the controller name and instance identifier in *rack/slot/module/port* notation. |
| **Step 12** | **bundle** *bundle-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 20` | Creates a multilink interface with the specified bundle ID. |
| **Step 13** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t3)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-t3)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Configuring the Interfaces

Perform this task to configure the interfaces.

### Restrictions

- All serial links in an MLPPP bundle inherit the value of the **mtu** command from the multilink interface. Therefore, you should not configure the **mtu** command on a serial interface before configuring it as a member of an MLPPP bundle. The Cisco IOS XR software blocks the following:
  - Attempts to configure a serial interface as a member of an MLPPP bundle if the interface is configured with a nondefault MTU value.
  - Attempts to change the **mtu** command value for a serial interface that is configured as a member of an MLPPP bundle.

## SUMMARY STEPS

1. **configure**
2. **interface multilink** *interface-path-id*
3. **ipv4 address** *ip-address*
4. **multilink fragment-size** *bytes* or **multilink fragment delay** *delay-ms*
5. **keepalive** {*interval* | **disable**}[*retry*]
6. **exit**
7. **interface** *type interface-path-id*
8. **encapsulation** *type*
9. **multilink group** *group-id*
10. **end** or **commit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface multilink** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1` | Specifies the multilink interface name and instance identifier in *rack/slot/module/port/bundle-id* notation, and enters interface configuration mode. |
| **Step 3** | **ipv4 address** *ip-address*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24` | Assigns an IP address and subnet mask to the interface in the format:<br>*A.B.C.D/prefix* or *A.B.C.D/mask* |
| **Step 4** | **multilink fragment-size** *bytes* or **multilink fragment delay** *delay-ms*<br>**Example:** | (Optional) Specifies the size of the multilink fragments, such as 128 bytes. Some fragment sizes may not be supported. The default is no fragments.<br>or |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config-if)# multilink fragment-size 350`<br>or<br>`RP/0/RSP0/CPU0:router(config-if)# multilink fragment delay 2` | (Optional) Specifies the multilink fragment delay in milliseconds. This sets the MLPPP fragment size so that it is equivalent in length to the transmission time delay for any individual member-link (T1s with bandwidths of 1536000bps/192000Bps).<br>If the user specifies **fragment delay 2,** the fragment size is (192000*.002)=384B. The usage of this command is exclusive to the usage of **fragment size**. Either command overrides the other. |
| **Step 5** | **keepalive** {*interval* \| **disable**}[*retry*]<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# keepalive disable` | Sets the keepalive timer for the channel, where:<br>• *interval*—Number of seconds (from 1 to 30) between keepalive messages. The default is 10.<br>• **disable**—Turns off the keepalive timer.<br>• *retry*—(Optional) Number of keepalive messages (from 1 to 255) that can be sent to a peer without a response before transitioning the link to the down state. The default is 3.<br>**Note** • To connect with some Cisco IOS devices, multilink keepalives need to be disabled on both devices. |
| **Step 6** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# exit` | Exits interface configuration mode and enters global configuration mode. |
| **Step 7** | **interface** *type interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1:0` | Specifies the interface name and instance identifier in *rack/slot/module/port/t1-number:channel-group* notation, and enters interface configuration mode. |
| **Step 8** | **encapsulation** *type*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp` | Specifies the type of encapsulation; in this case, PPP. |
| **Step 9** | **multilink group** *group-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# multilink group 20` | Specifies the multilink group ID for this interface. |
| **Step 10** | **end** or **commit** | Saves configuration changes. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br><br>`RP/0/RSP0/CPU0:router(config-t3)# end`<br><br>or<br><br>`RP/0/RSP0/CPU0:router(config-t3)# commit` | • When you issue the **end** command, the system prompts you to commit changes:<br><br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`<br><br>- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br><br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br><br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br><br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

## Configuring MLPPP Optional Features

Perform this task to configure either of the following optional features:

- Minimum number of active links
- Multilink interleave

**Note** Minimum number active links must be configured at both endpoints.

### SUMMARY STEPS

1. **configure**
2. **interface multilink** *interface-path-id*
3. **multilink**
4. **ppp multilink minimum-active links** *value*
5. **multilink interleave**
6. **no shutdown**
7. **end** or **commit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# configure` | Enters global configuration mode. |
| **Step 2** | **interface multilink** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1` | Specifies the multilink interface name and instance identifier in *rack/slot/module/port/bundle-id* notation, and enters interface configuration mode. |
| **Step 3** | **multilink**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# multilink` | Enters interface multilink configuration mode. |
| **Step 4** | **ppp multilink minimum-active links** *value*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 12` | (Optional) Specifies the minimum number of active links for the multilink interface.<br>**Note** • When support for the Noise Attribute is configured to signal PPP to remove links on MLPPP bundles when LNM thresholds are crossed on a link, the links will not be removed below this miminum-active threshold. |
| **Step 5** | **multilink interleave**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if-multilink)# multilink interleave` | (Optional) Enables interleave on a multilink interface. |
| **Step 6** | **no shutdown**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if-mutlilink)# no shutdown` | Removes the shutdown configuration.<br>• The removal of the shutdown configuration removes the forced administrative down on the controller, enabling the controller to move to an up or a down state. |
| **Step 7** | **end** or **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-t3)# end`<br>or<br>`RP/0/RSP0/CPU0:router(config-t3)# commit` | Saves configuration changes.<br>• When you issue the **end** command, the system prompts you to commit changes:<br>`Uncommitted changes found, commit them before exiting(yes/no/cancel)?`<br>`[cancel]:` |

| | Command or Action | Purpose |
|---|---|---|
| | | - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.<br>- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.<br>- Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.<br>• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session. |

# Configuring ICSSO for PPP and MLPPP

This section provides the following ICSSO configuration procedures:

### Before you begin

The Cisco ASR 9000 Series Router supports ICSSO in the following MR-APS, minimum equipment, hardware configurations:

- Two 6-slot or 8-slot chassis
- Four route/switch processors (RSPs), two per chassis (offers a higher degree of reliability)
- Two 20G SIPs, 1 per chassis
- Two of the following SPA types 1 per chassis:
  - 2-Port Channelized OC-12/DS0 SPA
  - 4-Port Channelized T3 SPA
  - 8-Port Channelized T1/E1 SPA
- Two 40 Gigabit Ethernet line cards, 2 per chassis
- Two 4-Port 10 Gigabit Ethernet line cards, 1 per chassis
- 1-Port Channelized OC-3/STM-1 SPA (SPA-1XCHSTM1/OC3)

### Restrictions

The following restrictions apply to ICSSO for PPP and MLPPP:

- ICSSO is supported only on two independent routers. ICSSO for two line cards on the same router is not supported.
- Automated synchronization or verification of the IOS XR system configuration between the ICSSO peer routers is not available.

- The following restrictions apply to ICSSO on the 2-Port Channelized OC-12/DS0 SPA:
  - ICSSO is supported only on T1/T3 PPP and T1/MLPPP interfaces.
  - T1 member links must terminate on the same SPA.
  - Member links in an MLPPP bundle being protected by MR-APS must all be contained in the same SONET port, this SONET port being a part of the MR-APS protection pair.
  - T1/PPP, T3/PPP and MLPPP encapsulated interfaces on the OC-12 SONET interface can be protected.
- The following restrictions apply to ICSSO on the 1-Port Channelized T3 SPA:
  - Supported for PPP on T3, T1, E1 channels only.
  - Supported for member links in an MLPPP on E1 channels only.
- The following restrictions apply to ICSSO on the 8-Port Channelized T1/E1 SPA:
  - Supported for PPP on T1 and E1 channels only.
  - Supported for member links in an MLPPP on E1 channels only.

## Configuring a Basic ICSSO Implementation

Use the following procedure to configure a simple version of ICSSO.

### SUMMARY STEPS

1. **config**
2. **redundancy**
3. **multi-router aps**
4. **group** *group_number*
5. **controller sonet** *path*
6. **member ipv4** *address* **backup-interface type** *interface-path-id*
7. **commit**
8. **show running config**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# config` | Enters global configuration mode. |
| **Step 2** | **redundancy**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# redundancy` | Enters redundancy configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **multi-router aps**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-redundancy)# multi-router aps` | Configures Multi-Router APS redundancy and enters APS redundancy configuration mode. |
| **Step 4** | **group** *group_number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-redundancy-aps)# group 1` | Configures the APS redundancy group and assigns the group number. |
| **Step 5** | **controller sonet** *path*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-redundancy-aps-group)# controller sonet 0/1/0/0` | Specifies a SONET controller as the APS redundancy backup. |
| **Step 6** | **member ipv4** *address* **backup-interface type** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-redundancy-group-controller)# member ipv4 10.10.10.10 backup-interface GigabitEthernet 0/6/0/1` | Specifies the IP address of the backup interface used by IP-FRR. |
| **Step 7** | **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-redundancy-group-controller)# commit` | Saves the configuration. |
| **Step 8** | **show running config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# show running config` | Displays the current configuration on the router, including MR-APS, SONET controller, and IP address information for verifying the configuration. |

## Configuring MR-APS

Use the following procedure to configure MR-APS.

### SUMMARY STEPS

1. **config**
2. **aps group** *number*
3. **channel {0 | 1} remote** *ip-address*
4. **channel {0 | 1} local sonet** *interface-path-id*
5. **exit**

6. **aps rprplus**
7. **interface GigabitEthernet** *interface-path-id*
8. **description** *text*
9. **ipv4 address** *ipv4-address mask*
10. **commit**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# config` | Enters global configuration mode. |
| **Step 2** | **aps group** *number*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# aps group 1` | Adds an automatic protection switching (APS) group and enter APS group configuration mode. |
| **Step 3** | **channel {0 \| 1} remote** *ip-address*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-aps)# channel 0 remote 99.10.1.2` | Assigns a port and interface that is physically located in a remote router as a SONET APS channel.<br>• 0 designates the channel as protect channel.<br>• 1 designates the channel as a working channel. |
| **Step 4** | **channel {0 \| 1} local sonet** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-aps)# channel 1 local SONET 0/1/0/0` | Assigns a local SONET physical port as a SONET APS channel.<br>• 0 designates the channel as protect channel.<br>• 1 designates the channel as a working channel. |
| **Step 5** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-aps)# exit` | Exits to the previous mode. |
| **Step 6** | **aps rprplus**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-aps)# aps rprplus` | Extends the APS hold timer for a switchover. |
| **Step 7** | **interface GigabitEthernet** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/6/0/0` | Creates a Gigabit Ethernet interface as the path to the MR-APS peer, and enters interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | **description** *text*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# description MR-APS PGP interface for aps group 1` | Adds a text description to this interface. |
| **Step 9** | **ipv4 address** *ipv4-address mask*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if )# ipv4 address 99.10.1.1 255.255.255.0` | Sets the primary IPv4 address and subnet mask for an interface. |
| **Step 10** | **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# commit` | Saves the current configuration. |

## Configuring SSRP on Serial and Multilink Interfaces

Use the following procedure to configure SSRP on serial and multilink interfaces:

### SUMMARY STEPS

1. **config**
2. **ssrp profile** *profile-name*
3. **peer ipv4 address** *A.B.C.D*
4. **exit**
5. **ssrp location** *node_id*
6. **group** *group-id* **profile** *profile_name*
7. **group** *group-id* **profile** *profile_name*
8. **exit**
9. **interface serial** *interface-path-id[.subinterface]*
10. **ssrp group** *group-number* **id** *id-number* **ppp**
11. **encapsulation ppp**
12. **multilink**
13. **group** *group-id*
14. **exit**
15. **keepalive disable**
16. **exit**
17. **interface serial** *interface-path-id[.subinterface]*
18. **ssrp group** *group-number* **id** *id-number* **ppp**
19. **encapsulation ppp**
20. **multilink**
21. **group** *group-id*
22. **exit**
23. **keepalive disable**

24. **exit**
25. **interface multilink** *interface-path-id*
26. **ipv4 address** *ipv4-address mask*
27. **ssrp group** *group-number* **id** *id-number* **ppp**
28. **encapsulation ppp**
29. **shutdown**
30. **keepalive disable**
31. **exit**
32. **controller MgmtMultilink** *interface-path-id*
33. **bundle** *bundleID*
34. **bundle** *bundleID*
35. **commit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **config**<br>**Example:**<br>`RP/0/RSP0/CPU0:router# config` | Enters global configuration mode. |
| **Step 2** | **ssrp profile** *profile-name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# ssrp profile Profile_1` | Configures the Session State Redundancy Protocol (SSRP) profile and enters the SSRP configuration mode. |
| **Step 3** | **peer ipv4 address** *A.B.C.D*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# peer ipv4 address 10.10.10.10` | Configures the IPv4 address for a Session State Redundancy Protocol (SSRP) peer. |
| **Step 4** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-aps)# exit` | Exits to the previous mode. |
| **Step 5** | **ssrp location** *node_id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# ssrp location 0/1/CPU0` | Specifies the node on which to create a Session State Redundancy Protocol (SSRP) group and enters the SSRP node configuration mode |
| **Step 6** | **group** *group-id* **profile** *profile_name*<br>**Example:** | Creates a Session State Redundancy Protocol (SSRP) group and associates it with a profile. |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config-ssrp)# group 1 profile Profile_1` | |
| **Step 7** | **group** *group-id* **profile** *profile_name*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-ssrp-node)# group 2 profile Profile_2` | Creates a second Session State Redundancy Protocol (SSRP) group and associates it with a profile. |
| **Step 8** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-ssrp-node)# exit` | Exits to the previous mode. |
| **Step 9** | **interface serial** *interface-path-id[.subinterface]*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/1:0` | Physical interface or virtual interface.<br>**Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>For more information about the syntax for the router, use the question mark (**?**) online help function. |
| **Step 10** | **ssrp group** *group-number* **id** *id-number* **ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 1 ppp` | Attaches an SSRP group on the interface. |
| **Step 11** | **encapsulation ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp` | Enables encapsulation for communication with routers using the Point-to-Point Protocol (PPP). |
| **Step 12** | **multilink**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# multilink` | Enters the multilink interface configuration mode. |
| **Step 13** | **group** *group-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# group 1` | Attaches a Session State Redundancy Protocol (SSRP) group to this interface. |
| **Step 14** | **exit**<br>**Example:** | Exits to the previous mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `RP/0/RSP0/CPU0:router(config)# exit` | |
| **Step 15** | **keepalive disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# keepalive disable` | Disables the keepalive timer for this interface. |
| **Step 16** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# exit` | Exits to the previous mode. |
| **Step 17** | **interface serial** *interface-path-id[.subinterface]*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/2:0` | Physical interface or virtual interface.<br>**Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>For more information about the syntax for the router, use the question mark (**?**) online help function. |
| **Step 18** | **ssrp group** *group-number* **id** *id-number* **ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 2 ppp` | Attaches an SSRP group on the interface. |
| **Step 19** | **encapsulation ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp` | Enables encapsulation for communication with routers using the Point-to-Point Protocol (PPP). |
| **Step 20** | **multilink**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# multilink` | Enters the multilink interface configuration mode. |
| **Step 21** | **group** *group-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# group 1` | Attaches a Session State Redundancy Protocol (SSRP) group to this interface. |
| **Step 22** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# exit` | Exits to the previous mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 23** | **keepalive disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# keepalive disable` | Disables the keepalive timer for this interface. |
| **Step 24** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# exit` | Exits to the previous mode. |
| **Step 25** | **interface multilink** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# interface Multilink 0/1/0/0/1` | Physical interface or virtual interface.<br>**Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>For more information about the syntax for the router, use the question mark (**?**) online help function. |
| **Step 26** | **ipv4 address** *ipv4-address mask*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if )# ipv4 address 10.10.10.10 255.255.255.0` | Sets the primary IPv4 address and subnet mask for an interface. |
| **Step 27** | **ssrp group** *group-number* **id** *id-number* **ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 3 ppp` | Attaches an SSRP group on the interface. |
| **Step 28** | **encapsulation ppp**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp` | Enables encapsulation for communication with routers using the Point-to-Point Protocol (PPP). |
| **Step 29** | **shutdown**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# shutdown` | Brings the interface administratively down for configuration. |
| **Step 30** | **keepalive disable**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# keepalive disable` | Disables the keepalive timer for this interface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 31** | **exit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-if)# exit` | Exits to the previous mode. |
| **Step 32** | **controller MgmtMultilink** *interface-path-id*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config)# controller MgmtMultilink 0/1/0/0` | Configure a controller for a generic multilink bundle and enters MgmtMultilink configuration mode. |
| **Step 33** | **bundle** *bundleID*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 1` | Creates a multilink interface bundle. |
| **Step 34** | **bundle** *bundleID*<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 2` | Creates a multilink interface bundle. |
| **Step 35** | **commit**<br>**Example:**<br>`RP/0/RSP0/CPU0:router(config-mgmtmultilink)# commit` | Saves the current configuration. |

# Configuration Examples for PPP

This section provides the following configuration examples:

## Configuring a POS Interface with PPP Encapsulation: Example

The following example shows how to create and configure a POS interface with PPP encapsulation:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# ppp pap sent-username P1_TEST-8 password xxxx
RP/0/RSP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access
RP/0/RSP0/CPU0:router(config-if)# ppp chap password encrypted xxxx
RP/0/RSP0/CPU0:router(config-if)# end
```

```
Uncommitted changes found, commit them? [yes]: yes
```

The following example shows how to configure POS interface 0/3/0/1 to allow two additional retries after an initial authentication failure (for a total of three failed authentication attempts):

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/RSP0/CPU0:router(config-if)# ppp max-bad-auth 3
```

# Configuring a Serial Interface with PPP Encapsulation: Example

The following example shows how to create and configure a serial interface with PPP MS-CHAP encapsulation:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# ppp authentication ms-chap MIS-access
RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap password encrypted xxxx
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

# Configuring MLPPP: Example

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0/1
RP/0/RSP0/CPU0:router# mode t1
RP/0/RSP0/CPU0:router(config-t3)# clock source internal
RP/0/RSP0/CPU0:router(config-t3)# exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/1/1
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit
RP/0/RSP0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0
RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 20
RP/0/RSP0/CPU0:router(config-t3)# commit
RP/0/RSP0/CPU0:router(config-t3)# exit

RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/20
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24
RP/0/RSP0/CPU0:router(config-if)# multilink fragment-size 128
RP/0/RSP0/CPU0:router(config-if)# keepalive disable
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/1:0
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# multilink group 20
RP/0/RSP0/CPU0:router(config-t3)# commit
```

```
RP/0/RSP0/CPU0:router(config-t3)# exit

RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1
RP/0/RSP0/CPU0:router(config-if)# multilink
RP/0/RSP0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 10

RP/0/RSP0/CPU0:router(config-if-multilink)# multilink interleave
RP/0/RSP0/CPU0:router(config-if-mutlilink)# no shutdown
RP/0/RSP0/CPU0:router(config-t3)# commit
```

# ICSSO for PPP and MLPPP Configuration: Examples

This section provides the following examples of ICSSO configuration and related configurations:

## ICSSO Configuration: Example

The following example shows how to configure ICSSO on a SONET controller:

```
config
 redundancy
  multi-router aps
  group 1
  controller sonet 0/1/0/0
   member ipv4 10.10.10.10 backup-interface GigabitEthernet 0/6/0/1
   commit
show running config
```

## Channelized SONET Controller Configuration for Use with ICSSO: Example

The following example shows how to configure channelized SONET controllers for use with ICSSO:

```
config
 controller SONET0/7/1/0
  framing sonet
  sts 1
  mode t3
!
  sts 2
  mode t3
!
  sts 3
  mode t3
!
 controller T3 0/7/0/1
  mode t1
  framing auto-detect
!
 controller T1 0/7/0/1/1
  channel-group 0
  timeslots 1-24
```

# MR-APS Configuration: Example

The following example shows how to configure MR-APS:

```
config
 aps group 1
  channel 0 remote 99.10.1.2
  channel 1 local SONET0/1/0/0
!
 aps rprplus
!
 interface GigabitEthernet0/6/0/0
  description MR-APS PGP interface for aps group 1
  ipv4 address 99.10.1.1 255.255.255.0
```

The following example shows how to configure a redundancy group manager:

```
// mr-aps part:
aps group 1
 channel 0 remote 99.10.1.2
 channel 1 local SONET0/1/0/0
!
// ssrp part:
ssrp location 0/1/CPU0
 group 1 profile TEST
!
ssrp profile TEST
 peer ipv4 address 99.10.1.2
!
// redundancy group manager part:
redundancy
 multi-router aps
  group 1
   controller SONET0/1/0/0
    member ipv4 99.30.1.2 backup-interface GigabitEthernet0/6/0/4
   !

// ospf part:
router ospf 1
 nsr
 nsf ietf
 redistribute connected instance IPCP
 redistribute static
 area 0
  interface GigabitEthernet0/6/0/4
  !
 !
!

show redundancy-group multi-router aps
```

# SSRP on Serial and Multilink Interfaces Configuration: Example

The following example shows how to configure SSRP on serial interfaces with PPP encapsulation and multilink interfaces:

```
config
 ssrp profile TEST
  peer ipv4 address 99.10.1.2
```

```
!
 ssrp location 0/1/CPU0
  group 1 profile TEST
!
 interface Serial0/1/0/0/1/1:0
  ssrp group 1 id 1 ppp
  encapsulation ppp
  multilink
  group 1
!
 keepalive disable
!
 interface Serial0/1/0/0/1/2:0
  ssrp group 1 id 2 ppp
  encapsulation ppp
  multilink
  group 1
!
 keepalive disable
!
 interface Multilink0/1/0/0/1
  ipv4 address 51.1.1.1 255.255.255.0
  ssrp group 1 id 3 ppp
  encapsulation ppp
  shutdown
!
keepalive disable
!
 controller MgmtMultilink0/1/0/0
  bundle 1
```

**Note** For more information on configuring serial interfaces, refer to the Configuring Serial Interfaces module of this document.

For more information on configuring Multilink, refer to Configuring Multilink PPP.

# VRF on Multilink Configuration for Use with ICSSO: Example

The following example shows how to configure VPN Routing and Forwarding (VRF) on a Multilink interface for use with ICSSO:

```
config
 vrf EvDO-vrf
  address-family ipv4 unicast
!
 interface Multilink 0/0/0/0/1
  description To EvDO BTS Number 1
  vrf EvDO-vrf
  ipv4 address 150.0.1.3 255.255.255.0
  encapsulation ppp
!
```

**Note** For more information on configuring VRF, refer to the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*. For more information on configuring Multilink, refer to Configuring Multilink PPP.

# VRF on Ethernet Configuration for Use with ICSSO: Example

The following example shows how to configure VPN Routing and Forwarding (VRF) on an Ethernet interface for use with ICSSO:

```
config
 vrf EvDO-vrf
  address-family ipv4 unicast
!
 interface GigabitEthernet 1/0/0/0.20
  description Inter-ASR9000 EvDO VLAN
  vrf EvDO-vrf
  encapsulation dot1q 20
```

**Note** For more information on configuring VRF, refer to the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*. For more information on configuring Ethernet, refer to the Configuring Ethernet OAM module of this document.

# OSPF Configuration for Use with ICSSO: Example

Aggregation routers that terminate PPP sessions to a set of cell sites, advertise their availability to LAN switches using Open Shortest Path First (OSPF). The following example shows how to configure OSPF for use with ICSSO:

```
config
 router ospf 1
  nsr
  nsf ietf
  redistribute connected instance IPCP
  redistribute static
  area 0
 interface GigabitEthernet 0/6/0/1
!
```

**Note** For more information on configuring OSPF, refer to the *Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*.

# Verifying ICSSO Configuration: Examples

The following examples show how to verify ICSSO configuration:

## Verifying SSRP Groups: Example

The following example shows how to verify SSRP Group configuration:

```
RP/0/RSP0/CPU0:Router# show ssrp groups all det loc 0/1/cpu0

Tue Nov 10 16:57:55.911 UTC

Group ID: 1
  Conn (ACT,SB): UP,UP
  Profile:  TEST
  Peer:     99.10.1.2
  Max-hops: 255
  Sessions: 3
  Channels Created
  Client:            PPP
    Active Init:      TRUE
    Standby Init:     TRUE
    Active State:     IDT-End-Sent
    Standby State:    IDT-End-Received
    Auth-Req Pending: FALSE
    Active   ID Out:              93
    Active   ID In:               93
    Active   Last Reply In:       93
    Active   Counter:              5

    Standby  ID Out:              50
    Standby  ID In:               50
    Standby  Last Reply In:       50
    Standby  Counter:              5

    Session   Interface
    ----------------------------
    1         Se0/1/0/0/1/1:0
    2         Se0/1/0/0/1/2:0
    3         Mu0/1/0/0/1
```

## Verifying ICSSO Status: Example

The following example shows how to verify ICSSO status:

```
RP/0/RSP0/CPU0:Router# show ppp sso sum loc 0/1/cpu0
Tue Nov 10 16:59:00.253 UTC

Not-Ready    : The session is not yet ready to run as Active or Standby
Stby-UnNegd  : In Standby mode, no replication state received yet
Act-Down     : In Active mode, lower layer not yet up
Deactivating : Session was Active, now going Standby
Act-UnNegd   : In Active mode, not fully negotiated yet
Stby-Negd    : In Standby mode, replication state received and pre-programmed
Activating   : Session was Standby and pre-programmed, now going Active
Act-Negd     : In Active mode, fully negotiated and up
-            : This layer not running

                       Not-   Stby-  Act- Deactiv- Act-   Stby-  Activ- Act-
Layer        | Total   Ready  UnNegd Down   ating  UnNegd Negd   ating  Negd
```

```
-------------+------- ------ ------ ------ ------ ------ ------ ------ ------
LCP          |      6      0      0      0      0      0      0      0      6
of-us-auth   |      6      0      0      0      0      0      0      0      6
of-peer-auth |      6      0      0      0      0      0      0      0      6
IPCP         |      2      0      0      0      0      0      0      0      2
```

## Verifying MR-APS Configuration: Example

The following examples show how to verify MR-APS configuration:

### Example 1:

```
RP/0/RSP0/CPU0:Router# show redundancy-group multi-router aps all

Tue Nov 10 17:00:14.018 UTC

Interchassis Group: 1
             State: FRR ADD SENT
        Controller: SONET0/1/0/0                        0x2000080
  Backup Interface: GigabitEthernet0/6/0/1              0x10000180
  Next Hop IP Addr: 10.10.10.10

Interchassis Group: Not Configured
             State: WAIT CONFIG
        Controller: SONET0/1/0/1                        0x20003c0
  Backup Interface: None                                0x0
  Next Hop IP Addr: 0.0.0.0
```

### Example 2:

```
RP/0/RSP0/CPU0:Router# show cef adj rem loc 0/6/cpu0

Tue Nov 10 17:00:30.471 UTC
Display protocol is ipv4
Interface    Address                                        Type    Refcount

SO0/1/0/0    Ifhandle: 0x2000080                            remote  2
             Adjacency: PT:0xa47c9cf4
             Interface: SO0/1/0/0
             Interface Type: 0x0, Base Flags: 0x110000 (0xa4a00494)
             Nhinfo PT: 0xa4a00494, Idb PT: 0xa4cd60d8, If Handle: 0x2000080
             Ancestor If Handle: 0x0

             Protect FRR: 0xa4a8a040
             Backup FRR: 0xa4a89f34
             Backup NH: 0xa4a00a74
             Backup IFH: 0x10000180
             Backup Interface: Gi0/6/0/1
             Backup IP: 10.10.10.10

             FRR Active: 0
```

## Verifying OSPF Configuration: Example

The following examples show how to verify OSPF configuration:

**Example 1:**

```
RP/0/RSP0/CPU0:Router# show route back
Tue Nov 10 17:01:48.974 UTC

Codes: C - connected, S - static, R - RIP, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR
       A - access/subscriber

C    51.1.1.2/32 is directly connected, 00:10:03, Multilink0/1/0/0/1
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
C    52.1.1.2/32 is directly connected, 00:11:47, Multilink0/1/0/0/2
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
S    110.0.0.2/32 [1/0] via 51.1.1.2, 00:11:40
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
```

**Example 2:**

```
RP/0/RSP0/CPU0:Router# show route 51.1.1.2
Tue Nov 10 17:02:26.507 UTC

Routing entry for 51.1.1.2/32
  Known via "connected IPCP", distance 0, metric 0 (connected)
  Installed Nov 10 16:51:45.703 for 00:10:40
  Routing Descriptor Blocks
    51.1.1.2 directly connected, via Multilink0/1/0/0/1
      Route metric is 0
  No advertising protos.
```

# Verifying Multilink PPP Configurations

Use the following show commands to verify and troubleshoot your multilink configurations: