Configuring PPP on Cisco IOS XR Software

This module describes how to perform the following Point-to-Point Protocol (PPP) related tasks on POS and serial interfaces in Cisco IOS XR software:

- Enable and configure PPP authentication protocols
- Disable PPP authentication
- Modify optional PPP timeout and retry parameters
- Configure Multilink PPP (MLPPP)

Feature History for Configuring PPP Interfaces

Release	Modification
Release 2.0	PPP authentication was introduced on the Cisco CRS-1 router.
Release 3.0	No modification.
Release 3.3.0	Support for serial interfaces with PPP encapsulation was introduced on the Cisco XR 12000 Series router.
Release 3.4.0	No modification.
Release 3.4.1	Support for Multilink PPP was introduced on the Cisco XR 12000 Series router.
Release 3.5.0	No modification.
Release 3.6.0	No modification.
Release 3.7.0	No modification.
Release 3.8.0	No modification.

Contents

Prerequisites for Configuring PPP Authentication

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

- To perform these configuration tasks, your Cisco IOS XR software system administrator must assign you to a user group associated with a task group that includes the corresponding command task IDs. All command task IDs are listed in individual command references and in the *Cisco IOS XR Task ID Reference Guide*.

  If you need assistance with your task group assignment, contact your system administrator. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of *Cisco IOS XR Software System Security Configuration Guide*.
- 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 on Cisco IOS XR Software* module in this manual.
  - To enable PPP encapsulation on a serial interface, see the *Configuring Serial Interfaces on Cisco IOS XR Software* module in this manual.

Information About 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:

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.

How to Configure 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.

Prerequisites

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 on Cisco IOS XR Software* module in this manual.
- To enable PPP encapsulation on an interface, see the *Configuring Serial Interfaces on Cisco IOS XR Software* 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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**] **Example:** `RP/0/RP0/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. • Replace the *protocol* argument with **pap**, **chap**, **or ms-chap**. • 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*. • 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*.

	Command or Action	Purpose

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

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*} **Example:** `RP/0/RP0/CPU0:router# show ppp interfaces serial 0/2/0/0`	Displays PPP state information for an interface. • Enter the *type interface-path-id* argument to display PPP information for a specific interface. • 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. • Enter the **all** keyword to display detailed PPP information for all nodes installed in the router. • Enter the **location** *node-id* keyword argument to display detailed PPP information for the designated node. There are seven possible PPP states applicable for either the Link Control Protocol (LCP) or the Network Control Protocol (NCP).

Where To Go 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 319.
- If you enabled CHAP on an interface, configure a CHAP authentication password, as described in the "Configuring a CHAP Authentication Password" section on page 321
- 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 323

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.

Prerequisites

You must enable PAP authentication on the interface with the **ppp authentication** command, as described in the "Enabling PAP, CHAP, and MS-CHAP Authentication" section on page 316.

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`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/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. • Replace the *username* argument with the username sent in the PAP authentication request. • Enter **password clear** to select cleartext encryption for the password, or enter **password encrypted** if the password is already encrypted. • 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. • You must configure the **ppp pap sent-username** command for each interface. • Remote PAP support is disabled by default.

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

Step 5	**show running-config** **Example:** `RP/0/RP0/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.

Prerequisites

You must enable CHAP authentication on the interface with the **ppp authentication** command, as described in the "Enabling PAP, CHAP, and MS-CHAP Authentication" section on page 316.

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`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp chap password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp chap password clear xxxx`	Enables CHAP authentication on the specified interface, and defines an interface-specific CHAP password. • Enter **clear** to select cleartext encryption, or **encrypted** if the password is already encrypted. • Replace the *password* argument with a cleartext or already-encrypted password. This password is used to authenticate secure communications among a collection of routers. • 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). • The CHAP secret password is used by the routers in response to challenges from an unknown peer.

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

Step 5	**show running-config** **Example:** `RP/0/RP0/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.

Prerequisites

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" section on page 316.

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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp ms-chap password** [**clear** | **encrypted**] *password* **Example:** `RP/0/RP0/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. The MS-CHAP secret password is used by the routers in response to challenges from an unknown peer.

	Command or Action	Purpose

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

Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

How to Modify 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).

Prerequisites

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 on Cisco IOS XR Software* module in this manual.
- To enable PPP encapsulation on an interface, see the *Configuring Serial Interfaces on Cisco IOS XR Software* 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`
	
	Example:
	`RP/0/RP0/CPU0:router# configure`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp max-bad-auth** *retries* **Example:** `RP/0/RP0/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. • If you do not specify the number of authentication retries allowed, the router resets itself immediately after an authentication failure. • Replace the *retries* argument with number of retries after which the interface is to reset itself, in the range from 0 through 10. • The default is 0 retries. • The **ppp max-bad-auth** command applies to any interface on which PPP encapsulation is enabled.
Step 4	**ppp max-configure** *retries* **Example:** `RP/0/RP0/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. • Replace the *retries* argument with the maximum number of configure requests retries, in the range from 4 through 20. • The default maximum number of configure requests is 10. • If a configure request message receives a reply before the maximum number of configure requests are sent, further configure requests are abandoned.
Step 5	**ppp max-failure** *retries* **Example:** `RP/0/RP0/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. • Replace the *retries* argument with the maximum number of CONFNAKs to permit before terminating a negotiation, in the range from 2 through 10. • The default maximum number of CONFNAKs is 5.
Step 6	**ppp max-terminate** *number* **Example:** `RP/0/RP0/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. • 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. • The default maximum number of TermReqs is 2.

	Command or Action	Purpose
Step 7	**ppp timeout authentication** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp timeout authentication 20`	(Optional) Sets PPP authentication timeout parameters. • 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. • 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* **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp timeout retry 8`	(Optional) Sets PPP timeout retry parameters. • 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. • The default is 3 seconds.

Step 9	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 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*} **Example:** `RP/0/RP0/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.

How to Disable 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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp pap refuse** **Example:** `RP/0/RP0/CPU0:router(config-if)# ppp pap refuse`	Refuses Password Authentication Protocol (PAP) authentication from peers requesting it. • 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. • PAP authentication is disabled by default.

	Command or Action	Purpose

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

Step 5	**show running-config** **Example:** `RP/0/RP0/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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp chap refuse** **Example:** `RP/0/RP0/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. • CHAP authentication is disabled by default. • 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
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 5	**show running-config** **Example:** `RP/0/RP0/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** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface serial 0/4/0/1`	Enters interface configuration mode.
Step 3	**ppp ms-chap refuse** **Example:** `RP/0/RP0/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. • MS-CHAP authentication is disabled by default. • 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.

	Command or Action	Purpose

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

Step 5	**show running-config** **Example:** `RP/0/RP0/CPU0:router# show running-config`	Verifies PPP authentication information for interfaces that have PPP encapsulation enabled.

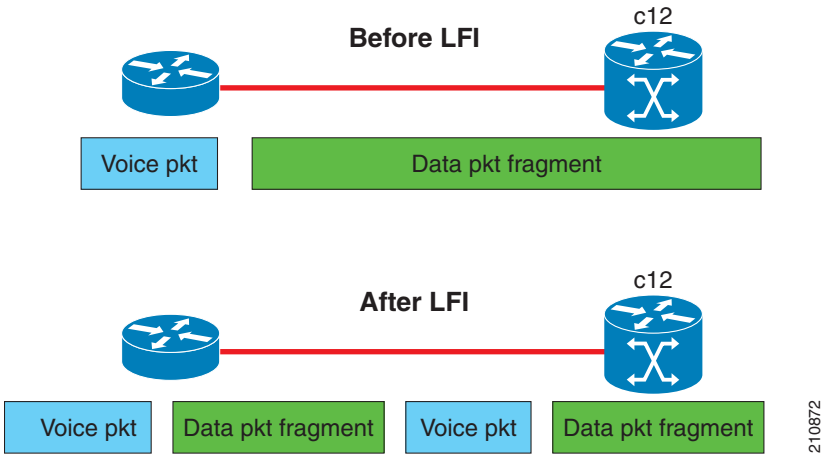
Information About Multilink PPP

Multilink Point-to-Point Protocol (MLPPP) provides a method for combining multiple physical links into one logical link. The implementation on Cisco IOS XR on Cisco XR 12000 Series routers 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 that run at less than 768 Kbps.

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 that run at less than 768 Kbps. LFI reduces delay and jitter by fragmenting large datagrams and interleaving them with low-delay traffic packets.

Figure 8 Link Fragmentation Interleave



Supported Cards

MLPPP is supported on the following line cards and SPAs:

- Cisco XR 12000 multiservice line cards
- 2-Port and 4-Port Channelized T3 SPAs (SPA-2XCT3/DS0, SPA-4XCT3/DS0)

LFI is supported on:

- Cisco 1-Port Channelized STM-1/OC-3 Shared Port Adapter

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:

- Fragment sizes of 128, 256, and 512 bytes
- Long sequence numbers (24-bit)
- Lost fragment detection timeout period of 80 ms
- Minimum-active-links configuration option
- LCP echo request/reply support over multilink interface
- Full T1 and E1 framed and unframed links

Limitations

MLPPP for Cisco IOS XR software has the following limitations:

- 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.

- Maximum of 12 links per bundle.
- Maximum of 28 bundles on the 2-Port Channelized T3 SPA.
- Maximum of 56 bundles on the 4-Port Channelized T3 SPA.
- Maximum of 224 bundles per line card.
- 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.

Note If you change the MTU value on an interface that is configured for PPP encapsulation, the line protocol will flap.

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.

How to Configure Multilink PPP

This section contains the following procedures:

- Configuring the Controller, page 334
- Configuring the Interfaces, page 337
- Configuring MLPPP Optional Features, page 339
- Removing an MLPPP member, page 341

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** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller** *type interface-path-id* **Example:** `RP/0/0/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* **Example:** `RP/0/0/CPU0:router# mode t1`	Configures the type of multilinks to channelize; for example, 28 T1s.
Step 4	**clock source** {**internal** | **line**} **Example:** `RP/0/0/CPU0:router(config-t3)# clock source internal`	(Optional) Configures the clocking for the port. **Note** The default clock source is **internal**.
Step 5	**exit** **Example:** `RP/0/0/CPU0:router(config-t3)# exit`	Exits controller configuration mode.
Step 6	**controller t1** *interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# controller t1 0/1/0/0/0`	Enters T1 configuration mode.
Step 7	**channel-group** *channel-group-number* **Example:** `RP/0/0/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.

	Command or Action	Purpose
Step 8	**timeslots** *range* **Example:** `RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 7-12`	Associates one or more DS0 time slots to a channel group and creates an associated serial subinterface on that channel group. • Range is from 1 to 24 time slots. • You can assign all 24 time slots to a single channel group, or you can divide the time slots among several channel groups. **Note** The time slot range must be from 1 to 24 for the resulting serial interface to be accepted into a MLPPP bundle.
Step 9	**exit** **Example:** `RP/0/0/CPU0:router(config-t1-channel_group)# exit`	Exits channel group configuration mode.
Step 10	**exit** **Example:** RP/0/0/CPU0:router(config-t1)# exit	Exits T1 configuration mode and enters global configuration mode.
Step 11	**controller mgmtmultilink** *interface-path-id* **Example:** `RP/0/0/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.

	Command or Action	Purpose
Step 12	**bundle** *bundle-id* **Example:** `RP/0/0/CPU0:router(config-mgmtmultilink)# bundle 20`	Creates a multilink interface with the specified bundle ID.

Step 13	**end**
	or
	commit
	
	Example:
	`RP/0/0/CPU0:router(config-t3)# end`
	or
	`RP/0/0/CPU0:router(config-t3)# 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 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.

Note If you change the MTU value on an interface that is configured for PPP encapsulation, the line protocol will flap.

SUMMARY STEPS

1. **configure**
2. **interface multilink** *interface-path-id*
3. **ipv4 address** *address/mask*
4. **multilink fragment-size** *size*
5. **keepalive** {*interval* | **disable**}
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** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface multilink** *interface-path-id* **Example:** `RP/0/0/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* **Example:** `RP/0/0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24`	Assigns an IP address and subnet mask to the interface in the format: *A.B.C.D/prefix* or *A.B.C.D/mask*
Step 4	**multilink fragment-size** *size* **Example:** `RP/0/0/CPU0:router(config-if)# multilink fragment-size 128`	(Optional) Specifies the size of the multilink fragments, such as 128 bytes. Some fragment sizes may not be supported. The default is no fragments.

Step 5	**keepalive** {*seconds* | **disable**}
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# keepalive 3`
	or
	`RP/0/RP0/CPU0:router(config-if)# keepalive disable`
		Specifies the frequency (in seconds) at which the Link Control Protocol (LCP) sends ECHOREQ packets to its peer. The default keepalive interval is 10 seconds.
		To restore the system to the default keepalive interval, use the **no keepalive** command.
		To disable the keepalive timer, use the **keepalive disable** command.

	Command or Action	Purpose
Step 6	**exit** **Example:** `RP/0/0/CPU0:router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**interface** *type interface-path-id* **Example:** `RP/0/0/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* **Example:** `RP/0/0/CPU0:router(config-if)# encapsulation ppp`	Specifies the type of encapsulation; in this case, PPP. **Note** PPP is supported only in Cisco IOS XR Release 3.4.1 and later releases.
Step 9	**multilink group** *group-id* **Example:** `RP/0/0/CPU0:router(config-if)# multilink group 1`	Specifies the multilink group ID for this interface.

Step 10	**end**
	or
	commit
	
	Example:
	`RP/0/0/CPU0:router(config-t3)# end`
	or
	`RP/0/0/CPU0:router(config-t3)# 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 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** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface multilink** *interface-path-id* **Example:** `RP/0/0/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** **Example:** `RP/0/0/CPU0:router(config-if)# multilink`	Enters interface multilink configuration mode.
Step 4	**ppp multilink minimum-active links** *value* **Example:** `RP/0/0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 12`	(Optional) Specifies the minimum number of active links for the multilink interface.
Step 5	**multilink interleave** **Example:** `RP/0/0/CPU0:router(config-if-multilink)# multilink interleave`	(Optional) Enables interleave on a multilink interface.

	Command or Action	Purpose
Step 6	**no shutdown** **Example:** `RP/0/0/CPU0:router(config-if-mutlilink)# no shutdown`	Removes the shutdown configuration. • 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
	
	Example:
	`RP/0/0/CPU0:router(config-t3)# end`
	or
	`RP/0/0/CPU0:router(config-t3)# 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.

Removing an MLPPP member

Perform this task to remove an MLPPP member link.

SUMMARY STEPS

1. **configure**
2. **controller** *type interface-path-id*
3. **shutdown**
4. **exit**
5. **interface type** *interface-path-id*
6. **no multilink group** *group-id*
7. **encapsulation** *type*
8. **end**
   or
   commit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**controller** *type interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# controller t1 0/4/2/0/11`	Enters controller configuration submode and specifies the controller name and instance identifier in *rack/slot/module/port* notation.
Step 3	**shutdown** **Example:** `RP/0/0/CPU0:router(config-t1)#shutdown`	Exits T1 controller configuration mode.
Step 4	**exit** **Example:** `RP/0/0/CPU0:router(config-t1)#exit`	Exits T1 configuration mode.
Step 5	**interface** *type interface-path-id* **Example:** `RP/0/0/CPU0:router(config)#interface serial 0/4/3/11:0`	Enters interface configuration mode.
Step 6	**no multilink group** *group-id* **Example:** `RP/0/0/CPU0:router(config-if)#no multilink group 111`	Removes the multilink group for this interface.

	Command or Action	Purpose
Step 7	**encapsulation** *type* **Example:** `RP/0/0/CPU0:router(config-if)#no encapsulation ppp`	Specifies the type of encapsulation; in this case, PPP.

Step 8	**end**
	or
	commit
	
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for PPP

This section provides the following configuration examples:

- Configuring a POS Interface with PPP Encapsulation: Example, page 343
- Configuring a Serial Interface with PPP Encapsulation: Example, page 344
- Verifying Multilink PPP Configurations, page 345

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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# ppp pap sent-username P1_CRS-8 password xxxx
RP/0/RP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access
RP/0/RP0/CPU0:router(config-if)# ppp chap password encrypted xxxx
RP/0/RP0/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/RP0/CPU0:router# configuration
RP/0/RP0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/RP0/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/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# ppp authentication ms-chap MIS-access
RP/0/RP0/CPU0:router(config-if)# ppp ms-chap password encrypted xxxx
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

Configuring MLPPP: Example

```
RP/0/0/CPU0:router# configure
RP/0/0/CPU0:router(config)# controller t3 0/1/0/0
RP/0/0/CPU0:router# mode t1
RP/0/0/CPU0:router(config-t3)# clock source internal
RP/0/0/CPU0:router(config-t3)# exit
RP/0/0/CPU0:router(config)# controller t1 0/1/0/0/0
RP/0/0/CPU0:router(config-t1)# channel-group 0
RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 7-12
RP/0/0/CPU0:router(config-t1-channel_group)# exit
RP/0/0/CPU0:router(config-t1)# exit
RP/0/0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0
RP/0/0/CPU0:router(config-mgmtmultilink)# bundle 20
RP/0/0/CPU0:router(config-t3)# commit
RP/0/0/CPU0:router(config-t3)# exit

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

RP/0/0/CPU0:router(config)# interface multilink 0/1/0/0/1
RP/0/0/CPU0:router(config-if)# multilink
RP/0/0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 12
RP/0/0/CPU0:router(config-if-multilink)# multilink interleave
RP/0/0/CPU0:router(config-if-mutlilink)# no shutdown
RP/0/0/CPU0:router(config-t3)# commit
```

Verifying Multilink PPP Configurations

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

- show multilink interfaces: Example, page 345
- show ppp interfaces multilink: Example, page 346
- show ppp interface serial: Example, page 346
- show imds interface multilink: Example, page 346

show multilink interfaces: Example

```
RP/0/0/CPU0:Router# show multilink interfaces multilink 0/3/1/0/301

Multilink0/3/1/0/301 is up, line protocol is up

     Fragmentation: disabled
     Member Links: 2 active, 0 inactive
      - Serial0/3/1/0/0:0: ACTIVE
      - Serial0/3/1/0/1:0: ACTIVE

RRP/0/0/CPU0:Router# show multilink interfaces

Multilink0/3/1/0/301 is up, line protocol is up

     Fragmentation: disabled
     Member Links: 2 active, 0 inactive
      - Serial0/3/1/0/0:0: ACTIVE
      - Serial0/3/1/0/1:0: ACTIVE

Multilink0/3/1/0/302 is up, line protocol is up

     Fragmentation: disabled
     Member Links: 2 active, 0 inactive
      - Serial0/3/1/1/1:0: ACTIVE
      - Serial0/3/1/1/0:0: ACTIVE

Serial0/3/1/0/0:0 is up, line protocol is up
     Multilink group id: 301
     Member status: ACTIVE

Serial0/3/1/1/0:0 is up, line protocol is up
     Multilink group id: 302
     Member status: ACTIVE

Serial0/3/1/0/1:0 is up, line protocol is up
     Multilink group id: 301
     Member status: ACTIVE

Serial0/3/1/1/1:0 is up, line protocol is up
     Multilink group id: 302
     Member status: ACTIVE
```

show ppp interfaces multilink: Example

```
RP/0/0/CPU0:Router# show ppp interfaces multilink 0/3/1/0/1

Multilink 0/3/1/0/1 is up, line protocol is up
LCP: Open
     Keepalives disabled
  IPCP: Open
     Local IPv4 address: 1.1.1.2
     Peer IPv4 address:  1.1.1.1
  Multilink
     Member Links: 2 active, 1 inactive (min-active 1)
      - Serial0/3/1/0/0:0: ACTIVE
      - Serial0/3/1/0/1:0: ACTIVE
      - Serial0/3/1/0/2:0: INACTIVE : LCP has not been negotiated
```

show ppp interface serial: Example

```
RP/0/0/CPU0:Router# show ppp interface Serial 0/3/1/0/0:0

Serial 0/3/1/0/0:0 is up, line protocol is up
  LCP: Open
     Keepalives disabled
     Local MRU: 1500 bytes
     Peer  MRU: 1500 bytes
     Local Bundle MRRU: 1596 bytes
     Peer  Bundle MRRU: 1500 bytes
     Local Endpoint Discriminator: 1b61950e3e9ce8172c8289df0000003900000001
     Peer  Endpoint Discriminator: 7d046cd8390a4519087aefb90000003900000001
  Authentication
     Of Peer: <None>
     Of Us:   <None>
  Multilink
     Multilink group id: 1
     Member status: ACTIVE
```

show imds interface multilink: Example

```
RP/0/0/CPU0:Router# show imds interface Multilink 0/3/1/0/1

IMDS INTERFACE DATA (Node 0x0)

Multilink0_3_1_0_1 (0x04001200)
-----------------------
flags: 0x0001002f    type: 55 (IFT_MULTILINK)    encap: 52 (ppp)
state: 3 (up)     mtu: 1600    protocol count: 3
control parent: 0x04000800     data parent: 0x00000000
       protocol         capsulation           state        mtu
    --------------- ------------------- --------------- --------
    12 (ipv4)
                    26 (ipv4)            3 (up)          1500
                    47 (ipcp)            3 (up)          1500
    16 (ppp_ctrl)
                    53 (ppp_ctrl)        3 (up)          1500
    0 (Unknown)
                    139 (c_shim)         3 (up)          1600
                    52 (ppp)             3 (up)          1504
                    56 (queue_fifo)      3 (up)          1600
                    60 (txm_nopull)      3 (up)          1600
```

Additional References

The following sections provide references related to PPP encapsulation.

Related Documents

Related Topic	Document Title
Cisco IOS XR master command reference	*Cisco IOS XR Master Commands List*
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*
Initial system bootup and configuration information for a router using Cisco IOS XR software	*Cisco IOS XR Getting Started Guide*
Cisco IOS XR AAA services configuration information	*Cisco IOS XR System Security Configuration Guide* and *Cisco IOS XR System Security Command Reference*
Information about configuring interfaces and other components on the Cisco CRS-1 router from a remote Craft Works Interface (CWI) client management application	*Cisco Craft Works Interface Configuration Guide*

Standards

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

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature	To locate and download MIBs for selected platforms using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC-1661	*The Point-to-Point Protocol (PPP)*
RFC- 1994	*PPP Challenge Handshake Authentication Protocol (CHAP)*

Technical Assistance

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