Using Multilink PPP over Serial Interface Links

This module tells you how to use Multilink PPP over serial interface links.

Note As of Cisco IOS XE Software Release 2.1, Multilink PPP over serial interface links is the only Multilink PPP type supported. Multiclass MLPPP is not supported.

- Finding Feature Information, page 1
- Prerequisites for Using Multilink PPP over Serial Interface Links, page 2
- Restrictions for Using Multilink PPP over Serial Interface Links, page 2
- Information About Using Multilink PPP over Serial Interface Links, page 2
- How to Configure Multilink PPP over Serial Interface Links, page 2
- Configuration Examples for Using Multilink PPP over Serial Interface Links, page 7
- Additional References, page 9
- Feature Information for Using Multilink PPP over Serial Interface Links, page 10

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Using Multilink PPP over Serial Interface Links

Be familiar with the concepts in the "Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP" module.

Enable a queueing mechanism such as low latency queueing (LLQ), weighted fair queueing (WFQ), class-based WFQ (CBWFQ) and Weighted Random Early Detection (WRED), as applicable, before configuring multilink.

Restrictions for Using Multilink PPP over Serial Interface Links

If a multilink bundle has one link or packet order is not important for interleaved packets, use Link Fragmentation and Interleaving (LFI) without multiclass. Use LFI with multiclass if a multilink bundle has multiple links.

Only Voice over IP (VoIP) is supported.

As of Cisco IOS XE Release 2.1, Multilink PPP over serial interface links is the only Multilink PPP type supported. Multiclass MLPPP is not supported.

Information About Using Multilink PPP over Serial Interface Links

MQC and Multilink PPP over Serial Interface Links

Before using Multilink PPP over serial interface links, a traffic policy (policy map) must be created. Policy maps are created using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

How to Configure Multilink PPP over Serial Interface Links

Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface

Before You Begin

Before proceeding with this task, you must create a policy map. The policy map contains the configuration parameters used to apply the specific quality of service feature to the network traffic. To create a policy map, use the MQC.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *multilink-bundle-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **service-policy output** *policy-map-name*
6. **service-policy input** *policy-map-name*
7. **ppp multilink fragment delay** *milliseconds* [*microseconds*]
8. **ppp multilink interleave**
9. **ppp multilink multiclass**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *multilink-bundle-number* **Example:** `Router(config)# interface multilink 1`	Creates a multilink bundle and enters interface configuration mode. • Enter the multilink bundle number.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.10.100.1 255.255.255.0`	Sets a primary IP address for an interface. This command can also set the optional secondary IP address for an interface. • Enter the primary IP address (and, optionally, the secondary IP address).
Step 5	**service-policy output** *policy-map-name* **Example:** `Router(config-if)# service-policy output policy1`	Attaches the previously created QoS traffic policy (policy map). The policy map evaluates and applies QoS features for traffic *leaving* the interface. • Enter the policy map name.

	Command or Action	Purpose
Step 6	**service-policy input** *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches the previously created QoS traffic policy (policy map). The policy map evaluates and applies QoS features for traffic *entering*the interface. • Enter the policy map name.
Step 7	**ppp multilink fragment delay** *milliseconds* [*microseconds*] **Example:** `Router(config-if)# ppp multilink fragment delay 20`	Specifies a maximum size in units of time for packet fragments on a Multilink PPP (MLP) bundle. • Enter the maximum amount of time, in milliseconds.
Step 8	**ppp multilink interleave** **Example:** `Router(config-if)# ppp multilink interleave`	Enables interleaving of packets among the fragments of larger packets on a multilink bundle.
Step 9	**ppp multilink multiclass** **Example:** `Router(config-if)# ppp multilink multiclass`	(Optional) Enables Multiclass Multilink PPP (MCMP) on an interface. **Note** Use this command only if there are multiple links in the multilink bundle.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Associating the Serial Interface with the Multilink Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot* / *port* : *timeslot*
4. **no fair-queue**
5. **encapsulation ppp**
6. **ppp multilink**
7. **ppp multilink group** *group-number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot* / *port* : *timeslot* **Example:** `Router# interface serial 4/1:23` **Example:**	Specifies a serial interface created on a channelized E1 or channelized T1 controller (for ISDN PRI, channel-associated signaling, or robbed-bit signaling), and enters interface configuration mode. • Enter the slot number and port number where the channelized E1 or T1 controller is located.
Step 4	**no fair-queue** **Example:** `Router(config-if)# no fair-queue`	Disables WFQ (or DWFQ for VIP-enabled routers).
Step 5	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Sets the serial interface encapsulation method used by the interface.

	Command or Action	Purpose
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables Multilink on an interface.
Step 7	**ppp multilink group** *group-number* **Example:** `Router(config-if)# ppp multilink group 1`	Restricts a physical link to joining only a designated multilink group interface. • Enter the multilink group number.
Step 8	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Verifying the Multilink PPP over Serial Interface Link Configuration

SUMMARY STEPS

1. **enable**
2. **show interfaces** [*type number*] [*first*] [*last*] [**accounting**]
3. **show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]]
4. **show policy-map interface** *interface-name* [**vc** [*vpi/*] *vci*] [**dlci** *dlci*] [**input** | **output**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interfaces** [*type number*] [*first*] [*last*] [**accounting**] **Example:** `Router# show interfaces`	(Optional) Displays statistics for all interfaces configured on the router or access server.

	Command or Action	Purpose
Step 3	**show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]] **Example:** `Router# show ppp multilink`	(Optional) Displays bundle information for mutlilink bundles.
Step 4	**show policy-map interface** *interface-name* [**vc** [*vpi/*] *vci*] [**dlci** *dlci*] [**input** | **output**] **Example:** `Router# show policy-map interface serial0/0/0`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific permanent virtual circuit (PVC) on the interface.
Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Using Multilink PPP over Serial Interface Links

Configuring Multilink PPP over Serial Interface Links on a Multilink Group Interface Example

The following is an example of configuring Multilink PPP over serial interface links on a multilink group interface:

```
Router> enable
Router# configure terminal
Router(config)# interface multilink 1
Router(config-if)# ip address 10.10.100.1 255.255.255.0

Router(config-if)# service-policy output policy1
Router(config-if)# service-policy input policy1
Router(config-if)# ppp multilink fragment delay 20
```

```
Router(config-if)# ppp multilink interleave

Router(config-if)# ppp multilink multiclass
Router(config-if)# end
```

Associating the Serial Interface with the Multilink Group Example

The following is an example of associating the serial interface serial4/1 with the multilink group:

```
Router> enable
Router# configure terminal
Router(config)# interface serial 4/1:23
Router(config-if)# no fair-queue
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink group 1
Router(config-if)# end
```

Example Verifying the Multilink PPP over Serial Interface Link Configuration

You can verify the Multilink PPP over serial interface links configuration by using one or more of the following **show** commands:

- **show interfaces**
- **show ppp multilink**
- **show policy-map interface**

The following section provides sample output of the **showpppmultilink** command only. For sample output of the other commands, see the *Cisco IOS Quality of Service Solutions Command Reference*.

show ppp multilink Command Output Example

The following is an example of the **showpppmultilink** command output. In this example, one multilink bundle called bundle-1 is on the system. This bundle has two member links: one active link and one inactive link.

```
Router# show ppp multilink
Multilink2, bundle name is bundle-1
  Endpoint discriminator is bundle-1
  Bundle up for 00:00:09, 1/255 load
  Receive buffer limit 12000 bytes, frag timeout 1500 ms
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 0 reordered
```

```
  0/0 discarded fragments/bytes, 0 lost received
  0x0 received sequence, 0x3 sent sequence
Member links:1 active, 1 inactive (max not set, min not set)
  Se3/2, since 00:00:10, 240 weight, 232 frag size
  Se3/3 (inactive)
```

Additional References

The following sections provide references related to using Multilink PPP over ATM links.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
LLQ, WFQ, CBWFQ, PQ, CQ, FIFO, and other queueing mechanisms	"Configuring Weighted Fair Queueing" module
MQC	"Applying QoS Features Using the MQC" module
Multilink PPP configurations	"Configuring Media-Independent PPP and Multilink PPP" module
Virtual template interfaces	"Configuring Virtual Template Interfaces" module
Multilink PPP overview module	"Reducing Latency and Jitter for Real-Time Traffic Using Multilink PPP" module
Multilink PPP over Frame Relay	"Using Multilink PPP over Frame Relay" module
Multilink PPP over dialer interface links	"Using Multilink PPP over Dialer Interface Links" module
Multilink PPP over serial interface links	"Using Multilink PPP over Serial Interface Links" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1990	*The PPP Multilink Protocol (MP)*
RFC 2686	*Multiclass Extension to Multilink PPP (MCML)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using Multilink PPP over Serial Interface Links

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Using Multilink PPP over Serial Interface Links

Feature Name	Software Releases	Feature Configuration Information
MLPPP QoS (CBWFQ, LLQ, WRED)	Cisco IOS XE Release 2.1	The MLPPP QoS feature implements Multilink PPP (MLPPP) using a distributed hierarchical queueing framework (HQF). The MLPPP QoS feature incorporates class-based weighted fair queueing (CBWFQ), low latency queueing (LLQ), and weighted random early detection (WRED) functionality.