Configuring Media-Independent PPP and Multilink PPP

First Published: May 10, 2001
Last Updated: November 20, 2014

The Configuring Media-Independent PPP and Multilink PPP module describes how to configure PPP and Multilink PPP (MLP) features on any interface. This module also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces.

Multilink PPP provides a method for spreading traffic across multiple physical WAN links.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 for Configuring Media-Independent PPP and Multilink PPP" section on page 50.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Media-Independent PPP and Multilink PPP

Understanding PPP and multilink operations.

Information About Media-Independent PPP and Multilink PPP

To configure the Media-Independent PPP and Multilink PPP, you should understand the following concepts:

- Point-to-Point Protocol, page 2
- CHAP or PPP Authentication, page 2
- Microsoft Point-to-Point Compression, page 3
- IP Address Pooling, page 4

Point-to-Point Protocol

Point-to-Point Protocol (PPP), described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- High-Speed Serial Interface (HSSI)
- ISDN
- Synchronous serial

The implementation of PPP supports authentication using Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), or Password Authentication Protocol (PAP), and the option 4, and option 5, and Magic Number configuration options.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

CHAP or PPP Authentication

PPP with CHAP or PAP authentication is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP were originally specified in RFC 1334, and CHAP was updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a *name*. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the hostname of the local router.

The required response has two parts:

- An encrypted version of the ID, a secret password, and the random number
- Either the hostname of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret password by performing the same encryption operation as indicated in the response and looking up the required hostname or username. The secret passwords must be identical on the remote device and the local router.

Because this response is sent, the password is never sent in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only when a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. The username and password specified in the authentication request are accepted, and the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.

For CHAP, configure hostname authentication and the secret password for each remote system with which authentication is required.

Microsoft Point-to-Point Compression

Microsoft Point-to-Point Compression (MPPC) is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections. The MPPC algorithm uses a Lempel-Ziv (LZ)-based algorithm with a continuous history buffer called a dictionary.

The Compression Control Protocol (CCP) configuration option for MPPC is 18.

Exactly one MPPC datagram is encapsulated in the PPP information field. The PPP protocol field indicates the hexadecimal type of 00FD for all compressed datagrams. The maximum length of the MPPC datagram sent over PPP is the same as the MTU of the PPP interface; however, this length cannot be greater than 8192 bytes because the history buffer is limited to 8192 bytes. If compressing the data results in data expansion, the original data is sent as an uncompressed MPPC packet.

The history buffers between compressor and decompressor are synchronized by maintaining a 12-bit coherency count. If the decompressor detects that the coherency count is out of sequence, the following error recovery process is performed:

1. The Reset Request (RR) packet is sent from the decompressor.
2. The compressor then flushes the history buffer and sets the flushed bit in the next packet it sends.
3. Upon receiving the flushed bit set packet, the decompressor flushes the history buffer.

Synchronization is achieved without CCP using the Reset Acknowledge (RA) packet, which can consume additional time.

Compression negotiation between a router and a Windows 95 client occurs through the following process:

1. Windows 95 sends a request for both STAC (option 17) and MPPC (option 18) compression.
2. The router sends a negative acknowledgment (NAK) requesting only MPPC.
3. Windows 95 resends the request for MPPC.

The router sends an acknowledgment (ACK) confirming MPPC compression negotiation.

IP Address Pooling

A point-to-point interface must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command, provided by TACACS+ or the Dynamic Host Configuration Protocol (DHCP), or from a locally administered pool.

IP address pooling uses a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through IPCP address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

The IP address pooling feature allows configuration of a global default address pooling mechanism, per-interface configuration of the address pooling mechanism, and per-interface configuration of a specific address or pool name.

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP or Serial Line Internet Protocol (SLIP) EXEC command—An asynchronous dialup user can enter a peer IP address or hostname when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Default IP address.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dialup interface can use. This address overrides any default IP address and prevents pooling from taking place.

- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dialup user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 1024 addresses) stored in two queues. The free queue contains addresses available to be assigned and the used queue contains addresses that are in use. Addresses are stored to the free queue in first-in, first-out (FIFO) order to minimize the chance the address will be reused, and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.
- Chat script (asynchronous serial interfaces only)—The IP address in the **dialer map** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- Virtual terminal/protocol translation—The translate command can define the peer IP address for a virtual terminal (pseudo asynchronous interface).
- The pool configured for the interface is used, unless TACACS+ returns a pool name as part of authentication, authorization, and accounting (AAA). If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

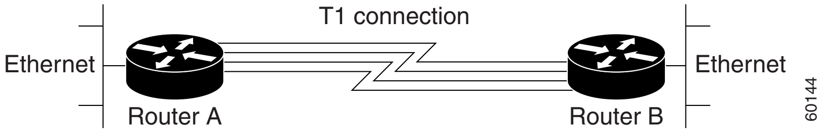
1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP or SLIP command, or from a chat script
5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer-provided address from IPCP negotiation (not accepted unless no other address exists)

MLP on Synchronous Serial Interfaces

Address pooling is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces that are running PPP and PPPoX sessions.

MLP provides characteristics are most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 1 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 1 Inverse Multiplexing Application Using Multilink PPP



How to Configure Media-Independent PPP and Multilink PPP

This section includes the following procedures:

- Configuring PPP and MLP, page 6
- Configuring MLP Interleaving and Queueing, page 28
- Monitoring and Maintaining PPP and MLP Interfaces, page 35

Configuring PPP and MLP

Perform the following task in interface configuration mode to configure PPP on a serial interface (including ISDN). This task is required for PPP encapsulation.

- Enabling PPP Encapsulation, page 6

You can also complete the tasks in the following sections; these tasks are optional but offer a variety of uses and enhancements for PPP on your systems and networks:

- Enabling CHAP or PAP Authentication, page 7
- Configuring Compression of PPP Data, page 9
- Configuring IP Address Pooling
- Disabling or Reenabling Peer Neighbor Routes
- Configuring Multilink PPP
- Configuring Multilink PPP
- Configuring MLP Interleaving
- Creating a Multilink Bundle
- Assigning an Interface to a Multilink Bundle
- Disabling PPP Multilink Fragmentation

See the "Monitoring and Maintaining PPP and MLP Interfaces" section on page 35 for tips on maintaining PPP. See the "Configuration Examples for PPP and MLP" section on page 36 to understand how to implement PPP and MLP in your network.

Enabling PPP Encapsulation

The **encapsulation ppp** command enables PPP on serial lines to encapsulate IP and other network protocol datagrams.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure fastethernet** *number*
4. **encapsulation ppp**
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`interface fastethernet`** *`number`*
	Example:
	`Device(config)# interface fastethernet 0/0`
		Enters interface configuration mode.

Step 4	**`encapsulation ppp`**
	Example:
	`Device(config-if) # encapsulation ppp`
		Enables PPP encapsulation.
		Note PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. Use the **no keepalive command** to disable echo requests.

Step 5	**`end`**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

Enabling CHAP or PAP Authentication

To enable CHAP or PAP authentication, perform the steps mentioned in this section.

Caution If you use a list name that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

For an example of CHAP, see the section "“Examples: CHAP with an Encrypted Password:” section on page 36”. CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP).*

For information about MS-CHAP, see *MS-CHAP Support*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**]
5. **ppp use-tacacs** [**single-line**]

   or

   aaa authentication ppp

6. **exit**
7. **username** *name* [**user-maxlinks** *link-number*] **password** *secret*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *number* **Example:** `Device(config)# interface fastethernet 0/0`	Enters Interface Configuration mode.
Step 4	**ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**] **Example:** `Device(config-if)# ppp authentication chap`	Defines the authentication methods supported and the order in which they are used. **Note** Use the **ppp authentication chap** command only with TACACS or extended TACACS. **Note** With AAA configured on the router and list names defined for AAA, the *list-name* optional argument can be used with AAA/TACACS+. Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

Step 5	**ppp use-tacacs** [**single-line**]
	or
	aaa authentication ppp
	Example:
	`Device(config-if)# ppp use-tacacs single-line`
	or
	`Device(config-if)# aaa authentication ppp`
		Configure TACACS on a specific interface as an alternative to global host authentication.

Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 7	`username` *name* [`user-maxlinks` *link-number*] `password` *secret* **Example:** `Device(config)# username name user-maxlinks 1 password password1`	Configures identification. • Optionally, you can specify the maximum number of connections a user can establish. • To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command and PPP encapsulation and name authentication on all the interfaces the user will be accessing.

Step 8	`end`
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. PPP encapsulations support both predictor and Stacker compression algorithms.

If most of your traffic is already compressed files, do not use compression.

To configure compression of PPP data, perform the steps in this section.

Software Compression

Software compression is available on all router platforms. Software compression is performed by the main processor in the router.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if the router CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **encapsulation PPP**
5. **compress** [**predictor** | **stac** | **mppc** [**ignore-pfc**]]
6. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`interface fastethernet`** *`number`*
	Example:
	`Device(config)# interface fastethernet 0/0`
		Enters interface configuration mode.

Step 4	**`encapsulation ppp`**
	Example:
	`Device(config-if)# encapsulation ppp`
		Enables encapsulation of a single protocol on the serial line.

Step 5	**`compress`** `[`**`predictor`** `|` **`stac`** `|` **`mppc`** `[`**`ignore-pfc`**`]]`
	Example:
	`Device(config-if)# compress predictor`
		Enables compression.

Step 6	**`end`**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode.

Configuring Microsoft Point-to-Point Compression

Perform this task to configure MPCC. This will help you set MPPC once PPP encapsulation is configured on the router.

Prerequisites

Ensure that PPP encapsulation is enabled before you configure MPPC. For information on how to configure PPP encapsulation, see the "Enabling PPP Encapsulation" section on page 6".

Restrictions

The following restrictions apply to the MPPC feature:

- MPPC is supported only with PPP encapsulation.
- Compression can be processor intensive because it requires a reserved block of memory to maintain the history buffer. Do not enable modem or hardware compression because it may cause performance degradation, compression failure, or data expansion.
- Both ends of the point-to-point link must be using the same compression method (STAC, Predictor, or MPPC, for example).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *number*
4. **compress** [**mppc** [**ignore-pfc**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *number* **Example:** `Device(config)# interface serial 2/0`	Enters interface configuration mode.
Step 4	**compress** [**mppc** [**ignore-pfc**]] **Example:** `Device(config-if)# compress mppc`	Enables encapsulation of a single protocol on the serial line. - The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by Link Control Protocol (LCP). For example, the uncompressed standard protocol field value for IP is 0x0021 and 0x21 when compression is enabled. When the **ignore-pfc** option is enabled, the router will continue to use the uncompressed value (0x0021). Using the **ignore-pfc** option is helpful for some asynchronous driver devices that use an uncompressed protocol field (0x0021), even though the protocol field compression is negotiated between peers.

Examples

Following is sample **debug ppp negotiation** command output showing protocol reject:

```
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
```

Configuring IP Address Pooling

You can define the type of IP address pooling mechanism used on router interfaces in one or both of the ways described in the following sections:

- Defining the Global Default Address Pooling Mechanism, page 12
- Configuring IP Address Assignment, page 15

Defining the Global Default Address Pooling Mechanism

The global default mechanism applies to all point-to-point interfaces that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in the following sections:

- Defining DHCP as the Global Default Mechanism, page 12
- Defining Local Address Pooling as the Global Default Mechanism, page 13

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

You can also control the DHCP network discovery mechanism; see the following section for more information:

- Controlling DHCP Network Discovery, page 14

Defining DHCP as the Global Default Mechanism

DHCP specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

Perform this task to enable DHCP as the global default mechanism.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool dhcp-proxy-client**
4. **ip dhcp-server** [*ip-address* | *name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`ip address-pool dhcp-proxy-client`**
	Example:
	`Device(config)# ip address-pool dhcp-proxy-client`
		Specifies the DHCP client-proxy feature as the global default mechanism.
		• The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
		Note You can provide as few as one or as many as ten DHCP servers for the proxy client (the Cisco router or access server) to use. The DHCP servers provide temporary IP addresses.

Step 4	**`ip dhcp-server`** `[`*`ip-address`* `|` *`name`*`]`
	Example:
	`Device(config)# ip dhcp-server 209.165.201.1`
		(Optional) Specifies the IP address of a DHCP server for the proxy client to use.

Step 5	**`end`**
	Example:
	`Device(config)# end`
		Exits global configuration mode.

Defining Local Address Pooling as the Global Default Mechanism

Perform this task to define local address pooling as the global default mechanism.

Note If no other pool is defined, a local pool called “default” is used. Optionally, you can associate an address pool with a named pool group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool local**
4. **ip local pool** {*named-address-pool* | **default**} *first-IP-address* [*last-IP-address*] [**group** *group-name*] [**cache-size** *size*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	`ip address-pool local`
	Example:
	`Device(config)# ip address-pool local`
		Specifies local address pooling as the global default mechanism.

Step 4	**ip local pool** {*named-address-pool* | d**efault**} *first-IP-address* [*last-IP-address*] [**group** *group-name*] [**cache-size** *size*] **Example:** `Device(config)# ip local pool default 192.0.2.1`	Creates one or more local IP address pools.

Controlling DHCP Network Discovery

Perform the steps in this section to allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server using PPP IPCP extensions.

The **ip dhcp-client network-discovery** global configuration command provides a way to control the DHCP network discovery mechanism. The number of DHCP Inform or Discovery messages can be set to 1 or 2, which determines how many times the system sends the DHCP Inform or Discover messages before stopping network discovery. You can set a timeout period from 3 to 15 seconds, or leave the default timeout period at 15 seconds. The default for the **informs** and **discovers** keywords is 0, which disables the transmission of these messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp-client network-discovery informs** *number-of-messages* **discovers** *number-of-messages* **period** *seconds*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`ip dhcp-client network-discovery informs`** *`number-of-messages`* **`discovers`** *`number-of-messages`* **`period`** *`seconds`*
	Example:
	`Device(config)# ip dhcp-client network-discovery informs 2 discovers 2 period 2`
		Provides control of the DHCP network discovery mechanism by allowing the number of DHCP Inform and Discover messages to be sent, and a timeout period for retransmission, to be configured.

Configuring IP Address Assignment

Perform this task to configure IP address allignment.

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using SLIP or PPP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** {**named-address-pool** | **default**} {*first-IP-address* [*last-IP-address]*} [**group** *group-name*] [**cache-size** *size*]}
4. **interface** *type number*
5. **peer default ip address pool** *pool-name-list*
6. **peer default ip address pool dhcp**
7. **peer default ip address** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip local pool** {*named-address-pool* | d**efault**} {*first-IP-address* [*last-IP-address*]} [**group** *group-name*] [**cache-size** *size*]}
	Example:
	`Device(config)# ip local pool default 192.0.2.0`
		Creates one or more local IP address pools.

Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 2/0`	Specifies the interface and enters interface configuration mode.
Step 5	**peer default ip address pool** *pool-name-list* **Example:** `Device(config-if)# peer default ip address pool 2`	Specifies the pool or pools for the interface to use.
Step 6	**peer default ip address pool dhcp** **Example:** `Device(config-if)# peer default ip address pool dhcp`	Specifies DHCP as the IP address mechanism on this interface.
Step 7	**peer default ip address** *ip-address* **Example:** `Device(config-if)# peer default ip address 192.0.2.2`	Specifies the IP address to assign to all dial-in peers on an interface.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the **debug lapb** command and the **debug ppp negotiations**, **debug ppp errors**, and **debug ppp packets** commands. You can determine whether Link Access Procedure, Balanced (LAPB) has been established on a connection by using the **show interface** command.

Disabling or Reenabling Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior or to reenable it once it has been disabled, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no peer neighbor-route**
5. **peer neighbor-route**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		- Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`interface`** *`type number`*
	Example:
	`Device(config)# interface ethernet 0/1`
		Specifies the interface and enters interface configuration mode.

Step 4	**`no peer neighbor-route`**
	Example:
	`Device(config-if)# no peer neighbor-route`
		Disables creation of neighbor routes.

Step 5	**`peer neighbor-route`**
	Example:
	`Device(config-if)# peer neighbor-route`
		Reenables creation of neighbor routes.
		Note If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Configuring Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. The Cisco implementation of MLP supports the fragmentation and packet sequencing specifications in RFC 1990. Additionally, you can change the default endpoint discriminator value that is supplied as part of user authentication. Refer to RFC 1990 for more information about the endpoint discriminator.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over synchronous and asynchronous serial and BRI and PRI types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

Perform the tasks in the following sections, as required for your network, to configure MLP:

- Configuring MLP on Synchronous Interfaces, page 18
- Configuring MLP on Asynchronous Interfaces, page 19
- Configuring MLP on a Single ISDN BRI Interface, page 21
- Configuring MLP on Multiple ISDN BRI Interfaces, page 23
- Configuring MLP Using Multilink Group Interfaces, page 25
- Changing the Default Endpoint Discriminator, page 27

Configuring MLP on Synchronous Interfaces

To configure Multilink PPP on synchronous interfaces, you configure the synchronous interfaces to support PPP encapsulation and Multilink PPP.

Perform this task to configure a synchronous interface.

SUMMARY STEPS

1. **enable**
2. **configuration terminal**
3. **interface serial** 1
4. **no ip address**
5. **encapuslation ppp**
6. **ppp multilink**
7. **pulse-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface serial** *number* **Example:** `Device(config)# interface serial 1`	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Specifies no IP address for the interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 7	**pulse-time** *seconds* **Example:** `Device(config-if)# pulse-time 60`	Enables pulsing data terminal ready (DTR) signal intervals on an interface. **Note** Repeat these steps for additional synchronous interfaces, as needed.

Configuring MLP on Asynchronous Interfaces

Perform the following steps in this section to configure an asynchronous interface to support DDR and PPP encapsulation and then configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

At some point, adding more asynchronous interfaces does not improve performance, With the default maximum transmission unit (MTU) size, MLP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the maximum transmission unit (MTU) size is small or large bursts of short frames occur.

Note To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the **dialer load-threshold** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface async** *number*
4. **no ip address**
5. **dialer in-band**
6. **dialer rotary-group** *number*
7. **dialer load-threshold** *load* [**inbound** | **outbound** | **either**]

8. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface async** *number* **Example:** `Device(config)# interface async 0/0`	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Specifies no IP address for the interface.
Step 5	Device(config-if)# **encapsulation ppp** **Example:** `Device# configure terminal`	Enables PPP encapsulation.
Step 6	**dialer in-band** **Example:** `Device(config-if)# encapsulation ppp`	Enables DDR on the interface.
Step 7	**dialer rotary-group** *number* **Example:** `Device(config-if)# dialer rotary-group 1`	Includes the interface in a specific dialer rotary group.
Step 8	**dialer load-threshold** *load* [**inbound** | **outbound** | **either**] **Example:** `Device(config-if)# dialer load-threshold 100`	Configures bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
Step 9	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.

Configuring MLP on a Single ISDN BRI Interface

To enable MLP on a single ISDN BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

Perform this task to enable PPP on an ISDN BRI interface.

If you do not use PPP authentication procedures (Step 8), your telephone service must pass caller ID information.

The load threshold number is required. For an example of configuring MLP on a single ISDN BRI interface, see the "Example: MLP on One ISDN BRI Interface" section on page 43.

Note When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a high idle timer. The **dialer-load threshold 1** command does not keep a multilink bundle of *n* links connected indefinitely, and the **dialer-load threshold 2** command does not keep a multilink bundle of two links connected indefinitely.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **encapsulation ppp**
6. **dialer idle-timeout** *seconds* [**inbound** | **either**]
7. **dialer load-threshold** *load*
8. **dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
9. **dialer-group** *group-number*
10. **ppp authentication pap**
11. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface bri** *number* **Example:** `Device(config)# interface bri 1`	Specifies an interface and benters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 192.0.2.0 255.255.255.224`	Provides an appropriate protocol address for the interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**dialer idle-timeout** *seconds* [**inbound** | **either**] **Example:** `Device(config-if)# dialer idle-timeout 60`	Specifies the duration of idle time in seconds after which a line will be disconnected. • By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 7	**dialer load-threshold** *load* **Example:** `Device(config-if)# dialer load-threshold 60`	Specifies the dialer load threshold for bringing up additional WAN links.
Step 8	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]] **Example:** `Device(config-if)# dialer map protocol 192.0.2.1`	Configures the ISDN interface to call the remote site.
Step 9	**dialer-group** *group-number* **Example:** `Device(config-if)# dialer-group 3`	Controls access to this interface by adding it to a dialer access group.
Step 10	**ppp authentication pap** **Example:** `Device(config-if)# ppp authentication pap`	(Optional) Enables PPP authentication.
Step 11	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Configures MLP on the dialer rotary group.

Configuring MLP on Multiple ISDN BRI Interfaces

To enable MLP on multiple ISDN BRI interfaces, set up a dialer rotary interface and configure it for Multilink PPP, and then configure the BRI interfaces separately and add them to the same rotary group.

To set up the dialer rotary interface for the BRI interfaces, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *dialer number*
4. **ip address** *address mask*
5. **encapsulation ppp**
6. **dialer in-band**
7. **dialer idle-timeout** *seconds* [**inbound** | **either**]
8. **dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
9. **dialer rotary-group** *number*
10. **dialer load-threshold** *load*
11. **dialer-group** *number*
12. **ppp authentication chap**
13. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface dialer** *number* **Example:** `Device(config)# interface dialer 1`	Specifies the dialer rotary interface and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.0.2.0 255.255.255.224`	Specifies the protocol address for the dialer rotary interface.

	Command or Action	Purpose
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**dialer in-band** **Example:** `Device(config-if)# dialer in-band`	Specifies in-band dialing.
Step 7	**dialer idle-timeout** *seconds* [**inbound** | **either**] **Example:** `Device(config-if)# dialer idle-timeout 60`	Specifies the duration of idle time in seconds after which a line will be disconnected. • By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 8	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]] **Example:** `Device(config-if)# dialer map protocol 192.0.2.1`	Maps the next hop protocol address and name to the dial string needed to reach it.
Step 9	**dialer rotary-group** *number* **Example:** `Device(config-if)# dialer rotary-group 1`	Adds the interface to the rotary group.
Step 10	**dialer load-threshold** *load* **Example:** `Device(config-if)# dialer load-threshold 2`	Specifies the dialer load threshold, using the same threshold as the individual BRI interfaces.
Step 11	**dialer-group** *number* **Example:** `Device(config-if)# dialer-group 2`	Controls access to the interface by adding it to a dialer access group.
Step 12	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables PPP CHAP authentication.
Step 13	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.

If you do not use PPP authentication procedures (Step 10), your telephone service must pass caller ID information.

Repeat Steps 1 through 9 for each BRI that you want to belong to the same dialer rotary group.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. The **dialer load-threshold 1** command does not keep a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command does not keep a multilink bundle of two links connected indefinitely.)

Note Prior to Cisco IOS Release 12.1, when MLP was used in a dialer profile, a virtual access interface was always created as the bundle. It was bound to both the B channel and the dialer profile interfaces after creation and cloning. The dialer profile interface could act as the bundle without help from a virtual access interface. But with the Dynamic Multiple Encapsulations feature available in Cisco IOS Release 12.1, it is no longer the virtual access interface that is added into the connected group of the dialer profile, but the dialer profile itself. The dialer profile becomes a connected member of its own connected group. See the "Dynamic Multiple Encapsulations over ISDN Example" in the module "Configuring Peer-to-Peer DDR with Dialer Profiles" in this module, for more information about dynamic multiple encapsulations and its relation to Multilink PPP.

For an example of configuring MLP on multiple ISDN BRI interfaces, see the section "Example: MLP on Multiple ISDN BRI Interfaces" section on page 43.

Configuring MLP Using Multilink Group Interfaces

MLP can be configured by assigning a multilink group to a virtual template configuration. Virtual templates allow a virtual access interface to dynamically clone interface parameters from the specified virtual template. If a multilink group is assigned to a virtual template, and then the virtual template is assigned to a physical interface, all links that pass through the physical interface will belong to the same multilink bundle.

Note If a multilink group interface has one member link, the amount of bandwidth available will not change when a multilink interface is shut down. Therefore, you can shut down the multilink interface by removing its link.

A multilink group interface configuration will override a global multilink virtual template configured with the **multilink virtual template** command.

Multilink group interfaces can be used with ATM, PPP over Frame Relay, and serial interfaces.

To configure MLP using a multilink group interface, perform the following tasks:

- Configure the multilink group.
- Assign the multilink group to a virtual template.
- Configure the physical interface to use the virtual template.

Perform the following tasks in this section to configure the multilink group. For an example of how to configure MLP over an ATM PVC using a multilink group, see the section "Example: MLP Using Multilink Group Interfaces over ATM" section on page 44.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*

4. **ip address** *address mask*
5. **encapsulation ppp**
6. **exit**
7. **interface virtual template** *number*
8. **ppp multilink group** *group-number*
9. **exit**
10. **interface atm** *interface-number***.***subinterface-number* **point-to-point**
11. **pvc** *vpi***/***vli*
12. **protocol ppp virtual-template** *name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	**`interface multilink`** *`group-number`*
	Example:
	`Device(config)# interface multilink 2`
		Creates a multilink bundle and enters interface configuration mode to configure the bundle.

Step 4	**`ip address`** *`address mask`*
	Example:
	`Device(config-if)# ip address 192.0.2.1 255.255.255.224`
		Sets a primary IP address for an interface.

Step 5	**`encapsulation ppp`**
	Example:
	`Device(config-if)# encapsulation ppp`
		Enables PPP encapsulation.

Step 6	**`exit`**
	Example:
	`Device(config-if)# exit`
		Exits interface configuration mode.

Step 7	**`interface virtual template`** *`number`*
	Example:
	`Device(config)# interface virtual template 1`
		Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 2`	Restricts a physical link to joining only a designated multilink group interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 10	**interface atm** *interface-number.subinterface-number* **point-to-point** **Example:** `Device(config)# interface atm 1.2 point-to-point`	Configures an ATM interface and enters interface configuration mode.
Step 11	**pvc** *vpi***/***vci* **Example:** `Device(config-if)# pvc 1/100`	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode.
Step 12	**protocol ppp virtual-template** *name* **Example:** `Device(config-if-atm-vc)# protocol ppp virtual-template 2`	Configures VC multiplexed encapsulation on a PVC.

Step 13	**end**
	Example:
	`Device(config-if-atm-vc)# end`
		Exits ATM virtual circuit configuration mode.

Changing the Default Endpoint Discriminator

By default, when the system negotiates use of MLP with the peer, the value that is supplied for the endpoint discriminator is the same as the username used for authentication. That username is configured for the interface by the Cisco IOS **ppp chap hostname** or **ppp pap sent-username** command, or defaults to the globally configured hostname (or stack group name, if this interface is a Stack Group Bidding Protocol, or SGBP, group member).

Perform this task to override or change the default endpoint discriminator. For an example of how to change the default endpoint discriminator, see the "Example: Changing the Default Endpoint Discriminator" section on page 37.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virutal template** *number*

4. **ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string char-string**}

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

Step 3	`interface virtual template` *number* **Example:** `Device(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces and enters interface configuration mode.

Step 4	**ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string** *char-string*}
	Example:
	`Device(config-if)# ppp multilink endpoint ip 192.0.2.0`
		Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer.

Configuring MLP Interleaving

Perform the following tasks to configure MLP and interleaving on a configured and operational interface or virtual interface template.

Configuring MLP Interleaving and Queueing

Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic; small real-time packets are not multilink encapsulated and are sent between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows.

Weighted fair queueing on MLP works on the packet level, not at the level of multilink fragments. Thus, if a small real-time packet gets queued behind a larger best-effort packet and no special queue has been reserved for real-time packets, the small packet will be scheduled for transmission only after all the fragments of the larger packet are scheduled for transmission.

Weighted fair queueing is supported on all interfaces that support Multilink PPP, including MLP virtual access interfaces and virtual interface templates. Weighted fair queueing is enabled by default.

Fair queueing on MLP overcomes a prior restriction. Previously, fair queueing was not allowed on virtual access interfaces and virtual interface templates. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

Interleaving applies only to interfaces that can configure a multilink bundle interface. These restrictions include virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces.

Multilink and fair queueing are not supported when a multilink bundle is off-loaded to a different system using Multichassis Multilink PPP (MMP). Thus, interleaving is not supported in MMP networking designs.

MLP support for interleaving can be configured on virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces. To configure interleaving, complete the following tasks:

- Configure the dialer interface, BRI interface, PRI interface, or virtual template.
- Configure MLP and interleaving on the interface or template.

Note Fair queueing, which is enabled by default, must remain enabled on the interface.

Note Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves. For example, the following line shows interleaves:

```
Output queue: 315/64/164974/31191 (size/threshold/drops/interleaves)
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual template** *number*
4. **ppp multilink**
5. **ppp multilink interleave**
6. **ppp multilink fragment delay** *milliseconds*
7. **ip rtp reserve** *lowest-udp-port range-of-ports* [*maximum-bandwidth*]
8. **exit**
9. **multilink virtual-template** *virtual-template-number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Device> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Device# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface virtual template** *number* **Example:** `Device(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.
Step 4	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 5	**ppp multilink interleave** **Example:** `Device(config-if)# configure terminal`	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.
Step 6	**ppp multilink fragment delay** *milliseconds* **Example:** `Device(config-if)# ppp multilink fragment delay 50`	Specifies a maximum size, in units of time, for packet fragments on an MLP bundle.
Step 7	**ip rtp reserve** *lowest-udp-port range-of-ports* [*maximum-bandwidth*] **Example:** `Device(config-if)# ip rtp reserve 1 2`	Reserves a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**multilink virtual-template** *virtual-template-number* **Example:** `Device(config)# multilink virtual-template 1`	For virtual templates only, applies the virtual template to the multilink bundle. **Note** This step is not used for ISDN or dialer interfaces.

Configuring MLP Inverse Multiplexer and Distributed MLP

The distributed MLP (dMLP) feature combines T1/E1 lines in a WAN line card on a Cisco 7600 series router into a bundle that has the combined bandwidth of the multiple T1/E1 lines. You choose the number of bundles and the number of T1/E1 lines in each bundle, which allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line without having to purchase a T3 line.

Nondistributed MLP is not supported on the Cisco 7600 series router. With distributed MLP, you can increase the router’s total capacity.

The MLP Inverse Multiplexer feature was designed for Internet service providers (ISPs) that want to have the bandwidth of multiple T1 lines with performance comparable to that of an inverse multiplexer without the need of buying standalone inverse-multiplexing equipment. A Cisco router supporting dMLP can bundle multiple T1 lines in a CT3 or CE3 interface or channelized STM1. Bundling is more economical than purchasing an inverse multiplexer, and eliminates the need to configure another piece of equipment.

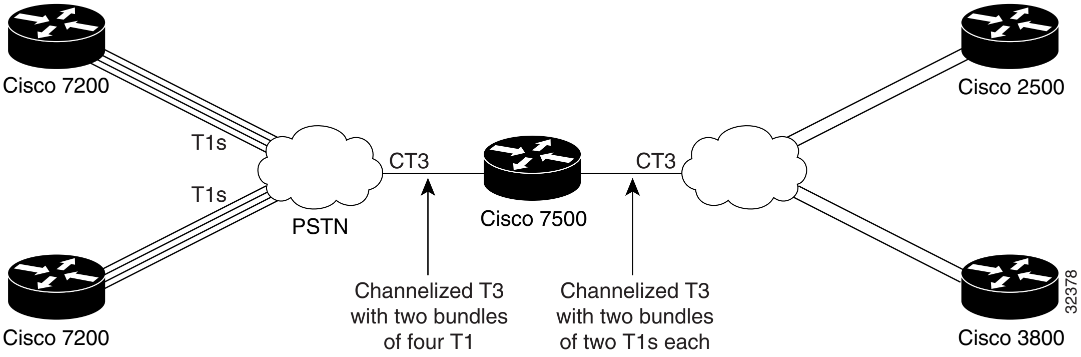
This feature supports the CT3 CE3 data rates without taxing the Route Processor (RP) and CPU by moving the data path to the line card. This feature also allows remote sites to purchase multiple T1 lines instead of a T3 line, which is especially useful when the remote site does not need the bandwidth of an entire T3 line.

This feature allows multilink fragmentation to be disabled, so multilink packets are sent using Cisco Express Forwarding on all platforms, if fragmentation is disabled. Cisco Express Forwarding is supported with fragmentation enabled or disabled.

Note If a router cannot send out all the packets (some packets are dropped by Quality of Service (QoS)), late drops occur. These late drops are displayed when the **show interface** command is executed.
If there is no service policy on the dMLP interface, when a **ppp multilink interleave** is configured on the dMLPPP interface, a QoS policy is enabled internally.

Figure 2 shows a typical network using a dMLP link. The Cisco 7600 series router is connected to the network with a CT3 line that has been configured with dMLPP to carry two bundles of four T1 lines each. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 2 Diagram of a Typical VIP MLP Topology


Cisco 7200
T1s
T1s
Cisco 7200
PSTN
CT3
Cisco 7500
CT3
Channelized T3 with two bundles of four T1
Channelized T3 with two bundles of two T1s each
Cisco 2500
Cisco 3800
32378

Before beginning the MLP Inverse Multiplexer configuration tasks, make note of the following prerequisites and restrictions.

Prerequisites

- Distributed Cisco Express Forwarding switching must be enabled for distributed MLP.
- One of the following port adapters is required:
  - CT3IP
  - PA-MC-T3
  - PA-MC-2T3+
  - PA-MC-E3
  - PA-MC-8T1
  - PA-MC-4T1
  - PA-MC-8E1
- All 16 E1s can be bundled from a PA-MC-E3 in a VIP4-80.

Restrictions

The following restrictions apply to the dMLP feature:

> **Note** Distributed MLP is supported only for member links configured at T1/E1 or subrate T1/E1 speeds. Channelized STM-1/T3/T1 interfaces also support dMLP at T1/E1 or subrate T1/E1 speeds. Distributed MLP is not supported for member links configured at clear-channel T3/E3 or higher interface speeds.

- T1 and E1 lines cannot be mixed in a bundle.
- T1 lines in a bundle should have the same bandwidth.
- All lines in a bundle must reside on the same port adapter.
- MLP bundles across FlexWAN or Enhanced FlexWAN port adapters are not supported.
- Hardware compression is not supported.
- Encryption is not supported.
- Software compression is not recommended because CPU usage would void performance gains.
- The maximum differential delay supported is 50 milliseconds (ms).
- Fragmentation is not supported on the transmit side.
- dMLP across shared port adapters (SPAs) is not supported.
- Hardware and software compression is not supported.
- Encryption is not supported.
- The maximum differential delay supported is 50 ms when supported in hardware, and 100 ms when supported in software.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

To configure a multilink bundle, perform the tasks in the following sections:

- Creating a Multilink Bundle, page 32 (required)
- Assigning an Interface to a Multilink Bundle, page 33 (required)
- Disabling PPP Multilink Fragmentation, page 35 (optional)

Creating a Multilink Bundle

Perform the following tasks to create a multilink bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask*
5. **encapsulation ppp**
6. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.0.2.9 255.255.255.224`	Assigns an IP address to the multilink interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

Perform this task to assign an interface to a multilink bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group number*
4. **no ip address**
5. **keepalive**
6. **encapsulation ppp**
7. **ppp multilink group** *group-number*
8. **ppp multilink**
9. **ppp authentication chap**
10. **pulse-time seconds**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes any specified IP address.
Step 5	**keepalive** **Example:** `Device(config-if)# keepalive`	Sets the frequency of keepalive packets.
Step 6	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 7	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink 12`	Restricts a physical link to joining only the designated multilink-group interface.
Step 8	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 9	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables CHAP authentication.
Step 10	**pulse-time** *seconds* **Example:** `Device(config-if)# pulse-time 10`	(Optional) Configures DTR signal pulsing.

Caution Do not install a router to the peer address while configuring an MLP lease line. This installation can be disabled when **no ppp peer-neighbor-route** command is used under the MLPPP bundle interface.

Disabling PPP Multilink Fragmentation

Perform the following task to disable PPP multilink fragmentation.

SUMMARY STEPS

1. **enable**
2. **configuration terminal**
3. **interface multilink** *group number*
4. **ppp multilink fragment disable**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.

Step 4	**ppp multilink fragment disable**
	Example:
	`Device(config-if)#`
	`ppp multilink fragment disable`
		(Optional) Disables PPP multilink fragmentation.

Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits privileged EXEC mode.

Monitoring and Maintaining PPP and MLP Interfaces

Perform this task to display MLP and MMP bundle information.

SUMMARY STEPS

1. **enable**
2. **show ppp multilink**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ppp multilink** **Example:** `Device# show ppp multilink`	Displays MLP and MMP bundle information.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.

Configuration Examples for PPP and MLP

The following sections provide various PPP configuration examples:

- Examples: CHAP with an Encrypted Password:, page 36
- Example: DHCP Network Control, page 38
- Example: IP Address Pooling, page 38
- Example: MPPC Interface Configuration, page 40
- Examples: MLP, page 41
- Example: MLP Interleaving and Queueing for Real-Time Traffic, page 44
- Example: Multilink Interface Configuration for Distributed MLP, page 45
- Example: PAP commands for a one way authentication, page 46
- Example: T3 Controller Configuration for an MLP Multilink Inverse Multiplexer, page 47
- Example: User Maximum Links Configuration, page 47

Examples: CHAP with an Encrypted Password:

The following examples show how to enable CHAP on serial interface 0 of three devices:

Configuration of Device yyy

```
hostname yyy
interface serial 0
```

```
 encapsulation ppp
 ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Device xxx

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

Configuration of Device zzz

```
hostname zzz
interface serial 0
 encapsulation ppp
 ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords are encrypted and the display looks similar to the following:

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

Example: Changing the Default Endpoint Discriminator

The following partial example changes the MLP endpoint discriminator from the default CHAP hostname C-host1 to the E.164-compliant telephone number 555-0100:

```
.
.
.
interface dialer 0
 ip address 10.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name R-host1
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp chap hostname C-host1
 ppp multilink endpoint phone 555-0100
.
.
.
```

Example: DHCP Network Control

The following partial example shows how to add the **ip dhcp-client network-discovery** command to the "Example: IP Address Pooling" section on page 38 to allow peer routers to more dynamically discover DNS and NetBIOS name servers. If the **ip dhcp-client network-discovery** command is disabled, the system falls back to the static configurations made using the **async-bootp dns-server** and **async-bootp nb-server** global configuration commands.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
ip dhcp-client network-discovery informs 2 discovers 2 period 12
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
.
.
.
```

Example: IP Address Pooling

The following example shows how to configure a modem to dial in to a Cisco access server and obtain an IP address from the DHCP server. This configuration allows the user to log in and browse an NT network. Notice that the dialer 1 and group-async 1 interfaces are configured with the **ip unnumbered loopback** command, so that the broadcast can find the dialup clients and the client can see the NT network.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
!
!
```

```
controller t1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller t1 1
 framing esf
 clock source line secondary
 linecode b8zs
!
interface loopback 0
 ip address 10.47.252.254 255.255.252.0
!
interface ethernet 0
 ip address 10.47.0.5 255.255.252.0
 ip helper-address 10.47.0.131
 ip helper-address 10.47.0.255
 no ip route-cache
 no ip mroute-cache
!
interface serial 0
 no ip address
 no ip mroute-cache
 shutdown
!
interface serial 1
 no ip address
 shutdown
!
interface serial 0:23
 no ip address
 encapsulation ppp
 no ip mroute-cache
 dialer rotary-group 1
 dialer-group 1
 isdn incoming-voice modem
 no fair-queue
 no cdp enable
!
interface group-async 1
 ip unnumbered loopback 0
 ip helper-address 10.47.0.131
 ip tcp header-compression passive
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 async mode interactive
 peer default ip address dhcp
 no fair-queue
 no cdp enable
 ppp authentication chap
 group-range 1 24
!
interface dialer 1
 ip unnumbered loopback 0
 encapsulation ppp
 dialer in-band
 dialer-group 1
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink
```

```
!
router ospf 172
 redistribute connected subnets
 redistribute static
 network 10.47.0.0 0.0.3.255 area 0
 network 10.47.156.0 0.0.3.255 area 0
 network 10.47.168.0 0.0.3.255 area 0
 network 10.47.252.0 0.0.3.255 area 0
!
ip local pool RemotePool 10.47.252.1 10.47.252.24
ip classless
ip route 10.0.140.0 255.255.255.0 10.59.254.254
ip route 10.2.140.0 255.255.255.0 10.59.254.254
ip route 10.40.0.0 255.255.0.0 10.59.254.254
ip route 10.59.254.0 255.255.255.0 10.59.254.254
ip route 172.23.0.0 255.255.0.0 10.59.254.254
ip route 192.168.0.0 255.255.0.0 10.59.254.254
ip ospf name-lookup
no logging buffered
access-list 101 deny ip any host 255.255.255.255
access-list 101 deny ospf any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
snmp-server community public RO
!
line con 0
line 1 24
 autoselect during-login
 autoselect ppp
 modem InOut
 transport input all
line aux 0
line vty 0 4
 password Password
!
scheduler interval 100
end
```

Example: MPPC Interface Configuration

The following example shows how to configure asynchronous interface 1 to implement MPPC and ignore the protocol field compression flag negotiated by LCP:

```
interface async1
 ip unnumbered ethernet0
 encapsulation ppp
 async default routing
 async dynamic routing
 async mode interactive
 peer default ip address 172.21.71.74
 compress mppc ignore-pfc
```

The following example creates a virtual access interface (virtual template interface 1) and serial interface 0, which is configured for X.25 encapsulation. MPPC values are configured on the virtual template interface and will ignore the negotiated protocol field compression flag.

```
interface ethernet0
 ip address 172.20.30.102 255.255.255.0
!
interface virtual-template1
 ip unnumbered ethernet0
 peer default ip address pool vtemp1
 compress mppc ignore-pfc
!
interface serial0
 no ipaddress
no ip mroute-cache
encapsulation x25
x25 win 7
x25 winout 7
x25 ips 512
x25 ops 512
clock rate 50000
!
ip local pool vtemp1 172.20.30.103 172.20.30.104
ip route 0.0.0.0 0.0.0.0 172.20.30.1
!
translate x25 31320000000000 virtual-template 1
```

Examples: MLP

This section contains the following MLP examples:

- Example: MLP on Synchronous Serial Interfaces, page 41
- Example: MLP on One ISDN BRI Interface, page 43
- Example: MLP on Multiple ISDN BRI Interfaces, page 43
- Example: MLP Inverse Multiplexer Configuration, page 44
- Example: MLP Using Multilink Group Interfaces over ATM, page 44
- Example: Changing the Default Endpoint Discriminator, page 37

Example: MLP on Synchronous Serial Interfaces

The following example shows how the configuration commands are used to create the inverse multiplexing application:

Device A Configuration

```
hostname DeviceA
!
!
username DeviceB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Device B Configuration

```
hostname DeviceB
!
!
username DeviceB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.2.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Example: MLP on One ISDN BRI Interface

The following example shows how to enable MLP on BRI interface 0. When a BRI is configured, no dialer rotary group configuration is required, because an ISDN interface is a rotary group by default.

```
interface bri 0
 description connected to ntt 81012345678902
 ip address 172.31.1.7 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 30
 dialer load-threshold 40 either
 dialer map ip 172.31.1.8 name user1 81012345678901
 dialer-group 1
 ppp authentication pap
 ppp multilink
```

Example: MLP on Multiple ISDN BRI Interfaces

The following example shows how to configure multiple ISDN BRI interfaces to belong to the same dialer rotary group for Multilink PPP. The **dialer rotary-group** command is used to assign each of the ISDN BRI interfaces to that dialer rotary group.

```
interface BRI 0
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI 2
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface Dialer 0
 ip address 10.0.0.2 255.0.0.0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 500
 dialer map ip 10.0.0.1 name user1 broadcast 81012345678901
 dialer load-threshold 30 either
 dialer-group 1
 ppp authentication chap
 ppp multilink
```

Example: MLP Using Multilink Group Interfaces over ATM

The following example shows how to configure MLP over an ATM PVC using a multilink group:

```
interface multilink 1
 ip address 10.200.83.106 255.255.255.252
 ip tcp header-compression iphc-format delay 20000
 service policy output xyz
 encapsulation ppp
 exit
 ppp multilink
 ppp multilink fragment delay 10
 ppp multilink interleave
 ppp timeout multilink link remove 10
 ip rtp header-compression iphc-format

interface virtual-template 3
 bandwidth 128
 ppp multilink group 1

interface atm 4/0.1 point-to-point
 pvc 0/32
 abr 100 80
 protocol ppp virtual-template 3
```

Example: MLP Inverse Multiplexer Configuration

This example shows how to verify the display information of the newly created multilink bundle:

```
Device# show ppp multilink

Multilink1, bundle name is group1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links:4 active, 0 inactive (max not set, min not set)
 Serial1/0/0:1
 Serial1/0/0/:2
 Serial1/0/0/:3
 Serial1/0/0/:4
```

Example: MLP Interleaving and Queueing for Real-Time Traffic

The following example defines a virtual interface template that enables MLP interleaving and a maximum real-time traffic delay of 20 milliseconds, and then applies that virtual template to the MLP bundle:

```
interface virtual-template 1
 ip unnumbered ethernet 0
 ppp multilink
 ppp multilink interleave
 ppp multilink fragment delay 20
 ip rtp interleave 32768 20 1000
multilink virtual-template 1
```

The following example enables MLP interleaving on a dialer interface that controls a rotary group of BRI interfaces. This configuration permits IP packets to trigger calls.

```
interface BRI 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 2
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 4
 encapsulation ppp
 dialer rotary-group 1
!
interface Dialer 0
 description Dialer group controlling the BRIs
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name name1 14802616900
 dialer-group 1
 ppp authentication chap
! Enables Multilink PPP interleaving on the dialer interface and reserves
! a special queue.
 ppp multilink
 ppp multilink interleave
 ip rtp reserve 32768 20 1000
! Keeps fragments of large packets small enough to ensure delay of 20 ms or less.
 ppp multilink fragment delay 20
dialer-list 1 protocol ip permit
```

Example: Multilink Interface Configuration for Distributed MLP

In the following example, four multilink interfaces are created with distributed Cisco Express Forwarding switching and MLP enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hosstname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0:1
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp multilink
 ppp multilink group 1
```

```
interface serial 1/0/0:2
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0:3
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0:4
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```

Example: PAP commands for a one way authentication

The following example shows how to authenticate PAP commands for a one way authentication scenario:

Note Only the relevant sections of the configuration are shown.

```
Calling Side (Client) Configuration
interface BRI0

! --- BRI interface for the dialout.

 ip address negotiated
 encapsulation ppp

! --- Use PPP encapsulation. This command is a required for PAP.

 dialer string 3785555 class 56k

! --- Number to dial for the outgoing connection.

 dialer-group 1
 isdn switch-type basic-ni
 isdn spid1 51299611110101 9961111
 isdn spid2 51299622220101 9962222
 ppp authentication pap callin

! --- Use PAP authentication for incoming calls.
! --- The callin keyword has made this a one-way authentication scenario.
! --- This router (client) will not request that the peer (server) authenticate
! --- itself back to the client.
```

```
 ppp pap sent-username PAPUSER password 7 <deleted>

! --- Permit outbound authentication of this router (client) to the peer.
! --- Send a PAP AUTH-REQ packet to the peer with the username PAPUSER and password.
! --- The peer must have the username PAPUSER and password configured on it.

Receiving Side (Server) Configuration
username PAPUSER password 0 cisco

! --- Username PAPUSER is the same as the one sent by the client.
! --- Upon receiving the AUTH-REQ packet from the client, we will verify that the
! --- username and password match the one configured here.

interface Serial0:23

! --- This is the D-channel for the PRI on the access server receiving the call.

 ip unnumbered Ethernet0
 no ip directed-broadcast
 encapsulation ppp

! --- Use PPP encapsulation. This command is a required for PAP.

 dialer-group 1
 isdn switch-type primary-ni
 isdn incoming-voice modem
 peer default ip address pool default
 fair-queue 64 256 0
 ppp authentication pap

! --- Use PAP authentication for incoming calls.
! --- This router (server) will request that the peer authenticate itself to us.
! --- Note: the callin option is not used as this router is not initiating the call.
```

Example: T3 Controller Configuration for an MLP Multilink Inverse Multiplexer

The following example shows how to configure the T3 controller and create four channelized interfaces:

```
controller T3 1/0/0
framing m23
cablelength 10
t1 1 timeslots 1-24
t1 2 timeslots 1-24
t1 3 timeslots 1-24
t1 4 timeslots 1-24
```

Example: User Maximum Links Configuration

The following example shows how to configure the username user1 and establish a maximum of five connections. user1 can connect through serial interface 1/0, which has a dialer map configured for it, or through PRI interface 0/0:23, which has dialer profile interface 0 dedicated to it.

The **aaa authorization network default local** command must be configured. PPP encapsulation and authentication must be enabled on all the interfaces that user1 can connect to.

```
aaa new-model
aaa authorization network default local
enable secret password1
enable password password2
!
username user1 user-maxlinks 5 password password3
!
interface Serial0/0:23
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap
 ppp multilink
!
interface Serial1/0
 ip address 209.165.201.1 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 10.2.2.13 name user1 12345
 dialer-group 1
 ppp authentication chap
!
interface Dialer0
 ip address 209.165.200.225 255.255.255.0
 encapsulation ppp
 dialer remote-name user1
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

Additional References

The following sections provide references related to the Configuring Media-Independent PPP and Multilink PPP feature.

Related Documents

Related Topic	Document Title
Asynchronous SLIP and PPP	*"Configuring Asynchronous SLIP and PPP"* module in the *Cisco IOS Dial Technologies Configuration Guide*
MCHAP	*MS-CHAP Support*
Dial commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Dial Technologies Command Reference.*

RFCs

RFC	Title
RFC 1994	PPP Challenge Handshake Authentication Protocol (CHAP)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Media-Independent PPP and Multilink PPP

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring Media-Independent PPP and Multilink PPP

Feature Name	Releases	Feature Information
Multilink PPP	11.2(1) 12.2(8)T 11.2(6)P 12.1(3)T 12.3(13)BC 12.2(27)SBB 12.2(31)SB2 15.0(1)M 12.2(33)SRE 15.2(2)S Cisco IOS XE Release 3.14S	Multilink PPP provides a method for spreading traffic across multiple physical WAN links. The following sections provide information about this feature: • Information About Media-Independent PPP and Multilink PPP, page 2 • How to Configure Media-Independent PPP and Multilink PPP, page 6 The following commands were introduced or modified: **ppp multilink**, **ppp multilink group**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.