Configuring Media-Independent PPP and Multilink PPP

This module describes how to configure Media-Independent PPP and Multilink PPP features that can be configured on any interface.

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 at the end of this module.

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.

Information About Media-Independent PPP and Multilink PPP

PPP Encapsulation Overview

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)
- Synchronous serial

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, a device might shut down a link if it detects a loop.

Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links while providing multivendor interoperability, packet fragmentation, proper sequencing, and load calculation on both inbound and outbound traffic. Cisco's implementation of Multilink PPP 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.

Multilink PPP 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 or outbound traffic, as required for the traffic between the specific sites. Multilink PPP provides bandwidth on demand and reduces transmission latency across WAN links.

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

Multilink PPP Minimum Links Mandatory

Multilink PPP allows multiple PPP links to be established in parallel to the same destination. Multilink PPP is often used to increase the amount of bandwidth between points. The Multilink PPP Minimum Links Mandatory feature enables you to configure the minimum number of links that are required in a Multilink PPP bundle to keep the bundle active.

The Multilink PPP Minimum Links Mandatory feature causes all Network Control Protocols (NCPs) for a Multilink PPP bundle to be disabled until the Multilink PPP bundle has the required minimum number of links. When a new link is added to a Multilink PPP bundle to bring the number of links up to the required number of minimum links, the NCPs are activated for the Multilink PPP bundle. When a link is removed from a Multilink PPP bundle, the number of links falls below the required minimum number of links for that Multilink PPP bundle, and the NCPs are disabled for that Multilink PPP bundle.

CHAP or PAP Authentication

PPP with Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) is often used to inform the central site about the remote devices that are connected to the site.

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

CHAP and PAP were originally specified in RFC 1334, and CHAP is updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each device or access server identifies itself using a *name*. This identification process prevents a device from placing another call to a device 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 device or access server. You can configure either CHAP or PAP on a serial interface.

Note To enable CHAP or PAP authentication on a device, the device must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local device 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 host name of the local device.

The required response consists of the following two parts:

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

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

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 a proper response, the remote device cannot connect to the local device.

CHAP transactions occur only when a link is established. The local device 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 device or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, Cisco 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 is passed to that device.

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 to support multiple simultaneous connections. The MPPC algorithm uses a Lempel-Ziv (LZ)-based compression 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 maximum transmission unit (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.

History buffers between the compressor and the 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. A 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 by using the Reset Acknowledge (RA) packet, which can consume additional time.

The following steps describe how compression negotiation between a device and a Windows 95 client occurs:

1. Windows 95 sends a request for both Stacker compression (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.
4. The device 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.

For additional information about address pooling on asynchronous interfaces and Serial Line Internet Protocol (SLIP), see the "Configuring Asynchronous SLIP and PPP" module in the *Dial Configuration Guide*.

Peer Address Allocation

A peer IP address can be allocated to an interface using one of the following methods:

- 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—The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
- 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.
- DHCP-retrieved IP address—If configured, the devices acts as proxy clients for the dialup user and retrieve 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 the addresses that are available to be assigned, and the used queue contains addresses that are in use. Addresses are stored to the free queue in the 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.

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. Configured address from the **peer default ip address** command or address from the protocol **translate** command
4. Peer-provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

Address pooling is available on all asynchronous serial interfaces and synchronous serial interfaces that are running PPP.

Multilink PPP Interleaving and Queueing

Interleaving on Multilink PPP 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 the fragments of 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 Multilink PPP works at 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 fragments of the larger packet are scheduled for transmission.

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

Interleaving applies only to interfaces that can configure a multilink bundle interface.

Multilink PPP and weighted fair queueing are not supported when a multilink bundle is offloaded to a different system using Multichassis Multilink PPP. Thus, interleaving is not supported in Multichassis Multilink PPP networking designs.

How to Configure Media-Independent PPP and Multilink PPP

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. **interface 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.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Enabling CHAP or PAP Authentication

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

CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*. For information about MS-CHAP, see the *MS-CHAP Support* document.

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. Do one of the following:
   - **ppp use-tacacs** [**single-line**]
   - **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**]	Enables the specified authentication method. **Note** Use the **ppp authentication chap** command only with TACACS or extended TACACS.

	Command or Action	Purpose
	Example: `Device(config-if)# ppp authentication chap`	**Note** With AAA configured on the device 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	Do one of the following:
	• **ppp use-tacacs** [**single-line**]
	• **aaa authentication ppp**
	Example:
	`Device(config-if)# ppp use-tacacs single-line`
	Example:
	`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 and enters global configuration mode.
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 interfaces that a user will access.

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

Enabling Link Quality Monitoring

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM monitors the link quality. If the quality drops below a configured percentage, the router shuts down the link. The percentages are calculated for both incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the destination peer.

Note LQM is not compatible with Multilink PPP.

When LQM is enabled, every keepalive period is sent to Link Quality Reports (LQRs) in place of keepalives. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in RFC 1989, *PPP Link Quality Monitoring*.

Perform this task to enable LQM on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **ppp quality** *percentage*
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	**ppp quality** *percentage* **Example:** `Device(config-if)# ppp quality 10`	Enables LQM on an interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface 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 through lossless data compression. PPP encapsulations support both predictor and Stacker compression (STAC) algorithms.

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

	Command or Action	Purpose
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 and returns to privileged EXEC mode.

Configuring MPPC

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

Before You Begin

Ensure that PPP encapsulation is enabled before you configure MPPC.

Note 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 use the same compression method (STAC, Predictor, or MPPC).

SUMMARY STEPS

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

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Example

The following is sample output from the **debug ppp negotiation** command 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

Choosing the IP Address Assignment Method

The IP Address Pooling feature allows the 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.

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

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.

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.

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 define 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.

	Command or Action	Purpose
		Note You can provide as few as one or as many as ten DHCP servers for the proxy client (Cisco router or access server) to use. 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 and returns to privileged EXEC mode.

Defining 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.

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

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*]
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
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* | **default**} *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.

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Controlling DHCP Network Discovery

Perform this task to allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server by using PPP IP Control Protocol (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 the number of 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 value 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*
4. **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 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 specified number of DHCP Inform and Discover messages to be sent and a timeout period for retransmission to be configured.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring IP Address Assignment

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 one 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 by using SLIP or PPP.

Perform this task to define a nondefault address pool for use on an interface.

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*
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	**ip local pool** {*named-address-pool* | **default**} {*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.

	Command or Action	Purpose
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.

Step 8	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring PPP Reliable Link

PPP reliable link is Cisco's implementation of RFC 1663, *PPP Reliable Transmission*, which defines a method of negotiating and using Numbered Mode Link Access Procedure, Balanced (LAPB) to provide a reliable serial link. Numbered Mode LAPB provides retransmission of error packets across a serial link.

Although the LAPB protocol overhead consumes some bandwidth, you can offset the bandwidth consumption by the use of PPP compression over a reliable link. PPP compression is separately configurable and is not required for use by a reliable link.

Note PPP reliable link is available only on synchronous serial interfaces. PPP reliable link cannot be used over V.120 and does not work with Multilink PPP.

Perform this task to configure PPP reliable link on the specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ppp reliable-link**
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 2/0`	Specifies the interface and enters interface configuration mode.
Step 4	**ppp reliable-link** **Example:** `Device(config-if)# peer default ip address pool 2`	Enables PPP reliable link. **Note** Enabling reliable links does not guarantee that all connections through the specified interface will use the reliable link. It only guarantees that the device will attempt to negotiate reliable link on this interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the following commands:

- **debug lapb**
- **debug ppp negotiations**
- **debug ppp errors**
- **debug ppp packets**

You can determine whether LAPB has been established on a connection by using the **show interface** command.

Disabling or Reenabling Peer Neighbor Routes

Cisco 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. Perform this task to disable this default behavior or to reenable it after it has been disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no peer neighbor-route**
5. **peer neighbor-route**
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** *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 the creation of neighbor routes.
Step 5	**peer neighbor-route** **Example:** `Device(config-if)# peer neighbor-route`	Reenables the creation of neighbor routes. **Note** If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP

Configuring Multilink PPP on Synchronous Interfaces

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

SUMMARY STEPS

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

	Command or Action	Purpose
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.

Step 8	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Creating 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**
7. **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.

	Command or Action	Purpose
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.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Assigning an Interface to a Multilink Bundle

Caution Do not install a device to the peer address while configuring an MLPP lease line. This can be disabled using the **no ppp peer-neighbor-route** command under the MLPPP bundle interface.

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*
11. **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 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.

	Command or Action	Purpose
Step 7	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink 12`	Restricts a physical link to join 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.

Step 11	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP Using Multilink Group Interfaces

Multilink PPP 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 using the **multilink virtual template** command.

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

To configure Multilink PPP 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.

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/vci*
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.
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.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global 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.
Step 8	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 2`	Restricts a physical link to join only a designated multilink group interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global 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 and returns to privileged EXEC mode.

Configuring Multilink PPP Minimum Links Mandatory

Perform this task to configure the minimum number of links in a Multilink PPP bundle required to keep that bundle active.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ppp multilink**
4. **ppp multilink min-links** *links* **mandatory**
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	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 4	**ppp multilink min-links** *links* **mandatory** **Example:** `Device(config-if)# ppp multilink min-links 5 mandatory`	Specifies the required minimum number of links in a Multilink PPP bundle. • If the minimum number of links in an Multilink PPP bundle falls below the number specified by the *links* argument, the Multilink PPP bundle is disabled.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Changing the Default Endpoint Discriminator

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

Perform this task to override or change the default endpoint discriminator.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string** *char-string*}
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 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 that a system uses when negotiating the use of Multilink PPP with a peer.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP Interleaving and Queueing

Multilink PPP support for interleaving can be configured on virtual templates. To configure interleaving, complete the following tasks:

- Configure a virtual template.
- Configure Multilink PPP and interleaving on the interface or template.

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

Configuring Multilink PPP Interleaving

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)

Perform this task to configure Multilink PPP interleaving.

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*
10. **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.

	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 a Multilink PPP 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 a Multilink PPP 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 and enters global 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.

Step 10	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Disabling PPP Multilink Fragmentation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ppp multilink fragment disable**
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 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	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining PPP and Multilink PPP Interfaces

Perform this task to display Multilink PPP and Multichassis Multilink PPP 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 Multilink PPP and Multichassis Multilink PPP bundle information.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.

Configuration Examples for PPP and Multilink PPP

Example: Configuring Multilink PPP with Traffic Shaping

The following example shows how to configure Multilink PPP with traffic shaping and quality of service (QoS). In this example, two bundles with four links in each bundle are configured between two devices. The **ppp chap hostname** command entries are required for originating and terminating multiple bundles on a single pair of devices.

```
controller T3 0/3/1
 framing c-bit
 cablelength 224
 t1 1 channel-group 0 timeslots 1-24
 t1 2 channel-group 0 timeslots 1-24
 t1 3 channel-group 0 timeslots 1-24
 t1 4 channel-group 0 timeslots 1-24
 t1 5 channel-group 0 timeslots 1-24
 t1 6 channel-group 0 timeslots 1-24
 t1 7 channel-group 0 timeslots 1-24
 t1 8 channel-group 0 timeslots 1-24
!
class-map match-all DETERMINISTICOUT
  match ip precedence 3
class-map match-all VOICEVIDEOCONTROLOUT
  match ip precedence 2
```

```
class-map match-all VOICEOUT
  match ip precedence 1
class-map match-all ROUTINGPROTOCOLS
  match ip precedence 5
class-map match-all CONTROLLEDLOADOUT
  match ip precedence 4
!
policy-map QOS304QCHILD
 class VOICEOUT
    priority level 1
    police cir percent 30
 class VOICEVIDEOCONTROLOUT
    priority level 2
    police cir percent 5
 class DETERMINISTICOUT
    bandwidth remaining ratio 20
 class CONTROLLEDLOADOUT
    bandwidth remaining ratio 18
 class ROUTINGPROTOCOLS
    bandwidth remaining ratio 4
 class class-default
    bandwidth remaining ratio 22
policy-map ASRMLPPP6MBPARENT
 class class-default
    shape average percent 98
   service-policy QOS304QCHILD
!
interface Multilink1
 ip address 192.168.1.1 255.255.255.0
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
 service-policy output ASRMLPPP6MBPARENT
!
interface Multilink2
 ip address 192.168.2.1 255.255.255.0
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
 service-policy output ASRMLPPP6MBPARENT
!
interface serial 0/3/1/1:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/2:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/3:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/4:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
```

```
interface serial 0/3/1/5:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/6:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/7:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/8:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
```

Example: Enabling CHAP Authentication with an Encrypted Password

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

Configuration of Router yyy

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

Configuration of Router xxx

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

Configuration of Router zzz

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

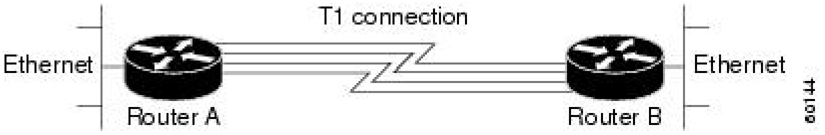
When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

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

Example: Configuring Multilink PPP on Synchronous Serial Interfaces

Multilink PPP provides characteristics most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. The figure below 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



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

Router A Configuration

```
hostname RouterA
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface virtual-template 1
 ip unnumbered Ethernet 0
 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 GigabitEthernet0/0/0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Router B Configuration

```
hostname RouterB
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface virtual-template 1
 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: Configuring Multilink PPP Using Multilink Group Interfaces over ATM

The following example shows how to configure Multilink PPP 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
 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: Configuring Multilink PPP Interleaving and Queueing for Real-Time Traffic

The following example shows how to define a virtual interface template that enables Multilink PPP interleaving with a maximum real-time traffic delay of 20 milliseconds and then applies the virtual template to the Multilink PPP 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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
PPP commands	Dial Technologies Command Reference
Wide-area networking commands	Wide-Area Networking Command Reference

MIBs

MIB	MIBs Link
No MIBs were introduced or modified for this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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 Media-Independent PPP and Multilink PPP

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 . An account on Cisco.com is not required.

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

Feature Name	Releases	Feature Information
Media-Independent PPP and Multilink PPP	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

Feature Name	Releases	Feature Information
Multilink PPP Minimum Links Mandatory	Cisco IOS XE Release 2.1	The Multilink PPP Minimum Links Mandatory feature enables you to configure the minimum number of links in a MLP bundle required to keep that bundle active. The following commands were introduced or modified: **multilink min-links**, **ppp multilink links minimum**.
DHCP Proxy Client	Cisco IOS XE Release 2.3	The DHCP proxy client feature allows you to manage a pool of IP addresses available to PPP or SLIP dial-in clients without a known IP address.