

# Configuring Media-Independent PPP and Multilink PPP

---

This chapter describes how to configure the Point-to-Point Protocol (PPP) and Multilink PPP features that can be configured on any interface. This chapter also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces.

See the “Configuring Asynchronous PPP and SLIP” chapter for information about PPP features and requirements that apply only to asynchronous lines and interfaces.

For a complete description of the PPP commands in this chapter, refer to the *Dial Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

## Implementation Information

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
- HSSI
- ISDN
- 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, the router might shut down a link if it detects a loop.

The software provides the Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) on serial interfaces running PPP encapsulation. For detailed information about authentication, see the *Security Configuration Guide*.

With Cisco IOS Release 11.2 F, Cisco now supports fast switching of incoming and outgoing DECnet and CLNS packets over PPP.

## PPP Configuration Task List

To configure PPP on a serial interface (including ISDN), perform the following task in interface configuration mode:

- Enable PPP Encapsulation

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:

- Enable CHAP or PAP Authentication
- Enable Link Quality Monitoring (LQM)
- Configure Compression of PPP Data
- Configure Microsoft Point-to-Point Compression (MPPC)
- Configure IP Address Pooling
- Configure PPP Reliable Link
- Disable or Reenable Peer Neighbor Routes
- Configure PPP Half-Bridging
- Configure Multilink PPP
- Configure MLP Interleaving and Queuing for Real-Time Traffic
- Monitor and Maintain PPP and MLP Interfaces

See the “PPP Configuration Examples” and the “MLP Interleaving and Queuing for Real-Time Traffic Examples” sections at the end of this chapter.

## Enable PPP Encapsulation

You can enable PPP on serial lines to encapsulate IP and other network protocol datagrams. To do so, use the following command in interface configuration mode:

Command	Purpose
<code>encapsulation ppp</code>	Enable PPP encapsulation.

## Enable CHAP or PAP Authentication

The Point-to-Point Protocol (PPP) with Challenge Handshake Authentication Protocol (CHAP) authentication or Password Authentication Protocol (PAP) 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 is 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 Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (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 host name of the local router.

The required response consists of two parts:

- An encrypted version of the ID, a secret password (or *secret*), and the random number
- Either the host name 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 by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

By transmitting this response, the secret is never transmitted 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 at the time 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. If the username and password specified in the authentication request are accepted, 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, you must perform the following tasks:

- Step 1** Enable PPP encapsulation.
- Step 2** Enable CHAP or PAP on the interface.
- Step 3** For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
<code>encapsulation ppp</code>	Enable PPP on an interface.

## Enable Link Quality Monitoring (LQM)

---

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
<b>ppp authentication</b> {chap   chap pap   pap chap   pap} [if-needed] [list-name   default] [callin]	Define the authentication methods supported and the order in which they are used.

The **ppp authentication chap** optional keyword **if-needed** can be used only with TACACS or extended TACACS.

With authentication, authorization, and accounting (AAA) configured on the router and list names defined for AAA, the optional keyword *list-name* can be used with AAA/TACACS+.



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

Add a **username** entry for each remote system from which the local router or access server requires authentication.

To specify the password to be used in CHAP or PAP caller identification, use the following command in global configuration mode:

Command	Purpose
<b>username</b> name password secret	Configure identification.

Make sure this password does not include spaces or underscores.

To configure Terminal Access Controller Access Control System (TACACS) on a specific interface as an alternative to global host authentication, use the following command in interface configuration mode:

Command	Purpose
<b>ppp use-tacacs</b> [single-line] or <b>aaa authentication ppp</b>	Configure TACACS.

Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

For an example of CHAP, see the section “CHAP with an Encrypted Password Examples” at the end of this chapter. CHAP is specified in RFC 1994, “PPP Challenge Handshake Authentication Protocol (CHAP).”

## Enable Link Quality Monitoring (LQM)

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router shuts down the link. The percentages are calculated for both the 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.

When LQM is enabled, Link Quality Reports (LQRs) are sent, in place of keepalives, every keepalive period. 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," by William A. Simpson of Computer Systems Consulting Services.

To enable LQM on the interface, use the following command in interface configuration mode:

Command	Purpose
<code>ppp quality percentage</code>	Enable LQM on the interface.

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and taken down.

## Configure 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 the majority of your traffic is already compressed files, do not use compression.

Most routers support software compression only, but in the Cisco 7000 series hardware compression and distributed compression are also available, depending on the interface processor and compression service adapter hardware installed in the router.

To configure compression, complete the tasks in one of the following sections:

- Software Compression
- Hardware-Dependent Compression

### Software Compression

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

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

To configure compression over PPP, use the following commands in interface configuration mode:

Step	Command	Purpose
1	<code>encapsulation ppp</code>	Enable encapsulation of a single protocol on the serial line.
2	<code>compress [predictor   stac   mppc [ignore-pfc]]</code>	Enable compression.

### Hardware-Dependent Compression

When you configure Stacker compression on Cisco 7000 series routers with RSP7000, on Cisco 7200 series routers, and on Cisco 7500 series routers, there are three methods of compression: hardware compression, distributed compression, and software compression.

Hardware and distributed compression are available on routers that have the SA-Comp/1 and SA-Comp/4 data compression service adapters (CSAs). CSAs are available on Cisco 7200 series routers, on Cisco 7500 series routers with second-generation Versatile Interface Processors (VIP2s), and on Cisco 7000 series routers with the 7000 Series Route Switch Processor (RSP7000) and 7000 Series Chassis Interface (RSP7000CI). (CSAs require VIP2 model VIP2-40.)

To configure hardware or distributed compression over PPP, use the following commands in interface configuration mode:

Step	Command	Purpose
1	<b>encapsulation ppp</b>	Enable encapsulation of a single protocol on the serial line.
2	<b>compress stac</b> [ <b>distributed</b>   <b>software</b> ] (Cisco 7000 series with RSP7000 and Cisco 7500 series) or <b>compress stac</b> [ <b>csa slot</b>   <b>software</b> ] (Cisco 7200 series)	Enable compression.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method:

- If the router contains a compression service adapter (CSA), compression is performed in the CSA hardware (hardware compression).
- If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression).
- If the VIP2 is not available, compression is performed in the router's main processor (software compression).

Using hardware compression in the CSA frees the router's main processor for other tasks. You can also configure the router to use the VIP2 to perform compression by using the **distributed** option, or to use the router's main processor by using the **software** option. If the VIP2 is not available, compression is performed in the router's main processor.

When compression is performed in software installed in the router's main processor, it might significantly affect system performance. We recommend that you disable compression in the router's main processor if the router CPU load exceeds 40 percent. To display the CPU load, use the **show process cpu EXEC** command.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method.

## Configure Microsoft Point-to-Point Compression (MPPC)

Microsoft Point-to-Point Compression (MPPC) is a scheme used to compress Point-to-Point Protocol (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 transmitted over PPP is the same as the MTU of the PPP interface; however this

length cannot be greater than 8192 bytes since 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 12 bit coherency count. If the decompressor detects that the coherency count is out of sequence, then the error recovery process begins and is described through the following steps:

- 1 Reset Request (RR) packet 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 that can be additionally time-consuming.

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

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

## MPPC Requirements

The following restrictions apply to the MPPC feature:

- MPPC is only supported 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).

## Configure MPPC

PPP encapsulation must be enabled before you can configure MPPC. For information on how to configure PPP encapsulation refer to the “Enable PPP Encapsulation” section outlined in the beginning of this chapter.

There is only one command required to configure MPPC. The existing **compress** command now supports the **mppc** keyword, which prepares the interface to initiate CCP and negotiates MPPC with the Microsoft client. Once PPP encapsulation is configured on the router, perform the following task in interface configuration mode:

Command	Purpose
<b>compress [mppc [ignore-pfc]]</b>	Enable MPPC on the interface.

The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by 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 which use an uncompressed protocol field (0x0021), even though the pfc is negotiated between peers. Figure 274 displays protocol rejections when the **debug ppp negotiation** command is enabled. These errors can be remedied by setting the **ignore-pfc** option.

**Figure 274** Sample Debug PPP Negotiation 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
```

## Configure IP Address Pooling

Point-to-point interfaces 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 or provided by TACACS+, 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 the IP Control Protocol (IPCP) address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

See the “Configuring Asynchronous PPP and SLIP” chapter for additional information about address pooling on asynchronous interfaces and about SLIP.

## 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 SLIP EXEC command—An asynchronous dial-up user can enter a peer IP address or host name 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—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 dial-up 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 dial-up 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.
- VTY/Protocol translation—The translate command can define the peer IP address for a VTY (pseudo asynchronous interface).

The pool configured for the interface is used, unless TACACS+ returns a pool name as part of 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)

## Interfaces Affected

Address pooling is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces running the Point-to-Point Protocol (PPP).

## Choose the IP Address Assignment Method

The IP address pooling feature now 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.

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

- Define the Global Default Address Pooling Mechanism
- Configure Per-Interface IP Address Assignment, as needed

## Define 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 one of following sections:

- Define DHCP as the Global Default Mechanism

- Define Local Address Pooling as the Global Default Mechanism

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.

### Define DHCP as the Global Default Mechanism

The Dynamic Host Configuration Protocol (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.

To enable DHCP as the global default mechanism, use the following commands in global configuration mode:

Step	Command	Purpose
1	<b>ip address-pool dhcp-proxy-client</b>	Specify DHCP client-proxy as the global default mechanism.
2	<b>ip dhcp-server</b> [ <i>ip-address</i>   <i>name</i> ]	(Optional) Specify the IP address of a DHCP server for the proxy client to use.

In Step 2, 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. DHCP servers provide temporary IP addresses.

### Define Local Address Pooling as the Global Default Mechanism

To specify that the global default mechanism to use is local pooling, use the following commands in global configuration mode:

Step	Command	Purpose
1	<b>ip address-pool local</b>	Specify local pooling as the global default mechanism.
2	<b>ip local pool</b> { <b>default</b>   <i>pool-name</i> } <i>low-ip-address</i> [ <i>high-ip-address</i> ]	Create one or more local IP address pools.

If no other pool is defined, the local pool called *default* is used.

## Configure Per-Interface IP Address Assignment

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can then 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 Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP).

To define a nondefault address pool for use on an interface, use the following commands beginning in global configuration mode:

Command	Purpose
<b>ip local pool</b> <i>poolname</i> { <i>low-ip-address</i> [ <i>high-ip-address</i> ]}	Create one or more local IP address pools.
<b>interface</b> <i>type number</i>	Specify the interface and enter interface configuration mode.
<b>peer default ip address pool</b> <i>pool-name</i>	Specify the pool for the interface to use.

To define DHCP as the IP address mechanism for an interface, use the following commands beginning in global configuration mode:

Command	Purpose
<b>interface</b> <i>type number</i>	Specify the interface and enter interface configuration mode.
<b>peer default ip address pool dhcp</b>	Specify DHCP as the IP address mechanism on this interface.

To define a specific IP address to be assigned to all dial-in peers on an interface, use the following commands beginning in global configuration mode:

Command	Purpose
<b>interface</b> <i>type number</i>	Specify the interface and enter interface configuration mode.
<b>peer default ip address</b> <i>ip-address</i>	Specify the IP address to assign.

## Configure 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 LAPB to provide a reliable serial link. Numbered Mode LAPB provides retransmission of errored packets across the serial link.

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

PPP reliable link is available only on synchronous serial interfaces, including ISDN BRI and ISDN PRI interfaces. PPP reliable link cannot be used over V.120.

To configure PPP reliable link on a specified interface, use the following commands in interface configuration mode:

Command	Purpose
<b>ppp reliable-link</b>	Enable PPP reliable link.

Having reliable link enabled does not guarantee that all connections through the specified interface will in fact use reliable link. It only guarantees that the router will attempt to negotiate reliable link on this interface.

### Restrictions

PPP reliable link does not work with Multilink PPP.

PPP reliable link is not available on asynchronous serial interfaces, including ISDN BRI and ISDN PRI interfaces. PPP reliable link cannot be used over V.120.

### Troubleshooting

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 LAPB has been established on a connection by using the **show interface** command.

## Disable or Reenable 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, use the following commands in interface configuration mode:

Command	Purpose
<b>no peer neighbor-route</b>	Disable creation of neighbor routes.
<b>peer neighbor-route</b>	Reenable creation of neighbor routes.

---

**Note** If entered on a dialer or async-group interface, this command affects all member interfaces.

---

## Configure PPP Half-Bridging

For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serial or ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface, and the router's serial or ISDN interface functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router processes. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the same Ethernet subnetwork.

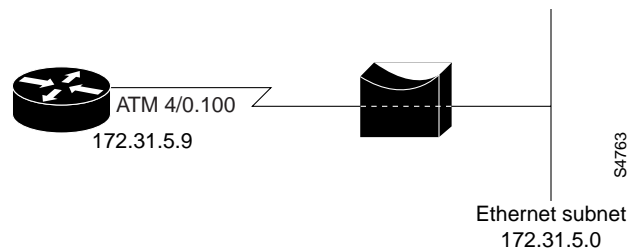
---

**Note** An interface cannot function as both a half-bridge and a bridge.

---

Figure 275 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure 275 Router Serial Interface Configured as a Half-Bridge



**Note** The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.

To configure a serial interface to function as a half-bridge, complete the following tasks beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface serial</b> <i>number</i>	Specify the interface (and enter interface configuration mode).
2	<b>ppp bridge appletalk</b> <b>ppp bridge ip</b> <b>ppp bridge ipx</b> [ <i>novell-ether</i>   <i>arpa</i>   <i>sap</i>   <i>snap</i> ]	Enable PPP half-bridging for one or more routed protocols: AppleTalk, IP, or IPX.
3	<b>ip address</b> <i>n.n.n.n</i> <b>appletalk address</b> <i>network.node</i> <b>appletalk cable-range</b> <i>cable-range network.node</i> <b>ipx network</b> <i>network</i>	Provide a protocol address on the same subnetwork as the remote network.

**Note** You must enter the **ppp bridge** command either when the interface is shut down or before you provide a protocol address for the interface.

For more information about AppleTalk addressing see the “Configuring AppleTalk” chapter; for more information about IPX addresses and encapsulations, see the “Configuring Novell IPX” chapter. Both chapters are in the *Network Protocols Configuration Guide, Part 2*.

## Configure Multilink PPP

The Multilink Point-to-Point Protocol (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. Cisco’s implementation of Multilink PPP supports the fragmentation and packet sequencing specifications in RFC 1717.

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 dialer load threshold that you define. 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.

Multilink PPP is designed to work over single or multiple interfaces of the following types that are configured to support both dial-on-demand rotary groups and PPP encapsulation:

- Synchronous serial interfaces
- Asynchronous serial interfaces
- Basic Rate Interfaces (BRIs)
- Primary Rate Interfaces (PRIs)

### Configure Multilink PPP on Synchronous Interfaces

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

To configure a synchronous interface, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface serial</b> <i>number</i>	Specify an asynchronous interface.
2	<b>no ip address</b>	Specify no IP address for the interface.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>no fair-queue</b>	Disable WFQ on the interface.
5	<b>ppp multilink</b>	Enable Multilink PPP.
6	<b>pulse-time</b> <i>seconds</i>	Enable pulsing DTR signal intervals on the interface.

Repeat these steps for additional synchronous interfaces, as needed.

### Configure Multilink PPP on Asynchronous Interfaces

To configure Multilink PPP on asynchronous interfaces, you configure the asynchronous interfaces to support DDR and PPP encapsulation, then you configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

To configure an asynchronous interface to support DDR and PPP encapsulation, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface async</b> <i>number</i>	Specify an asynchronous interface.
2	<b>no ip address</b>	Specify no IP address for the interface.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>dialer in-band</b>	Enable DDR on the interface.
5	<b>dialer rotary-group</b> <i>number</i>	Include the interface in a specific dialer rotary group.

Repeat this step for additional asynchronous interfaces, as needed.

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

To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface dialer</b> <i>number</i>	Define a dialer rotary group.
2	<b>no ip address</b>	Specify no IP address for the interface.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>dialer in-band</b>	Enable DDR on the interface.
5	<b>dialer load-threshold</b> <i>load</i> [ <b>inbound</b>   <b>outbound</b>   <b>either</b> ]	Configure bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
6	<b>ppp multilink</b>	Enable Multilink PPP.

## Configure Multilink PPP on a Single ISDN BRI Interface

To enable Multilink PPP on a single Integrated Services Digital Network (ISDN) BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

To enable PPP on an ISDN BRI interface, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface bri</b> <i>number</i>	Specify an interface.
2	<b>ip address</b> <i>ip-address mask</i> [ <b>secondary</b> ]	Provide an appropriate protocol address for the interface.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>dialer idle-timeout</b> <i>seconds</i>	(Optional) Specify a dialer idle timeout.
5	<b>dialer load-threshold</b> <i>load</i>	Specify the dialer load threshold for bringing up additional WAN links.
6	<b>dialer map</b> <i>protocol next-hop-address</i> [ <b>name</b> <i>hostname</i> ] [ <b>spc</b> ] [ <b>speed</b> <b>56</b>   <b>64</b> ] [ <b>broadcast</b> ] [ <i>dial-string[:isdn-subaddress]</i> ]	Configure the ISDN interface to call the remote site.
7	<b>dialer-group</b> <i>group-number</i>	Control access to this interface by adding it to a dialer access group.
8	<b>ppp authentication pap</b>	(Optional) Enable PPP authentication.
9	<b>ppp multilink</b>	Enable Multilink PPP on the dialer rotary group

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 Multilink PPP on a single ISDN BRI interface, see the “Multilink PPP on One ISDN Interface Example” section later in this chapter.

When Multilink PPP 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 no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer-load threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

When you use dialer profiles with Multilink PPP, you must specify the **ppp multilink** command on both the physical and logical interfaces in order for Multilink PPP to be negotiated by the router. See the section “Multilink PPP with Dialer Profiles Example” for more information.

## Configure Multilink PPP on Multiple ISDN BRI Interfaces

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

To set up the dialer rotary interface for the BRI interfaces, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface dialer</b> <i>number</i>	Specify the dialer rotary interface.
2	<b>ip address</b> <i>address mask</i>	Specify the protocol address for the dialer rotary interface.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>dialer in-band</b>	Specify in-band dialing.
5	<b>dialer idle-timeout</b> <i>seconds</i>	(Optional) Specify the dialer idle timeout period, using the same timeout period as the individual BRI interfaces.
6	<b>dialer map</b> <i>protocol next-hop-address</i> [ <i>name hostname</i> ] [ <i>spc</i> ] [ <i>speed 56   64</i> ] [ <i>broadcast</i> ] [ <i>dial-string[:isdn-subaddress]</i> ]	Map the next-hop protocol address and name to the dial string needed to reach it.
7	<b>dialer load-threshold</b> <i>load</i>	Specify the dialer load threshold, using the same threshold as the individual BRI interfaces.
8	<b>dialer-group</b> <i>number</i>	Control access to this interface by adding it to a dialer access group.
9	<b>ppp authentication chap</b>	(Optional) Enable PPP Challenge Handshake Authentication Protocol (CHAP) authentication.
10	<b>ppp multilink</b>	Enable Multilink PPP.

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

To configure each of the BRI interfaces to belong to the same rotary group, use the following commands beginning in global configuration mode:

Step	Command	Purpose
1	<b>interface bri</b> <i>number</i>	Specify one of the BRI interfaces.
2	<b>no ip address</b>	Specify that it does not have an individual protocol address.
3	<b>encapsulation ppp</b>	Enable PPP encapsulation.
4	<b>dialer idle-timeout</b> <i>seconds</i>	Set the dialer idle timeout period, using the same timeout for each of the BRI interfaces you configure.
5	<b>dialer rotary-group</b> <i>number</i>	Add the interface to the rotary group.
6	<b>dialer load-threshold</b> <i>load</i>	Specify the dialer load threshold for bringing up additional WAN links.

Repeat Steps 1 through 6 for each BRI you want to belong to the same dialer rotary group.

For an example of configuring Multilink PPP on multiple ISDN BRI interfaces, see the “Multiple ISDN Interfaces Configured for Multilink PPP Example” section later in this chapter.

When Multilink PPP 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 no longer keeps a multilink bundle of  $n$  links connected indefinitely and the **dialer load-threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

## Configure MLP Interleaving and Queuing for Real-Time Traffic

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 transmitted 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 transmitted earlier than other flows.

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

Fair-queuing on Multilink PPP overcomes a prior restriction. Previously, fair-queuing 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.

## Restrictions

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

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

## MLP Interleaving Configuration Tasks

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

- Step 1** Configure the dialer interface, BRI interface, PRI interface, or virtual template, as defined in the relevant chapters of this manual.
- Step 2** Configure Multilink PPP and interleaving on the interface or template.

---

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

---

To configure Multilink PPP and interleaving on a configured and operational interface or virtual interface template, use the following commands beginning in interface configuration mode:

Step	Command	Purpose
1	<b>ppp multilink</b>	Enable Multilink PPP.
2	<b>ppp multilink interleave</b>	Enable real-time packet interleaving.
3	<b>ppp multilink fragment-delay</b> <i>milliseconds</i>	Optionally, configure a maximum fragment delay.
4	<b>ip rtp reserve</b> <i>lowest-udp-port</i> <i>range-of-ports</i> [ <i>maximum-bandwidth</i> ]	Reserve a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.
5	<b>exit</b>	Exits interface configuration mode.
6	<b>multilink virtual-template 1</b>	For virtual templates only, apply the virtual template to the multilink bundle. <sup>1</sup>

1 This step is not used for ISDN or dialer interfaces.

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)
```

## Monitor and Maintain PPP and MLP Interfaces

To monitor and maintain virtual interfaces, you can use any of the following commands:

Command	Purpose
<b>show ppp multilink</b>	Display MLP and MMP bundle information.

## PPP Configuration Examples

The examples provided in this section show various PPP configurations as follows:

- CHAP with an Encrypted Password Examples
- MPPC Interface Configuration Examples
- PPP Reliable Link Examples
- Multilink PPP Examples
- MLP Interleaving and Queuing for Real-Time Traffic Examples

## CHAP with an Encrypted Password Examples

The following configuration examples enable CHAP on serial interface 0 of three devices.

### Configuration of Router yyy

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

### Configuration of Router xxx

```
hostname xxx
interface serial 0
  encapsulation ppp
  ppp authentication chap

username yyy password secretxy
username zzz password secretxz
```

### Configuration of Router zzz

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

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
  encapsulation ppp
  ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

## MPPC Interface Configuration Examples

The following example configures 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
```

## PPP Reliable Link Examples

The following example enables PPP reliable link and Stac compression on BRI 0:

```
interface BRI0
 description Enables stac compression on BRI 0
 ip address 172.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 172.1.1.2 name baseball 14195386368
 compress stac
 ppp authentication chap
 dialer-group 1
 ppp reliable-link
```

The following example shows output of the **show interfaces** command when PPP reliable link is enabled. The LAPB output lines indicate that PPP reliable link is provided over LAPB.

```
Router# show interface serial 0
Serial0 is up, line protocol is up
Hardware is HD64570
Description: connects to enkidu s 0
Internet address is 172.21.10.10/8
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set
LCP Open
Open: IPCP, CDP
LAPB DTE, state CONNECT, modulo 8, k 7, N1 12048, N2 20
  T1 3000, T2 0, interface outage (partial T3) 0, T4 0, PPP over LAPB
  VS 1, VR 1, tx NR 1, Remote VR 1, Retransmissions 0
  Queues: U/S frames 0, I frames 0, unack. 0, reTx 0
  IFRAMES 1017/1017 RNRs 0/0 REJs 0/0 SABM/Es 1/1 FRMRs 0/0 DISCs 0/0
Last input 00:00:18, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/64/0 (size/threshold/drops)
  Conversations 0/1 (active/max active)
  Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 3000 bits/sec, 4 packets/sec
5 minute output rate 3000 bits/sec, 7 packets/sec
  1365 packets input, 107665 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  2064 packets output, 109207 bytes, 0 underruns
  0 output errors, 0 collisions, 4 interface resets
  0 output buffer failures, 0 output buffers swapped out
  4 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

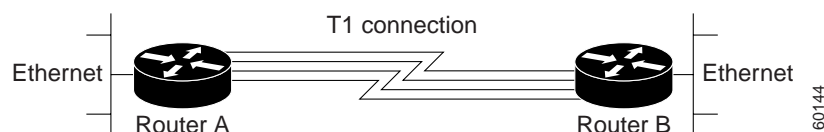
## Multilink PPP Examples

The following examples configure Multilink PPP on synchronous serial interfaces, on one BRI interface, and on multiple BRIs belonging to the same dialer rotary group that is configured for Multilink PPP.

### Multilink PPP on Synchronous Serial Interfaces Example

Multilink PPP provides characteristics most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 276 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

**Figure 276 Inverse Multiplexing Application Using Multilink PPP**



The following example shows the configuration commands 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-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
```

## Router B Configuration

```
hostname RouterB
!
!
username RouterB 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
```

### Multilink PPP with Dialer Profiles Example

The following example indicates how to configure Multilink PPP with dialer profiles: both the physical and logical interfaces must have the **ppp multilink** command applied to it.

```
!  
interface BRI2/1  
  no ip address  
  encapsulation ppp  
  dialer pool-member 2  
  no fair-queue  
  no cdp enable  
  ppp authentication chap  
  ppp multilink  
!  
!  
interface Dialer2  
  ip address 10.1.1.1 255.255.255.0  
  encapsulation ppp  
  dialer remote-name bavik  
  dialer string 229  
  dialer pool 2  
  dialer-group 1  
  no fair-queue  
  ppp authentication chap  
  ppp multilink  
!
```

### Multilink PPP on One ISDN Interface Example

The following example enables Multilink PPP on the BRI interface 0. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0  
  description connected to ntt 81012345678902  
  ip address 171.1.1.7 255.255.255.0  
  encapsulation ppp  
  dialer idle-timeout 30  
  dialer load-threshold 40 either  
  dialer map ip 171.1.1.8 name atlanta 81012345678901  
  dialer-group 1  
  ppp authentication pap  
  ppp multilink
```

## Multilink PPP on Multiple ISDN Interfaces Example

The following example configures 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 BRI0
  no ip address
  encapsulation ppp
  dialer idle-timeout 500
  dialer rotary-group 0
  dialer load-threshold 30 either
!
interface BRI1
  no ip address
  encapsulation ppp
  dialer idle-timeout 500
  dialer rotary-group 0
  dialer load-threshold 30 either
!
interface BRI2
  no ip address
  encapsulation ppp
  dialer idle-timeout 500
  dialer rotary-group 0
  dialer load-threshold 30 either
!
interface Dialer0
  ip address 99.0.0.2 255.0.0.0
  encapsulation ppp
  dialer in-band
  dialer idle-timeout 500
  dialer map ip 99.0.0.1 name atlanta broadcast 81012345678901
  dialer load-threshold 30 either
  dialer-group 1
  ppp authentication chap
  ppp multilink
```

## MLP Interleaving and Queuing for Real-Time Traffic Examples

The following example defines a virtual interface template that enables Multilink PPP interleaving and a maximum real-time traffic delay of 20 milliseconds, and then applies that 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
```

The following example enables Multilink PPP 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 8.1.1.1 255.255.255.0
  encapsulation ppp
  dialer map ip 8.1.1.2 name angus 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
```