

# Asynchronous PPP and SLIP Commands

---

Point-to-Point Protocol (PPP) and Serial Line Internet Protocol (SLIP) define methods of sending IP packets over standard EIA/TIA-232 asynchronous serial lines with minimum line speeds of 1200 baud. This chapter describes the commands used to configure your router to enable PPP and SLIP on asynchronous interfaces.

Using PPP or SLIP encapsulation over asynchronous lines is an inexpensive way of connecting PCs to a network. PPP and SLIP over asynchronous dial-up modems allow a home computer to be connected to a network without the cost of a leased line. Dial-up PPP and SLIP links can also be used for remote sites that need only occasional remote node or backup connectivity. Both public-domain and vendor-supported PPP and SLIP implementations are available for a variety of computer applications.

Use the commands in this chapter to configure PPP and SLIP on your router. For configuration information and examples, refer to the chapter “Configuring Asynchronous PPP and SLIP” in the *Dial Solutions Configuration Guide*.

---

**Note** Some commands previously documented in this chapter have been replaced by new commands. Although these commands continue to perform their normal functions in the current release, support for these commands will cease in future releases.

---

## async mode dedicated

To place a line into dedicated asynchronous mode using SLIP or PPP encapsulation, use the **async mode dedicated** interface configuration command. To return the line to interactive mode, use the **no** form of this command.

```
async mode dedicated  
no async mode dedicated
```

### Syntax Description

This command has no arguments or keywords.

### Default

Asynchronous mode is disabled.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

With dedicated asynchronous network mode, the interface will use either SLIP or PPP encapsulation, depending on which encapsulation method is configured for the interface. An EXEC prompt does not appear, and the router is not available for normal interactive use.

If you configure a line for dedicated mode, you will not be able to use the **async dynamic address** command, because there is no user prompt.

### Example

The following example assigns an IP address to an asynchronous line and places the line into network mode. Setting the stop bits to 1 enhances performance.

```
interface async 4  
  async default ip address 172.31.7.51  
  async mode dedicated  
  encapsulation slip  
  
line 20  
  location Joe's computer  
  stopbits 1  
  speed 115200
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**async mode interactive**

## async mode interactive

To return a line that has been placed into dedicated asynchronous network mode to interactive mode, thereby enabling the **slip** and **ppp** EXEC commands, use the **async mode interactive** interface configuration command. To prevent users from implementing SLIP and PPP at the EXEC level, use the **no** form of this command.

```
async mode interactive  
no async mode interactive
```

### Syntax Description

This command has no arguments or keywords.

### Default

Asynchronous mode is disabled.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Interactive mode enables the **slip** and **ppp** EXEC commands. In dedicated mode, there is no user EXEC level. The user does not enter any commands, and a connection is automatically established when the user logs in, according to the configuration.

### Example

The following example places async interface 6 into interactive asynchronous mode:

```
interface async 6  
  async default ip address 172.31.7.51  
  async mode interactive  
  ip unnumbered ethernet 0
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**async mode dedicated**

## autoselect

To configure a line to start an ARA, PPP, or SLIP session, use the **autoselect** line configuration command. Use the **no** form of this command to disable this function on a line.

**autoselect** { **arap** | **ppp** | **slip** | **during-login** }  
**no autoselect**

### Syntax Description

<b>arap</b>	Configures the Cisco IOS software to allow an ARA session to start up automatically.
<b>ppp</b>	Configures the Cisco IOS software to allow a PPP session to start up automatically.
<b>slip</b>	Configures the Cisco IOS software to allow a SLIP session to start up automatically.
<b>during-login</b>	The username and/or password prompt is displayed without pressing the Return key. After the user logs in, the autoselect function begins.

### Default

ARA session

### Command Mode

Line configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3. The **no autoselect** command and the **during-login** command first appeared in Cisco IOS Release 11.0.

This command eliminates the need for users to enter an EXEC command to start an ARA, PPP, or SLIP session.

---

**Note** SLIP does not support authentication. For PPP and ARAP, you must enable authentication.

---

The **autoselect** command configures the Cisco IOS software to identify the type of connection being requested. For example, when a user on a Macintosh running ARA selects the Connect button, the Cisco IOS software automatically starts an ARAP session. If, on the other hand, the user is running SLIP or PPP and uses the **autoselect ppp** or **autoselect slip** command, the Cisco IOS software automatically starts a PPP or SLIP session, respectively. This command is used on lines making different types of connections.

A line that does not have **autoselect** configured views an attempt to open a connection as noise. The router does not respond and the user client times out.

---

**Note** After the modem connection is established, a Return is required to evoke a response, such as to get the username prompt. You might need to update your scripts to include this requirement. Additionally, the activation character should be set to the default and the exec-character-bits set to 7. If you change these defaults, the application cannot recognize the activation request.

---

## Examples

The following example enables ARA on a line:

```
line 3
  arap enable
  autoselect arap
```

The following example enables PPP on a line:

```
line 7
  autoselect ppp
```

The following example enables ARA on a line and allows logins from users with a modified CCL script and an unmodified script to log in:

```
line 3
  arap enable
  autoselect arap
  autoselect during-login
  arap nolog if-needed
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**arap use-tacacs**  
**ppp authentication chap**  
**ppp authentication pap**  
**ppp use-tacacs**

## encapsulation

To configure SLIP or PPP encapsulation as the default on an asynchronous interface, use the **encapsulation** interface configuration command. To disable encapsulation, use the **no** form of this command.

```
encapsulation {slip | ppp}  
no encapsulation {slip | ppp}
```

### Syntax Description

<b>slip</b>	Specifies SLIP encapsulation for an interface configured for dedicated asynchronous mode or DDR.
<b>ppp</b>	Specifies PPP encapsulation for an interface configured for dedicated asynchronous mode or dial-on-demand routing (DDR).

### Default

SLIP encapsulation is enabled by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

On lines configured for interactive use, encapsulation is selected by the user when they establish a connection with the **slip** or **ppp EXEC** command.

IP Control Protocol (IPCP) is the part of PPP that brings up and configures IP links. After devices at both ends of a connection communicate and bring up PPP, they bring up the control protocol for each network protocol that they intend to run over the PPP link such as IP or IPX. If you have problems passing IP packets and the **show interface** command shows that line is up, use the **negotiations** command to see if and where the negotiations are failing. You might have different versions of software running, or different versions of PPP, in which case you might need to upgrade your software or turn off PPP option negotiations. All IPCP options as listed in RFC 1332 are supported on asynchronous lines. Only Option 2, TCP/IP header compression, is supported on synchronous interfaces.

PPP echo requests are used as keepalive packets to detect line failure. The **no keepalive** command can be used to disable echo requests. For more information about the **no keepalive** command, refer to the chapter “IP Services Commands” in the *Networking Protocols Command Reference, Part 1* and the chapter “Configuring IP Services” in the *Networking Protocols Configuration Guide, Part 1*.

In order to use SLIP or PPP, the Cisco IOS software must be configured with an IP routing protocol or with the **ip host-routing** command. This configuration is done automatically if you are using old-style **slip address** commands. However, you must configure it manually if you configure SLIP or PPP via the **interface async** command.

---

**Note** Disable software flow control on SLIP and PPP lines.

---

## Example

In the following example, async interface 1 is configured for PPP encapsulation.

```
router# config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
router(config)# interface async 1
router(config-if)# encapsulation ppp
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**keepalive**

## ip access-group

To configure an access list to be used for packets transmitted to and from the asynchronous host, use the **ip access-group** interface configuration command. To disable control over packets transmitted to or from an asynchronous host, use the **no** form of this command.

```
ip access-group access-list-number { in | out }  
no ip access-group access-list-number
```

### Syntax Description

<i>access-list-number</i>	Assigned IP access list number.
<b>in</b>	Defines access control on packets transmitted <i>from</i> the asynchronous host.
<b>out</b>	Defines access control on packets being sent <i>to</i> the asynchronous host.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

With this command in effect, the IP destination address of each packet is run through the access list for acceptability then dropped or passed.

### Example

The following example assumes that users are restricted to certain servers designated as SLIP or PPP servers, but that normal terminal users can access anything on the local network:

```
! access list for normal connections  
access-list 1 permit 172.16.0.0 0.0.255.255  
!  
! access list for SLIP packets.  
access-list 2 permit 172.16.42.55  
access-list 2 permit 172.16.111.1  
access-list 2 permit 172.16.55.99  
!  
! Specify the access list  
interface async 6  
  async dynamic address  
  ip access-group 1 out  
  ip access-group 2 in
```

## ip address

To set IP addresses for an interface, use the **ip address** interface configuration command. To remove the specified addresses, use the **no** form of this command.

```
ip address address mask [secondary]  
no ip address address mask [secondary]
```

### Syntax Description

<i>address</i>	IP address.
<i>mask</i>	Network mask for the associated IP network.
<b>secondary</b>	(Optional) Specifies additional IP addresses.

### Default

No IP addresses are specified.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

The subnet mask must be the same for all interfaces connected to subnets of the same network. Hosts can determine subnet masks using the Internet Control Message Protocol (ICMP) *Mask Request* message. The Cisco IOS software responds to this request with an ICMP *Mask Reply* message.

You can disable IP processing on a particular interface by removing its IP address with the **no ip address** interface configuration command. If the router detects another host using one of its IP addresses, it will print an error message on the console.

### Example

In the example that follows, 172.16.1.27 is the primary address and 192.168.7.17 and 192.168.8.17 are secondary addresses for async interface 1:

```
interface async 1  
ip address 172.16.1.27 255.255.255.0  
ip address 192.168.7.17 255.255.255.0 secondary  
ip address 192.168.8.17 255.255.255.0 secondary
```

## ip tcp async-mobility server

To enable asynchronous listening, which in turn allows TCP connections to TCP port 57, enter the **ip tcp async-mobility server** global configuration command. Enter the **no** form of this command to turn listening off.

```
ip tcp async-mobility server  
no ip tcp async-mobility server
```

### Syntax

This command has no keywords or arguments.

### Default

Disabled. Asynchronous listening is turned off.

### Mode

Global Configuration

### Usage

This command first appeared in Cisco IOS Release 11.2.

After asynchronous listening is turned on by the **ip tcp async-mobility server** command, enter the **tunnel host** command to establish a network layer connection to a remote host. Both commands must be used to enable asynchronous mobility.

### Example

The following example shows how to configure asynchronous mobility. The **tunnel host** command is used to establish a network layer connection with an IBM host called mktg.

```
5300# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
5300(config)# ip tcp async-mobility server  
5300(config)# exit  
5300#  
%SYS-5-CONFIG_I: Configured from console by console  
5300# tunnel ?  
WORD Address or hostname of a remote system  
  
5300# tunnel mktg
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**tunnel**

## ip tcp header-compression

To configure TCP header compression on the asynchronous link, use the **ip tcp header-compression** interface configuration command. To disable header compression, use the **no** form of this command.

```
ip tcp header-compression [on | off | passive]
no ip tcp header-compression
```

### Syntax Description

<b>on</b>	(Optional) Turns header compression on.
<b>off</b>	(Optional) Turns header compression off.
<b>passive</b>	(Optional) On SLIP lines, prevents transmission of compressed packets until a compressed packet arrives from the asynchronous link, unless a user specifies SLIP on the command line. For PPP, this option functions the same as the <b>on</b> option.

### Default

Header compression is on.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Header compression data areas are initialized to handle up to 16 simultaneous TCP connections. Currently, you cannot change this number. You can only turn header compression on or off or use the **passive** keyword.

On lines configured for PPP encapsulation, the keywords **passive** and **on** cause the same behavior because, before attempting header compression, PPP automatically negotiates whether compression is available at each end of the connection.

There are two ways to implement header compression when the line is configured for **ip tcp header-compression passive**:

- The user enters the **/compressed** option with the **slip EXEC** commands to force the line into compressed mode. This overrides the passive setting and causes the interface to behave as if header compression is enabled.
- The user enters **slip** or **slip default** and the connecting system sends compressed packets to the server. The server detects the use of compression by the connecting system and automatically enters compressed mode.

If a line is configured for passive header compression and you use the **slip** or **ppp EXEC** command to enter asynchronous mode, you will see that the interface is set to match the compression status used by the host at the other end of the asynchronous line.

```
router> slip 10.0.0.1
Password:
Entering SLIP mode.
```

## ip tcp header-compression

---

```
Interface IP address is 10.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

The message “Header compression will match your system” indicates that the interface is set to match the compression status used by the host at the other end of the asynchronous line. If the line was configured to have header compression on, this line would read “Header compression is On.”

### Example

The following example enables Van Jacobson TCP header compression. The **passive** keyword prevents transmission of compressed packets until a compressed packet arrives from the IP link. Notice that asynchronous routing and dynamic addressing are also enabled.

```
interface async 6
  async dynamic routing
  async dynamic address
  ip tcp header-compression passive
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ppp**  
**slip**  
**slip default**  
**slip /compressed**

## ip unnumbered

To conserve network resources, use the **ip unnumbered** interface configuration command. To disable unnumbered interfaces, use the **no** form of this command.

```
ip unnumbered type number  
no ip unnumbered
```

### Syntax Description

*type*        Interface type.

*number*      Interface number.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

You must use either the **ip address** or **ip unnumbered** command to provide the local address for an interface.

Unnumbered interfaces do not have an address. Network resources are conserved because fewer network numbers are used and routing tables are smaller.

Whenever the unnumbered interface generates a packet (for example, a routing update), it uses the address of the specified interface as the source address of the IP packet. It also uses the address of the specified interface to determine which routing processes are sending updates over the unnumbered interface. Restrictions include the following:

- You cannot use the **ping** command to determine whether the interface is up, because the interface has no address. SNMP can be used to remotely monitor interface status.
- You cannot netboot an executable image over an unnumbered serial interface.
- The arguments *type* and *number* must be another interface in the network server that has an IP address, not another unnumbered interface.

### Example

The following example configures async interface 6 as unnumbered:

```
interface async 6  
  ip unnumbered ethernet 0
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ip address**

## ipx compression cipx

To enable compression of IPX packet headers in a PPP session, use the **ipx compression cipx** interface configuration command. To disable compression of IPX packet headers in a PPP session, use the **no** form of this command.

**ipx compression cipx** *number-of-slots*  
**no ipx compression cipx**

### Syntax Description

*number-of-slots* Number of stored IPX headers allowed. The range is from 10 to 256. The default is 16.

A slot is similar to a table entry for a complete IPX header. When a packet is received, the receiver stores the complete IPX header in a slot and tells the destination which slot it used. As subsequent CIPX packets are sent, the receiver uses the slot number field to determine which complete IPX header to associate with the CIPX packet before passing the packet up to IPX.

### Default

No compression of IPX packets during a PPP session.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1.

This interface configuration command enables IPX header compression on PPP links.

### Example

The following example enables IPX header compression for PPP:

```
encapsulation ppp  
ipx compression cipx 128
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**show ipx compression**

## ipx ppp-client

To enable a non-routing IPX client to connect to an asynchronous interface, the interface must be associated with a loopback interface configured to run IPX. To permit such connections, use the **ipx ppp-client** interface configuration command. To disable a non-routing IPX client, use the **no** form of this command.

```
ipx ppp-client loopback number
no ipx ppp-client loopback number
```

### Syntax Description

<b>loopback</b>	Loopback interface configured with a unique IPX network number.
<i>number</i>	Number of the loopback interface.

### Default

IPX client connections are not permitted over PPP.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1.

This command enables IPX clients to log into the router from a device running a virtual terminal protocol, then issue the PPP command at the EXEC prompt to connect to a remote device.

You must first configure a loopback interface with a unique IPX network number. The loopback interface is then assigned to an asynchronous interface, which permits IPX clients to connect to the asynchronous interface.

### Example

The following example configures IPX to run over PPP on asynchronous interface 3:

```
ipx routing 0000.0c07.b509
interface loopback0
  no ip address
  ipx network 544
  ipx sap-interval 2000
interface ethernet0
  ip address 172.21.14.64
  ipx network AC150E00
  ipx encapsulation SAP
interface async 3
  ip unnumbered ethernet0
  encapsulation ppp
  async mode interactive
  async default ip address 172.18.1.128
  ipx ppp-client loopback0
  ipx sap-interval 0
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**interface loopback**

**ipx network**

## peer default ip address

Use the **peer default ip address** interface configuration command to specify an IP address, an address from a specific IP address pool, or an address from the DHCP mechanism to be returned to a remote peer connecting to this interface. This command sets the address used on the remote (PC) side. Use the **no** form of this command to disable a prior peer IP address pooling configuration on an interface.

To remove the default address from your configuration, use the **no** form of this command also.

```
peer default ip address { ip-address | dhcp | pool [pool-name] }
no peer default ip address
```

### Syntax Description

<i>ip-address</i>	Specific IP address to be assigned to a remote peer dialing in to this interface. To prevent the assignment of duplicate IP addresses on two or more interfaces, this form of the command cannot be applied to a dialer rotary group nor to an ISDN interface.
<b>dhcp</b>	Retrieve an IP address from the DHCP server.
<b>pool</b>	Use the Global Default Mechanism as defined by the <b>ip address-pool</b> command unless the optional <i>pool-name</i> is supplied.
<i>pool-name</i>	(Optional) Name of a local address pool created using the <b>ip local pool</b> command. The router retrieves an address from this pool regardless of the Global Default Mechanism setting.

### Default

**pool**

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.0.

This command applies to point-to-point interfaces that support the PPP or SLIP encapsulation.

---

**Note** This command replaces the **async default ip address** command.

---

This command allows an administrator to configure all possible address pooling mechanisms on a interface-by-interface basis. The **peer default ip address** command can be used to override the Global Default Mechanism defined by the **ip address-pool** command on an interface-by-interface basis.

For all interfaces not configured with a peer default IP address mechanism (equivalent to selecting the **peer default ip address pool** command), the router uses the Global Default Mechanism that is defined by the **ip address-pool** command.

## peer default ip address

---

If you select the **peer default ip address pool** *pool-name* command, then the router uses the locally configured pool on this interface and does not follow the Global Default Mechanism.

If you select the **peer default ip address** *ip-address* form of this command, the specified IP address is assigned to any peer connecting to this interface and any Global Default Mechanism is overridden for this interface.

If you select the **peer default ip address dhcp** form of this command, the DHCP proxy-client mechanism is used by default on this interface and any Global Default Mechanism is overridden for this interface.

### Example

The following example specifies address 192.31.7.51 for async interface 6:

```
line 20
  speed 115200
interface async 6
  peer default ip address 192.31.7.51
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**async dynamic address**

## ppp

To start an asynchronous connection using PPP, use the **ppp** EXEC command.

```
ppp {/default | {remote-ip-address | remote-name} [@tacacs-server]} [/routing]
```

### Syntax Description

<b>/default</b>	Makes a PPP connection when a default address has been configured.
<i>remote-ip-address</i>	IP address of the client workstation or PC. This parameter can only be specified if the line is set for dynamic addresses using the <b>async address dynamic</b> line configuration command.
<i>remote-name</i>	Name of the client workstation or PC. This parameter can be specified if the line is set for dynamic addresses using the <b>async address dynamic</b> line configuration command.
<i>@tacacs-server</i>	(Optional) IP address or IP host name of the TACACS server to which the user's TACACS authentication request is sent.
<b>/routing</b>	(Optional) Indicates that the remote system is a router and that routing messages should be exchanged over the link. The line must be configured for asynchronous routing using PPP encapsulation.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in a release prior to Cisco IOS Release 10.0.

When you connect from a remote node computer to an EXEC session on the access server and want to connect from the access server to a device on the network, issue the **ppp** command.

If you specify an address for the TACACS server (either **/default** or *tacacs-server*), the address must be the first parameter in the command after you type **ppp**. If you do not specify an address or enter **default**, you are prompted for an IP address or host name. You can enter **default** at this point.

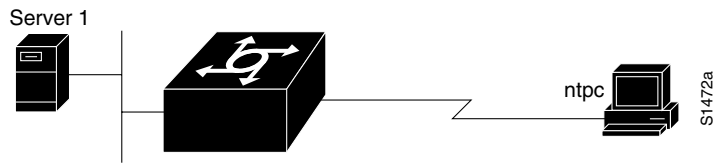
To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from the EXEC by using the **exit** command.

### Example

The following example shows a line that is in asynchronous mode using PPP encapsulation (see Figure 4). The PC's name is *ntpc*—assuming that the name *ntpc* is in the Domain Naming System (DNS) so that it can be resolved to a real IP address). The PC must be running a terminal emulator program.

```
router> ppp ntpc@server1
```

**Figure 4** Using the PPP EXEC Command



## ppp caller name

To set the caller option when no Calling Line Identification (CLID) is available, use the **ppp caller name** command in interface configuration mode. To remove the name, use the **no** form of this command.

**ppp caller name** *name*

**no ppp caller name** *name*

### Syntax Description

*name* Username string for this call.

### Defaults

Command is disabled by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

This command sets the username used when the CLID is not available. This username is used only in the case where the **ppp dnis** command is configured and the CLID is not available.

### Example

The following example shows how to configure a call to user1:

```
interface Serial0:15
  description "PRI D channel"
  ip unnumbered Loopback0
  encapsulation ppp
  no keepalive
  dialer pool-member 1 max-link 1
  isdn switch-type primary-net5
  isdn incoming-voice modem
  no fair-queue
  no cdp enable
  ppp caller name user1
  ppp authentication pap chap callin USERS&TUNNELS
  ppp chap hostname osh
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ppp dnis**

## ppp dnis

To configure a set of dialed number identification service (DNIS) numbers to check an incoming call against to automatically authenticate and authorize a user, use the **ppp dnis** command in interface configuration mode. To remove the numbers, use the **no** form of this command.

**ppp dnis** *DNIS-numbers*

**no ppp dnis** *DNIS-numbers*

### Syntax Description

*DNIS-numbers* Set of DNIS numbers that will be checked when a call comes in.

### Defaults

Command is disabled by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

This command enables a method of authenticating and authorizing a user based on the DNIS. The DNIS is the number dialed by the user. If the dialed number for this session matches one of the numbers configured in the **ppp dnis** command, the user is automatically authenticated and authorized for the session. Any other configured PPP authentication is not performed. In the case of DNIS authentication, the Calling Line Identification (CLID) is used as the username. If the CLID is unavailable, the username is the name configured with the **ppp caller name** command. If neither the CLID nor a caller name is configured, the username will automatically be set to “no-clid.”

### Example

The following example shows how to set the DNIS for a call:

```
interface Serial0:15
  description "PRI D channel"
  ip unnumbered Loopback0
  encapsulation ppp
  no keepalive
  dialer pool-member 1 max-link 1
  isdn switch-type primary-net5
  isdn incoming-voice modem
  no fair-queue
  no cdp enable
  ppp dnis 13693 132
  ppp authentication pap chap callin USERS&TUNNELS
  ppp chap hostname osh
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ppp caller name**

## ppp iphc max-header

To set the maximum size of the largest IP header that may be compressed when configuring Internet Protocol Header Compression (IPHC) control options over PPP, use the **ppp iphc max-header** command in interface configuration mode. To change the configuration, use the **no** form of this command.

**ppp iphc max-header** *bytes*  
**no ppp iphc max-header** *bytes*

### Syntax Description

<i>bytes</i>	Maximum size, in bytes, of the largest IP header that may be compressed. The range is from 60 to 168 bytes, and the default is 168 bytes.
--------------	---

### Defaults

168 bytes

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

There are two types of IP header compression used over PPP: Van Jacobsen header compression defined in RFC 1332 and enabled with the **ip tcp header-compression** command, and IPHC defined in RFC 2509 and enabled with the **ip rtp header-compression** command. The **ppp iphc** set of commands controls parameters that pertain to the form of IPHC described in RFC 2509.

The IPHC specification allows low speed links to run more efficiently by reducing the size of the IP headers as transmitted on the link. IPHC supports compressed Real-Time Transport Protocol (cRTP), compressed User Datagram Protocol (cUDP), and compressed Transaction Control Protocol (cTCP).

An IPHC-enabled interface sends only changes to the header instead of sending the entire header with every packet. At the beginning of a transmission, the transmitting end (the compressor) sends a full header packet to the receiving end (the decompressor). After the initial packet is sent, the compressor sends all other packets with headers that contain only the differences between them and the original full header. The decompressor maintains a copy of the original full header and reconstructs all the other packet headers by adding the changes to them.

The header data that is different with each packet is referred to as the session state, and is identified by a session ID or connection ID.

When the decompressor receives a compressed packet, it reconstructs the packet header by adding the difference to the saved uncompressed header. Typically, IPHC enables the header to be compressed to two bytes (four bytes if UDP checksums are used).

The following fields in a packet header usually remain the same throughout a transmission:

- IP source and destination addresses
- UDP and TCP source and destination ports

- RTP synchronization source (SSRC) fields

The following fields in a packet header usually change during a transmission:

- IP packet ID
- Checksum
- Sequence number
- RTP time stamp
- The RTP marker bit

## Example

The following example shows how to change the maximum size of the largest IP header that may be compressed from the default of 168 bytes to 114 bytes:

```
interface Multilink1
 ip address 10.100.253.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 ip tcp header-compression iphc-format
 no ip mroute-cache
 fair-queue 64 256 1000
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-format
 ip rtp priority 16384 50 64
 ppp iphc max-header 114
 ppp iphc max-time 10
 ppp iphc max-period 512
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ip rtp header-compression**  
**ip tcp header-compression**  
**ppp iphc max-period**  
**ppp iphc max-time**

## ppp iphc max-period

To set the maximum number of compressed packets that can be sent before a full header when configuring Internet Protocol Header Compression (IPHC) control options over PPP, use the **ppp iphc max-period** command in interface configuration mode. To change the configuration, use the **no** form of this command.

**ppp iphc max-period** *packets*

**no ppp iphc max-period** *packets*

### Syntax Description

<i>packets</i>	Maximum number of compressed packets that can be sent before a full header. The range is from 1 to 65,535 packets, and the default is 256 packets.
----------------	--

### Defaults

256 packets

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

There are two types of IP header compression used over PPP: Van Jacobsen header compression, which is defined in RFC 1332, and a newer compression type described in RFC 2509. The **ppp iphc** set of commands controls parameters that pertain to the form of IPHC described in RFC 2509.

The IPHC specification allows low speed links to run more efficiently when IP headers are extremely large. IPHC supports compressed Real-Time Transport Protocol (cRTP), compressed User Datagram Protocol (cUDP), and compressed Transaction Control Protocol (cTCP).

An IPHC-enabled interface sends only changes to the header instead of sending the entire header with every packet. At the beginning of a transmission, the transmitting end (the compressor) sends a full header packet to the receiving end (the decompressor). After the initial packet is sent, the compressor sends all other packets with headers that contain only the differences between them and the original full header. The decompressor maintains a copy of the original full header and reconstructs all the other packet headers by adding the changes to them.

The header data that is different with each packet is referred to as the session state, and is identified by a session ID or connection ID.

When the decompressor receives a compressed packet, it reconstructs the packet header by adding the difference to the saved uncompressed header. Typically, IPHC enables the header to be compressed to two bytes (four bytes if UDP checksums are used).

The following fields in a packet header usually remain the same throughout a transmission:

- IP source and destination addresses
- UDP and TCP source and destination ports
- RTP synchronization source (SSRC) fields

The following fields in a packet header usually change during a transmission:

- IP packet ID
- Checksum
- Sequence number
- RTP time stamp
- RTP marker bit

The **ppp iphc max-period** command is specifically related to an IPHC frame format known as *compressed\_non\_TCP*. The recovery of lost compressed\_non\_TCP frames on lossy links is much improved by allowing more full headers to flow and by configuring less compression.

## Example

The following example shows how to increase the maximum number of compressed packets that can be sent before a full header from 256 to 512 packets when configuring IPHC control options over PPP:

```
interface Multilink1
 ip address 10.100.253.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 ip tcp header-compression iphc-format
 no ip mroute-cache
 fair-queue 64 256 1000
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-format
 ip rtp priority 16384 50 64
 ppp iphc max-header 114
 ppp iphc max-time 10
 ppp iphc max-period 512
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ip rtp header-compression**  
**ip tcp header-compression**  
**ppp iphc max-header**  
**ppp iphc max-time**

## ppp iphc max-time

To set the maximum time allowed between full headers when configuring Internet Protocol Header Compression (IPHC) control options over PPP, use the **ppp iphc max-time** command in interface configuration mode. To change the configuration, use the **no** form of this command.

**ppp iphc max-time** *seconds*  
**no ppp iphc max-time** *seconds*

### Syntax Description

<i>seconds</i>	Maximum time, in seconds, allowed between full headers. The range is from 1 to 255 seconds, and the default is 5 seconds.
----------------	---

### Defaults

5 seconds

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

There are two forms of IP header compression used over PPP: Van Jacobsen header compression, which is defined in RFC 1332, and a newer form of compression described in RFC 2509. The **ppp iphc** set of commands controls parameters that pertain to the form of IPHC described in RFC 2509.

The IPHC specification allows low speed links to run more efficiently by reducing the size of IP headers as transmitted on the link. IPHC supports compressed Real-Time Transport Protocol (cRTP), compressed User Datagram Protocol (cUDP), and compressed Transaction Control Protocol (cTCP).

An IPHC-enabled interface sends only changes to the header instead of sending the entire header with every packet. At the beginning of a transmission, the transmitting end (the compressor) sends a full header packet to the receiving end (the decompressor). After the initial packet is sent, the compressor sends all other packets with headers that contain only the differences between them and the original full header. The decompressor maintains a copy of the original full header and reconstructs all the other packet headers by adding the changes to them.

The header data that is different with each packet is referred to as the session state, and is identified by a session ID or connection ID.

When the decompressor receives a compressed packet, it reconstructs the packet header by adding the difference to the saved uncompressed header. Typically, IPHC enables the header to be compressed to two bytes (four bytes if UDP checksums are used).

The following fields in a packet header usually remain the same throughout a transmission:

- IP source and destination addresses
- UDP and TCP source and destination ports
- RTP synchronization source (SSRC) fields

The following fields in a packet header usually change during a transmission:

- IP packet ID
- Checksum
- Sequence number
- RTP time stamp
- RTP marker bit

The **ppp iphc max-time** command is specifically related to an IPHC frame format known as *compressed\_non\_TCP*. The recovery of lost compressed\_non\_TCP frames on lossy links is much improved by allowing more full headers to flow and by configuring less compression.

## Example

The following example shows how to change the number of compressed packets that can be sent before a full header from the default 5 seconds to 10 seconds:

```
interface Multilink1
 ip address 10.100.253.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 ip tcp header-compression iphc-format
 no ip mroute-cache
 fair-queue 64 256 1000
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp header-compression iphc-format
 ip rtp priority 16384 50 64
 ppp iphc max-header 114
 ppp iphc max-time 10
 ppp iphc max-period 512
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ip rtp header-compression**  
**ip tcp header-compression**  
**ppp iphc max-header**  
**ppp iphc max-time**

## ppp lcp fast-start

To allow a Point-to-Point (PPP) interface to respond immediately to incoming packets once a connection is established, use the **ppp lcp fast-start** interface configuration command. To specify that PPP delay before responding, use the **no** form of this command.

**ppp lcp fast-start**

**no ppp lcp fast-start**

### Syntax Description

This command has no arguments or keywords.

### Default

Default is enabled.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

Some systems, typically those with external modems, may have problems with slow or electrically noisy hardware. If the **no ppp lcp fast-start** command is specified, PPP starts a debounce timer and waits for it to expire before attempting to communicate with the peer system, thereby reducing the probability of a false start on the interface.

If the **no ppp lcp fast-start** command is not specified, PPP will not use a debounce timer and will respond immediately to incoming packets once a connection is made.

The default fast-start enabled state should not be disabled unless there is a problem with slow or electronically noisy hardware. This setting prevents PPP from waiting for a debounce timer to expire before responding to inbound frames.

### Example

The following example disables fast start:

```
no ppp lcp fast-start
```

## ppp ms-chap refuse

To refuse Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) authentication from peers requesting it, use the **ppp ms-chap refuse** command in interface configuration mode. To allow MS-CHAP authentication, use the no form of this command.

**ppp ms-chap refuse [callin]**

**no ppp ms-chap refuse [callin]**

### Syntax Description

**callin** (Optional) Specifies that the router will refuse to answer MS-CHAP authentication challenges received from the peer, but will still require the peer to answer any MS-CHAP challenges the router sends.

### Defaults

This command is disabled by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

This command specifies that MS-CHAP authentication is disabled for all calls, meaning that all attempts by the peer to force the user to authenticate using MS-CHAP will be refused. If the **callin** keyword is used, MS-CHAP authentication is disabled for incoming calls from the peer, but will still be performed on outgoing calls to the peer.

If outbound Password Authentication Protocol (PAP) has been enabled (using the `ppp pap sent-username` command), PAP will be suggested as the authentication method in the refusal packet.

### Example

The following example shows how to disable MS-CHAP authentication if a peer calls in requesting MS-CHAP authentication. The method of encapsulation on interface ISDN BRI number 0 is PPP.

```
interface bri 0
  encapsulation ppp
  ppp ms-chap refuse
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**aaa authentication ppp**  
**ppp authentication**  
**ppp authentication ms-chap-v2**  
**ppp chap password**  
**ppp chap wait**  
**ppp pap sent-username**

## ppp ms-chap-v2 refuse

To refuse Microsoft Challenge Handshake Authentication Protocol (MS-CHAP) version 2 authentication from peers requesting it, use the **ppp ms-chap-v2 refuse** command in interface configuration mode. To allow MS-CHAP version 2 authentication, use the no form of this command.

**ppp ms-chap-v2 refuse** [**callin**]

**no ppp ms-chap-v2 refuse** [**callin**]

### Syntax Description

<b>callin</b>	(Optional) Specifies that the router will refuse to answer MS-CHAP authentication challenges received from the peer, but will still require the peer to answer any MS-CHAP challenges the router sends.
---------------	---

### Defaults

This command is disabled by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

This command specifies that MS-CHAP version 2 authentication is disabled for all calls, meaning that all attempts by the peer to force the user to authenticate using MS-CHAP version 2 will be refused. If the callin keyword is used, MS-CHAP version 2 authentication is disabled for incoming calls from the peer, but will still be performed on outgoing calls to the peer.

If outbound Password Authentication Protocol (PAP) has been enabled (using the **ppp pap sent-username** command), PAP will be suggested as the authentication method in the refusal packet.

### Example

The following example shows how to disable MS-CHAP version 2 authentication if a peer calls in requesting MS-CHAP version 2 authentication. The method of encapsulation on interface ISDN BRI number 0 is PPP.

```
interface bri 0
  encapsulation ppp
  ppp ms-chap-v2 refuse
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**aaa authentication ppp**  
**ppp authentication**  
**ppp authentication ms-chap**

```
ppp chap password  
ppp chap wait  
ppp pap sent-username
```

## service old-slip-prompts

To provide backward compatibility for client software scripts expecting SLIP and PPP dialogs to be formatted with software release 9.1 or earlier, use the **service old-slip-prompts** global configuration command. Use the **no** form of this command to disable this function.

**service old-slip-prompts**  
**no service old-slip-prompts**

### Syntax Description

This command has no arguments or keywords.

### Default

The prompts and information transmitted by SLIP and PPP are formatted with the current release of Cisco IOS software.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1.

### Example

The following example shows the output of a SLIP command after **service old-slip-prompts** is enabled:

```
router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
router(config)# service old-slip-prompts
router(config)# exit
router# slip
IP address or hostname: 2.2.2.2
Entering SLIP mode.
Your IP address is 2.2.2.2. MTU is 1500 bytes
```

## show ipx compression

To show the current status and statistics of IPX header compression during PPP sessions, use the **show ipx compression** EXEC command.

**show ipx compression detail** *int-spec*

### Syntax Description

**detail** Shows detailed link-state database information for NLSP.

*int-spec* Interface type, as listed in Table 113.

**Table 113** Interface Types

Keyword	Description
Async	Asynchronous interface.
Ethernet	Ethernet IEEE 802.3 interface.
Null	Null interface.
Serial	WAN serial interface.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1.

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ipx compression cipx**  
**show ipx interface**

## slip

To start a serial connection to a remote host by using SLIP, use the **slip** EXEC command.

```
slip [/default] { remote-ip-address | remote-name } [@tacacs-server] [/routing] [/compressed]
```

### Syntax Description

<b>/default</b>	(Optional) Makes a SLIP connection when a default address has been configured.
<i>remote-ip-address</i>	IP address of the client workstation or PC.
<i>remote-name</i>	Name of the client workstation or PC.
<i>@tacacs-server</i>	(Optional) IP address or IP host name of the TACACS server to which your TACACS authentication request is sent.
<b>/routing</b>	(Optional) Indicates that the remote system is a router. Line must be configured for asynchronous routing using SLIP encapsulation.
<b>/compressed</b>	(Optional) Indicates that IP header compression should be negotiated.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in a release prior to Cisco IOS Release 10.0.

When you connect from a remote node computer to the EXEC facility on a router and want to connect from the router to a device on the network, issue the **slip** command.

If you specify an address for the TACACS server by using **/default** or *tacacs-server*, the address must be the first parameter in the command after you enter **slip**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

If you do not use the *tacacs-server* argument to specify a TACACS server for SLIP address authentication, the TACACS server specified at login (if any) is used for the SLIP address query.

To optimize bandwidth on a line, SLIP enables compression of the SLIP packets using Van Jacobson TCP header compression as defined in RFC 1144.

Your system administrator must configure the system with the **ip tcp header-compression passive** command for the **/compressed** command option to be valid in EXEC mode. The **ip tcp header-compression** command forces header compression on or off. The default is to not compress the packets. The configuration file must have header compression on and the **slip /compressed** EXEC command must be entered for header compression to occur.

To terminate a session initiated with the slip command, disconnect from the device on the network using the command specific to that device. Then, exit from the EXEC by using the **exit** command.

## Examples

The following example makes a connection when a default IP address is assigned. Once a correct password is entered, you are placed in SLIP mode, and the IP address is displayed.

```
router> slip
Password:
Entering SLIP mode.
Your IP address is 192.31.7.28, MTU is 1524 bytes
```

The following example illustrates the prompts displayed and the response required when you use dynamic addressing to assign the SLIP address:

```
router> slip
IP address or hostname? 192.31.6.15
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

In the preceding example, the address 192.31.6.15 has been assigned as the default. Password verification is still required before SLIP mode can be enabled.

```
router> slip /default
Password:
Entering SLIP mode
Your IP address is 192.31.6.15, MTU is 1524 bytes
```

The following example illustrates the implementation of header compression on the interface with the IP address 128.66.2.1:

```
router> slip 128.66.2.1 /compressed
Password:
Entering SLIP mode.
Interface IP address is 128.66.2.1, MTU is 1500 bytes.
Header compression will match your system.
```

In the preceding example, the interface is configured for **ip tcp header-compression passive**, which permits the user to enter the **/compressed** keyword at the EXEC mode prompt. The message “Header compression will match your system” indicates that the user specified compression. If the line was configured for **ip tcp header-compression on**, this line would read “Header compression is On.”

The following example specifies a TACACS server named server1 for address authentication:

```
router> slip 1.0.0.1@server1
Password:
Entering SLIP mode.
Interface IP address is 1.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

