# Chapter 1
# SLIP Configuration and Management

1

This chapter describes how to configure the Serial Line Internet Protocol (SLIP) on the Cisco Systems terminal servers. You will find the following information in this chapter:

- An overview of the Serial Line Internet Protocol.
- Configuring procedures for SLIP, SLIP access controls, and specifying extended Boot Protocol (BootP) requests.
- EXEC commands for maintaining and monitoring SLIP connections.

Making SLIP connections is described in the “Terminal Server User Commands” chapter.

The commands to define the configuration are entered in configuration mode, which is privileged. To enter configuration mode, type the **configure** command at the EXEC prompt. You can then enter the commands described in this chapter. For more information about the configuration mode, see“The Configure Command” in the “Startup and Basic Configuration” chapter.

A command summary is included at the end of the chapter.

## The Serial Line Internet Protocol (SLIP)

As personal computers (PCs) become increasingly powerful, they can be connected to networks as full network nodes and not simply as dumb terminals. As a full network node, a PC can send and receive mail, transfer files using sophisticated file transfer protocols, use network file systems, form virtual terminal connections to other hosts, and so on. Several protocol suites can perform these functions, but the TCP/IP suite is preferred when a variety of computers must communicate.

The Serial Line Internet Protocol, or SLIP, is an inexpensive way of connecting PCs to a network. SLIP can be used over asynchronous dial-up modems, allowing computers in people’s homes to be connected to a network without the cost of a leased line. Dial-up SLIP links can also be used for remote sites that need only occasional or backup connectivity. Both public-domain and vendor-supported SLIP implementations are available for a variety of computer applications.

SLIP defines a method of sending Internet packets over standard RS-232 asynchronous serial lines. It is a de facto standard, commonly used for point-to-point serial connections running TCP/IP. SLIP is commonly used on dedicated serial links with line speeds between 1,200 and 19,200 baud.

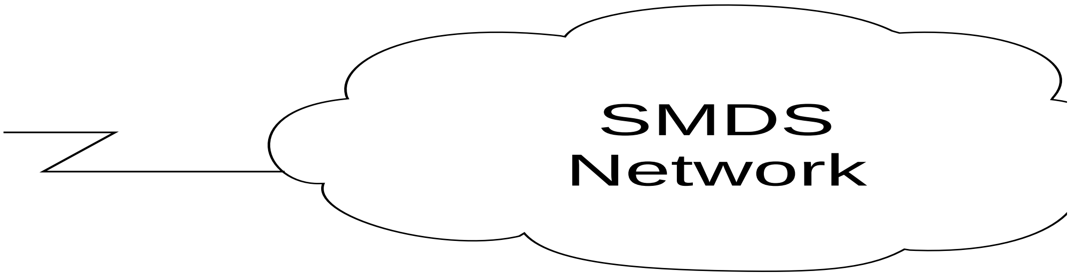
The version of SLIP described in this manual was originally implemented by researchers at the University of California at Berkeley in their 4.2 BSD version of the UNIX operating system. Although variants have been proposed, the Berkeley version has emerged as a de facto standard. Refer to RFC1055 for more information about SLIP.

## Cisco's Implementation of SLIP

The Cisco Systems terminal servers combine the ability to run SLIP with hardware and software optimized to handle terminal line traffic. The terminal server concentrates a large number of SLIP PC hosts onto a network interface, allowing the PCs to communicate to any host on the network. The terminal server can support any combination of SLIP lines and lines dedicated to normal asynchronous devices, such as terminals and modems.

Figure 1-1 illustrates a typical SLIP configuration.

*Figure 1-1* Sample SLIP Configuration



In addition to implementing the de facto standard, Cisco's implementation of SLIP offers both dedicated and dynamic address assignment, configurable hold queue and IP packet sizes, extended BootP requests, and permit/deny conditions for controlling access to the line.

### SLIP and Broadcasts

The Cisco Systems terminal servers recognize a variety of Internet broadcast addresses. When a terminal server receives an Internet packet with one of these addresses from a SLIP client, it rebroadcasts the packet onto the network without changing the Internet header. The terminal server does not alter the packet's broadcast address to match the form of broadcast address it prefers.

The terminal server receives a copy of SLIP client broadcasts, and responds to BootP requests with the Internet address of the line that received them. This facility allows the SLIP client software to automatically determine its own Internet address.

# *Configuring SLIP*

Following are the basic steps for configuring SLIP on the Cisco terminal server.

*Step 1:* Enable SLIP on a terminal server line using the **slip dedicated**, **slip address**, or **slip address dynamic** line subcommands. See the section "SLIP Connection Types" for more information about selecting these commands.

*Step 2:* Make the appropriate settings for the line. Line settings include baud rate, flow control, stop bits, and modem control. These settings are described in the sections "Configuring Asynchronous Lines" and "Configuring the Modem Control Lines" in the "System Configuration" chapter.

*Step 3:* Specify packet size limits and access lists for control of traffic to or from a SLIP-enabled line, if needed.

*Step 4:* Specify extended BootP requests, if needed.

Additionally, SLIP supports simultaneous routing of packets to networks connected via SLIP lines on the Cisco TRouter, the combination terminal server/router product.

The following sections describe how to configure the terminal server for SLIP operation. Descriptions of the EXEC commands used to monitor and maintain a SLIP link follow these sections.

## *SLIP Connection Types*

Users use one of two EXEC commands to make a SLIP connection, depending on how the line is configured:

**slip**
**slip default**

It is also possible to configure a dedicated SLIP line, in which case no EXEC command is required to make the connection.

The following paragraphs describe the ways in which the SLIP lines can be configured.

### *Dedicated SLIP Line*

A line can be permanently configured for SLIP using the **slip dedicated** and **slip address** line configuration commands. In this case, an EXEC is not started on the line, so the user need not enter a command to initiate SLIP. They need only to make the physical connection to the line and then are immediately placed in SLIP mode. No prompt or message is issued.

### *Permanent SLIP Address*

A line can be assigned a permanent SLIP address with the **slip address** line configuration command. In this case, the user issues the **slip** EXEC command to put the line into SLIP mode. If the server has been configured to authenticate SLIP connections, the user is prompted for a password before the line is placed in SLIP mode.

### *Dynamic SLIP Address*

A line can be configured for dynamic assignment of SLIP addresses with the **slip address dynamic** command. In this case, the user enters the **slip** EXEC command and is prompted for the IP address or logical host name to use. This address is validated via TACACS (when enabled) and the line is put into SLIP mode using the address requested.

This feature is useful in a situation where the user has a need to know the IP address of a line. A personal computer running an application that automatically dials in using SLIP and polls for electronic mail messages would be an example of this. The application can be set up to dial in periodically and enter the required IP address and password.

### *Default SLIP Address*

A line can also be given a default address to use. In this case, the user issues the **slip default** EXEC command, the transaction is validated by the TACACS server (when enabled), and the line is put into SLIP mode using the address configured with the IP address argument of the **slip address dynamic** configuration command.

This feature is useful when it is not reasonable for all users to know the IP address to use to gain access to a system, for example, a server that is available to many students on a campus would be an example of this. Instead of requiring each to know an IP address, they need only enter the EXEC **slip default** command and let the server select the line to use.

*Note:* When TACACS authentication of SLIP addresses is used, the configuration command **tacacs-server optional-passwords** may be used to suppress the password prompt if your TACACS server supports validation of addresses without passwords. See "Establishing Passwords and System Security (TACACS)" in the "System Configuration" chapter for information about TACACS verification.

## *Disabling SLIP on a Line*

Once a line is configured for SLIP, the EXEC responds to the **slip** or **slip default** EXEC commands by turning on SLIP on the line, displaying the Internet address and the size of the largest Internet packet the SLIP support can handle. The line exits SLIP mode when the modem is hung up or an EXEC **clear line** command is issued.

The **no slip** line subcommand disables SLIP mode. It has this syntax:

**no slip**

Use this command to disable SLIP on a line that has previously had SLIP enabled.

## *Specifying a SLIP Address*

The **slip address** line subcommand specifies the Internet address assigned to the SLIP client at the other end of the serial line connection. The command has this syntax:

**slip address** *internet-address*

The argument *internet-address* must be on the same network or subnet as one of the terminal server's network interfaces.

To put this line into SLIP mode, use the EXEC command **slip**; see "SLIP Connection Types" in this chapter.

*Example:*

This example sets IP address *182.32.7.51* on line 2.

```
line 2
slip address 182.32.7.51
```

## *Configuring a Dedicated SLIP Line*

The **slip dedicated** line subcommand puts the line in SLIP mode permanently. It has this simple syntax:

**slip dedicated**

The terminal server will not create an EXEC on this line, so it is not available for normal interactive use. No **slip** EXEC command is necessary to enable SLIP mode.

*Example:*

The **slip dedicated** statement permanently places the line into SLIP mode.

```
line 2
slip address 182.32.7.5
slip dedicated
```

## *Configuring the SLIP Line in Interactive Mode*

The **slip interactive** line subcommand allows a line to be used in either SLIP mode or interactive mode.

**slip interactive**

The **slip interactive** subcommand is generally used to void a **slip dedicated** line subcommand. It is the equivalent to the command **no slip dedicated** (although there is no such command). To put this line into SLIP mode, use the EXEC command **slip**; see "SLIP Connection Types" in this chapter.

Hanging up the modem or clearing the line puts the line back into interactive mode.

## *Configuring Dynamic Address Assignment*

Dynamic address assignment requires that the user enter an IP address to enter SLIP mode. The full syntax of the line subcommand to set this up follows:

**slip address dynamic**

This subcommand allows flexibility when the lines are in a rotary configuration, when you cannot predict to which line a user will connect. To put this line into SLIP mode, use the EXEC command **slip**.

This feature is supported when a TACACS server is used. (The host name sent in a TACACS request will be in all uppercase letters.)

---

*Note:* Dynamic address assignment can be assigned to a single line, or to a range of lines. See the examples for an illustration.

---

*Example 1:*

In the following example, line 3 is configured for dynamic address assignment:

```
line 3
slip address dynamic
```

*Example 2:*

In this example, the range of lines 10 through 15 are configured for dynamic address assignment.

```
line 10 15
slip address dynamic
```

## *Configuring Dynamic Address Assignment with a Default Address*

Lines configured for dynamic address assignment can also be given a default address to use so the user enters the EXEC command **slip default** to make connection. The syntax for this is as follows:

**slip address dynamic** *IP-address*

The argument *IP-address* is the IP address to use.

*Example:*

This example illustrates how to configure a default SLIP address.

```
line 1
slip address dynamic 108.91.3.4
```

## *Configuring the Hold Queue*

The **slip hold-queue** line subcommand specifies the limit of the SLIP output queue, which stores packets received from the network waiting to be sent to the SLIP client.

**slip hold-queue** *packets*

The argument *packets* is the maximum number of packets. The default is three packets; it is recommended that the queue size does not exceed 10.

*Example:*

This command changes the packet queue length to five packets.

```
line 5
slip address 182.32.7.5
slip hold-queue 5
```

## *Configuring the MTU Size of Internet Packets*

The **slip mtu** line subcommand specifies the size of the largest Internet packet that the SLIP support can handle.

**slip mtu** *bytes*

The argument *bytes* is the maximum number of bytes. The default is 1500 MTU.

You might want to change to a smaller MTU size if the SLIP application at the other end does not support packets of that size, or you want to assure a lower delay by using shorter packets. This can be desirable when the host Telnet echoing takes longer than 0.2 seconds. For instance, at 9600 baud, a 1500 byte packet takes about 1.5 seconds to transmit, so this delay would indicate that you want an MTU size of about 200.

On the other hand, the MTU size can be negotiated by TCP, regardless of what the terminal settings are, and this is the better way to do it. The terminal server performs IP fragmentation of packets larger than the specified MTU. Therefore, do not use this command unless the SLIP implementation supports re-assembly of IP fragments. Since each fragment occupies a spot in the output queue, it may also be necessary to increase the size of the SLIP hold queue.

*Example:*

These commands set the packet MTU size to 200 bytes.

```
line 5
slip address 182.32.7.5
slip mtu 200
```

## *Specifying SLIP Access Lists*

Access lists allow the system administrator to control the hosts that may be accessed by a PC running SLIP through a terminal server. In versions of the software prior to 8.3, turning on SLIP defeated any access lists that were defined for that line. As SLIP is different than a connection, separate access lists may be defined for SLIP and for normal connections. The software allows separate access lists to be defined for use when the line is running SLIP.

To configure an access list to be used on packets *from* the SLIP host, use this line subcommand:

**slip access-class** *number* **in**

When this command is entered, the IP destination address of each packet is run through the access list for acceptability, and dropped or passed. The argument *number* is the IP access list number (see the section “Configuring IP Access Lists” in the “TCP/IP Configuration ansd Management” chapter, for information about IP access lists).

To specify an access list to be used on packets being sent *to* the SLIP host, use this line subcommand:

**slip access-class** *number* **out**

When this command is entered, the IP source address is compared against the access list, and only those packets allowed by the access list are transmitted on the asynchronous line. The argument *number* is the IP access list number (see the section “Configuring IP Access Lists” in the “TCP/IP Configuration ansd Management” chapter, for information about IP access lists).

On the TRouter, it is also possible to use extended IP access lists on SLIP lines. In this case, the header of the packet is compared against the access list as appropriate.

*Example:*

This example assumes that SLIP users are restricted to certain servers designated as SLIP servers, but that normal terminal users can access anything on the local network.

```
! access list for normal connections
access-list 1 permit 131.108.0.0 0.0.255.255
!
! access list for SLIP packets.
access-list 2 permit 131.108.42.55
access-list 2 permit 131.108.111.1
access-list 2 permit 131.108.55.99
!
!Define all the lines with slip addresses and
!appropriate access lists
line 1 60
slip address dynamic
access-class 1 out
slip access-class 2 in
!
```

## *Specifying SLIP Extended BootP Requests*

The BootP (Boot Protocol) server for SLIP supports the extended BootP requests specified in RFC1084. These requests are specified with the **async-bootp** global configuration command. The full syntax for this command follows:

**async-bootp** *tag* **[***:hostname***]** *data...*
**no async-bootp**

The argument *tag* is the item being requested, and is one of the following expressed as file name, integer, or IP dotted decimal address:

- **bootfile**—Specifies use of a server boot file from which to download the boot program. Use the optional *:hostname* and *data* arguments to specify the file name.
- **subnet-mask** *mask*—Dotted decimal address specifying the network and local subnetwork mask (as defined by RFC950).
- **time-offset** *offset*—A signed 32-bit integer specifying the time offset of the local subnetwork in seconds from Coordinated Universal Time (UTC).
- **gateway** *address*—Dotted decimal address specifying the IP addresses of gateways for this subnetwork. A preferred gateway should be listed first.
- **time-server** *address*—Dotted decimal address specifying the IP address of time servers (as defined by RFC868).
- **IEN116-server** *address*—Dotted decimal address specifying the IP address of name servers (as defined by IEN 116).
- **DNS-server** *address*—Dotted decimal address specifying the IP address of Domain Name Servers (as defined by RFC1034).

- **log-server** *address*—Dotted decimal address specifying the IP address of an MIT-LCS UDP log server.
- **quote-server** *address*—Dotted decimal address specifying the IP address of Quote of the Day servers (as defined in RFC865).
- **lpr-server** *address*—Dotted decimal address specifying the IP address of Berkeley UNIX Version 4 BSD servers.
- **impress-server** *address*—Dotted decimal address specifying the IP address of Impress network image servers.
- **rlp-server** *address*—Dotted decimal address specifying the IP address of Resource Location Protocol (RLP) servers (as defined in RFC887).
- **hostname** *name*—The name of the client, (which may or may not be domain qualified, depending upon the site).
- **bootfile-size** *value*—A two-octet value specifying the number of 512 octet (byte) blocks in the default boot file.

Use the optional argument *:hostname* to indicate that this entry applies only to the host specified. The argument *:hostname* accepts both an IP address and logical host name.

The argument *data* can be a list of IP addresses entered in dotted decimal notation or as logical host names, a number, or a quoted string.

If no extended BootP commands are entered, by default the software generates a gateway and subnet mask appropriate for the local network.

Use the EXEC command **show async-bootp** to list the configured parameters. Use the **no async-bootp** command to clear the list.

*Example 1:*

This example illustrates how to specify different boot files, one for a PC, and one for a Macintosh.

```
async-bootp bootfile :128.128.1.1 “pcboot”
async-bootp bootfile :mac "macboot"
```

With this configuration, a BootP request from the host on *128.128.1.1* results in a reply listing the boot file name as *pcboot*. A bootP request from the host named *mac* results in a reply listing the boot file name as *macboot*.

*Example 2:*

This example specifies a subnet mask of 255.255.0.0.

```
async-bootp subnet-mask 255.255.0.0
```

*Example 3:*

This example specifies a negative time offset of the local subnetwork of -3600 seconds.

```
async-bootp time-offset -3600
```

*Example 4:*

This example specifies the IP address of a time server.

```
async-bootp time-server 128.128.1.1
```

## *Routing SLIP on the TRouter*

The Cisco TRouter can route packets to networks connected via a SLIP line. (Earlier versions of TRouter software supported only end-nodes connected via SLIP.)

*Example:*

To route to a network on a SLIP line, use the **ip route** command in combination with the **slip address** line subcommand.

The following example creates a static route that is added to the router's routing table when the line enters SLIP mode. The route is deleted when SLIP is turned off on that line, such as when a modem hangs up.

```
line 5
slip address 130.107.4.40
ip route 10.0.0.0 130.107.4.40
```

Once the SLIP line is active, this route behaves identically to any other static route, and may be redistributed using any of the dynamic routing protocols. Refer to the Cisco *Router Products Configuration and Reference* publication for details on configuring dynamic routing protocols.

*Note:* Dynamic routing updates are not sent over the SLIP lines. In an average-size network, routing updates could overload a low speed SLIP line. This means that systems on the remote network must be configured with static routes to reach the main network.

## SLIP Configuration Example

The following sample configuration assigns an Internet address to a SLIP line and puts the line in SLIP mode permanently. Setting the stop bits to 1 is a performance enhancement.

```
!
line 20
location Joe's computer
stopbits 1
speed 19200
slip address 182.32.7.51
slip dedicated
!
```

## Maintaining SLIP

This section describes the EXEC commands for maintaining SLIP support on the terminal server.

Maintaining SLIP is a simple task on the terminal server and TRouter. The **clear line** EXEC command disables SLIP mode and starts an EXEC on a nondedicated SLIP line. Enter this command at the EXEC prompt:

**clear line** *line-number*

The argument *line-number* specifies the line. This command is the only way to exit SLIP mode on a line without modem control.

## Monitoring SLIP

Use the EXEC **show** commands described in this section to obtain displays of activity on the SLIP line.

### Displaying the Mapped Internet Address

The **show ip aliases** command displays Internet addresses mapped to TCP ports (*aliases*) and SLIP addresses, which are treated similarly to the aliases. Enter this command at the EXEC prompt:

**show ip aliases**

To distinguish a SLIP address from a normal alias address, the command output uses the form SLIP TTY*n* for the “port” number, where *n* is the terminal number (in octal).

Sample output follows:

```
    IP Address    Port
  192.31.7.52     SLIP TTY35
  192.31.7.53     SLIP TTY36
  192.31.7.54     SLIP TTY37
  192.31.7.55     SLIP TTY40
```

The display lists the IP address and corresponding port number.

## Displaying the IP ARP Cache

The **show ip arp** EXEC command displays the Address Resolution Protocol (ARP) cache, where SLIP addresses appear as permanent ARP table entries. To display the IP ARP cache, use the following EXEC command:

**show ip arp**

An Address Resolution Protocol establishes correspondences between network addresses (an IP address, for example) and LAN hardware addresses (Ethernet addresses). A record of each correspondence is kept in a cache for a predetermined amount of time and then discarded. Following is sample output. Table 1-1 describes the fields displayed.

```
Protocol  Address         Age (min)   Hardware Addr   Type     Interface
Internet  131.108.1.140        137    aa00.0400.6408  ARPA     Ethernet0
Internet  131.108.1.111        156    0800.2007.8866  ARPA     Ethernet0
Internet  131.108.1.115         33    0000.0c01.0509  ARPA     Ethernet0
Internet  192.31.7.24            5    0800.0900.46fa  ARPA     Ethernet2
Internet  192.31.7.26           41    aa00.0400.6508  ARPA     Ethernet2
Internet  192.31.7.27            -    aa00.0400.0134  ARPA     Ethernet2
Internet  192.31.7.28           67    0000.0c00.2c83  ARPA     Ethernet2
Internet  192.31.7.17           67    2424.c01f.0711  ARPA     Ethernet2
Internet  192.31.7.18           64    0000.0c00.6fbf  ARPA     Ethernet2
Internet  192.31.7.21          114    2424.c01f.0715  ARPA     Ethernet2
Internet  131.108.1.33          15    0800.2008.c52e  ARPA     Ethernet0
Internet  131.108.1.55          44    0800.200a.bbfe  ARPA     Ethernet0
Internet  131.108.1.6           89    aa00.0400.6508  ARPA     Ethernet0
Internet  131.108.7.1            -    0000.0c00.750f  ARPA     Ethernet3
Internet  131.108.1.1            -    aa00.0400.0134  ARPA     Ethernet0
```

*Table 1-1* Show IP ARP Display Field Descriptions

| Field | Description |
|---|---|
| Protocol | Protocol for network address in Address field. |
| Address | The network address that corresponds to Hardware Addr. |
| Age (min) | Age, in minutes, of the last update of the cache entry. |
| Hardware Addr | LAN hardware address that corresponds to network address. |
| Type | Type of ARP (Address Resolution Protocol):<br>ARPA = Ethernet-type ARP<br>SNAP = RFC1042 ARP<br>Probe = HP Probe Protocol |

## *Displaying SLIP Line Status*

The **show line** EXEC command displays SLIP status for a line running in SLIP mode. Enter this command at the EXEC prompt:

**show line** *line-number*

The argument *line-number* specifies the line.

Following is sample output:

```
Tty Typ    Tx/Rx    A Modem  Roty AccO AccI  Uses    Noise
   0 CTY             -    -      -    -    -   1       0
   1 SLIP 9600/9600  -    -      -    -    -   0       0
   2 TTY  9600/9600  -    -      1    -    -   0       0
   3 TTY  9600/9600  - inout     1    -    -   0       0
   4 TTY  9600/9600  -    -      -    -    -   0       0
*  5 TTY  9600/9600  -    -      -    -    -   1       0
   6 SLIP 9600/9600  -    -      -    -    -   0       0
   7 TTY  9600/9600  -    -      -    -    -   0       0
  10 TTY  9600/9600  -    -      -    -    -   0       0
  11 TTY  9600/9600  -    -      -    -    -   0       0
  12 TTY  9600/9600  - inout     -    -    -   0       0
  13 TTY  9600/9600  -    -      -    -    -   0       0
  14 TTY  9600/9600  -    -      -    -    -   0       0
  15 TTY  9600/9600  -    -      -    -    -   0       0
  16 TTY  9600/9600  -    -      -    -    -   0       0
  17 TTY  9600/9600  -    -      -    -    -   0       0
```

## *Displaying the Status of SLIP-Configured Lines*

The **show slip** EXEC command displays the status of all lines configured for SLIP support. Enter this command at the EXEC prompt:

**show slip**

Following is sample output. Table 1-2 describes the fields displayed by this command.

```
Slip statistics:
  Rcvd: 0 packets, 0 bytes, 0 escapes
        0 format errors, 0 checksum errors, 0 overrun, 0 no buffer
  Sent: 0 packets, 0 bytes, 0 escapes, 0 dropped

Tty Mod   Address      Istate  Ostate   Qd InPack OutPac Inerr Dropped
MTU Qsz
 35 i/o 192.31.7.52   IDLE   IDLE     0     0     0    0     0 1524   0
 36 i/o 192.31.7.53   IDLE   IDLE     0     0     0    0     0 1524   0
 37 i/o 192.31.7.54   IDLE   IDLE     0     0     0    0     0 1006   0
 40 i/o 192.31.7.55   IDLE   IDLE     0     0     0    0     0 1006   0
```

*Table 1-2* SLIP Statistics Display Field Descriptions

| Field | Description |
|---|---|
| Rcvd: | Statistics on packets received. |
| packets | Packets received. |
| bytes | Total number of bytes. |
| escapes | Count of escape characters received. |
| format errors | Packets with a bad IP header, even before the checksum is calculated. |
| checksum errors | Count of checksum errors. |
| overrun | Number of giants received. |
| no buffer | Number of packets received when no buffer was available. |
| Sent | Statistics on packets sent. |
| packets | Packets sent. |
| bytes | Total number of bytes. |
| escapes | Count of escape characters sent. |
| dropped | Number of packets dropped. |
| Tty mod | Type of modem control. |
| IState and OState | Used by Cisco engineers for troubleshooting. |
| Qd | Number of packets on hold queue (Qsz is max). |
| InPack | Number of packets input for asynchronous line. |
| OutPac | Number of packets sent to asynchronous line. |
| Inerr | Number of total input errors; sum of format errors, checksum errors, overruns and no buffers. |
| Dropped | Number of packets received that would not fit on the hold queue (time to either fix the hosts’ TCPs, or increase the hold queue size). |
| * | An asterisk indicates a line currently in SLIP mode. |

## *Displaying SLIP BootP Parameters*

The **show async-bootp** EXEC command displays the parameters that have been configured for SLIP extended BootP requests. Enter this command at the EXEC prompt:

**show async-bootp**

Following is sample output:

```
The following extended data will be sent in BOOTP responses:

bootfile (for address 128.128.1.1) "pcboot"
bootfile (for address 131.108.1.111) "dirtboot"
subnet-mask 255.255.0.0
time-offset -3600
time-server 128.128.1.1
```

# *Debugging SLIP*

The privileged-mode EXEC **debug** commands described in this section are used to troubleshoot lines in SLIP mode. Generally, you enter these commands during troubleshooting sessions with Cisco staff.

For each **debug** command, there is a corresponding **undebug** command that turns the display off. Entering the **debug** commands will affect performance of the SLIP feature.

### debug slip

The **debug slip** EXEC command enables logging of all SLIP activity. The high volume of SLIP debugging output, which amounts to several lines per packet, noticeably affects overall system performance.

### debug slip-event

The **debug slip-event** EXEC command enables logging of selected SLIP events, such as various types of errors, enabling and disabling of SLIP mode on a line, and so on. The volume of output for this command is much lower than that for the **debug slip** command.

*Note:* For complete SLIP debugging output, use both the **debug slip** and the **debug slip-event** commands.

## *SLIP Global Configuration Command Summary*

This section provides an alphabetically arranged summary of the SLIP global configuration commands.

**[no] async-bootp** *tag* **[***:hostname***]** *data...*

Specifies extended BootP requests as defined in RFC1084. The argument *tag* is the item being requested, and is one of the following:

- **bootfile**—Specifies use of a server boot file from which to download the boot program. Use the optional *:hostname* and *data* arguments to specify the file name.
- **subnet-mask** *mask*—Dotted decimal address specifying the network and local subnetwork mask (as defined by RFC950).
- **time-offset** *offset*—A signed 32-bit integer specifying the time offset of the local subnetwork in seconds from Coordinated Universal Time (UTC).
- **gateway** *address*—Dotted decimal address specifying the IP addresses of N/4 gateways for this subnetwork. A preferred gateway should be listed first.
- **time-server** *address*—Dotted decimal address specifying the IP address of N/4 time servers (as defined by RFC868).
- **IEN116-server** *address*—Dotted decimal address specifying the IP address of N/4 name servers (as defined by IEN 116).
- **DNS-server** *address*—Dotted decimal address specifying the IP address of N/4 Domain Name Servers (as defined by RFC1034).
- **log-server** *address*—Dotted decimal address specifying the IP address of N/4 MIT-LCS UDP log server.
- **quote-server** *address*—Dotted decimal address specifying the IP address of N/4 Quote of the Day servers (as defined in RFC865).
- **lpr-server** *address*—Dotted decimal address specifying the IP address of N/4 Berkeley UNIX Version 4 BSD servers.
- **impress-server** *address*—Dotted decimal address specifying the IP address of N/4 Impress network image servers.
- **rlp-server** *address*—Dotted decimal address specifying the IP address of N/4 Resource Location Protocol (RLP) servers (as defined in RFC887).
- **hostname** *name*—The name of the client, (which may or may not be domain qualified, depending upon the site).
- **bootfile-size** *value*—A two-octet value specifying the number of 512 octet (byte) blocks in the default boot file.

The optional argument *:hostname* indicates that this entry applies only to the host specified. The argument *:hostname* accepts both an IP address and logical host name. The argument *data* can be a list of IP addresses entered in dotted decimal notation or as logical host names, a number, or a quoted string. Use the **no async-bootp** command to clear the list.

## *SLIP Line Configuration Subcommand Summary*

Following is a list of alphabetically arranged SLIP line configuration subcommands. You are also referred to the section “Configuring Asynchronous Lines” in the “System Configuration” chapter for additional information about line configuration.

**no slip**

Cancels SLIP support on the line.

**slip access-class** *number* **{in | out}**

Configures an access list to be used on packets to or from the SLIP host. The argument *number* is the IP access list number. The keyword **in** configures a list for packets from the SLIP host; keyword **out** compares the IP source address against the access list, and only those packets allowed by the access list are transmitted on the asynchronous line.

**slip address** *internet-address*

Specifies the Internet address assigned to the SLIP client at the other end of the serial line connection. The argument *internet-address* must be on the same network or subnet as the terminal server’s network interface.

**slip address dynamic [***IP-address***]**

Without an IP address, allows the IP address associated with a SLIP line to be assigned upon access. This subcommand allows flexibility when the lines are in a rotary configuration, when you cannot predict which line a user will be connected to. This feature is supported when a TACACS server is used.

With an IP address, allows a default address to be specified upon access.

**slip dedicated**

Places the line in SLIP mode permanently. The terminal server will not create an EXEC on this line, so it is not available for normal interactive use.

**slip hold-queue** *packets*

Specifies the limit of the SLIP output queue, which stores packets received from the network waiting to be sent to the SLIP client. The argument *packets* is the maximum number of packets. Default is two packets.

### slip interactive

Allows the line to be used in either SLIP mode or interactive mode. Hanging up the modem or clearing the line puts the line back into interactive mode.

### slip mtu *bytes*

Specifies the size of the largest Internet packet that the SLIP support can handle. The argument *bytes* is the maximum number of bytes. Default is 1500 bytes