CHAPTER 14

SLIP and PPP Configuration Commands

SLIP and PPP define methods of sending Internet Protocol (IP) packets over standard RS-232 asynchronous serial lines with minimum line speeds of 1200 baud.

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

Use the commands in this chapter to configure SLIP and PPP on your communication server. For configuration information and examples, refer to the *Communication Server Configuration Guide*.

See the *Communication Server and Protocol Translator Connection Guide* for information about SLIP and PPP user-level EXEC connection commands.

async-bootp

To support the extended BootP request specified in RFC 1084, and to specify information that will be sent in response to BootP requests, use the **async-bootp** global configuration command. Use the **no async-bootp** command to clear the list.

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

Syntax Description

tag The item being requested; information the communication server will send in response to BootP requests from clients. The *tag* argument is one of the keyword-argument pairs listed in Table 1-1. Arguments are expressed as filenames, integers, or IP dotted decimal addresses.

:*hostname* An optional argument that indicates that the entry applies only to the specified host. The argument can be either an IP address or a logical host name.

data Data for BootP responses. Can be a list of IP addresses entered in dotted decimal notation or as logical host names, a number, or a quoted string to support the request.

Table 1-1 lists the keyword and argument pairs that can be used for the argument *tag* in the async-bootp command.

Table 1-1 Supported Extended BootP Requests

Keyword and Argument Pair	Use
bootfile	Server boot file from which to download the boot program. Use the optional *:hostname* and *data* arguments to specify the host or hosts.
subnet-mask *mask*	Dotted decimal address specifying the network and local subnetwork mask (as defined by RFC 950).
time-offset *offset*	A signed 32-bit integer specifying the time offset of the local subnetwork in seconds from Coordinated Universal Time.
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 RFC 868).
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 RFC 1034).
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 RFC 865).
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.

Keyword and Argument Pair	Use
rlp-server *address*	Dotted decimal address specifying the IP address of Resource Location Protocol (RLP) servers (as defined in RFC 887).
hostname *name*	Name of the client (which might or might not be domain qualified, depending upon the site).
bootfile-size *value*	Two-octet value specifying the number of 512 octet (byte) blocks in the default boot file.

Default

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

Command Mode

Global configuration

Usage Guidelines

Each of the *tag* keyword-argument pairs is a field that can be filled in and sent in response to BootP requests from clients.

BootP supports the extended BootP requests specified in RFC 1084 and works for both SLIP and PPP encapsulation.

Use the **show async bootp** EXEC command to list the configured parameters. BootP works for both SLIP and PPP.

Examples

The following example illustrates how to specify different boot files: one for a PC and one for a Macintosh. With this configuration, a BootP request from the host on 128.128.1.1 results in a reply listing the boot filename as pcboot. A BootP request from the host named mac results in a reply listing the boot filename as macboot.

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

The following example specifies a subnet mask of 255.255.0.0:

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

The following example specifies a negative time offset of the local subnetwork of -3600 seconds:

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

The following example specifies the IP address of a time server:

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

async default ip address

To set the address used on the remote (PC) side, use the **async default** interface configuration command. Use the **no** form of the command to remove the default address from your configuration.

async default ip address *address*
no async default ip address

Syntax Description

address	Address of the client interface.

Default

None

Command Mode

Interface configuration

Example

The following example specifies address 182.32.7.51 on interface async 6:

```
line 20
speed 19200
interface async 6
async default ip address 182.32.7.51
```

Related Command

async dynamic address

async dynamic address

To specify dynamic asynchronous addressing, use the **async dynamic address** interface configuration command. Use the **no** form of this command to disable dynamic addressing.

async dynamic address
no async dynamic address

Syntax Description

There are no keywords or arguments in this command.

Default

None

Command Mode

Interface configuration

Usage Guidelines

You can control whether addressing is dynamic (the user specifies the address at the EXEC level when making the connection), or whether default addressing is used (the address is forced by the system). If you specify dynamic addressing, the communication server must be in interactive mode and the user will enter the address at the EXEC level.

It is common to configure an asynchronous interface to have a default address and to allow dynamic addressing. With this configuration, the choice between the default address or a dynamic addressing is made by the user when they enter the **slip** or **ppp** EXEC command. If the user enters an address, it is used, and if the user enters the **default** keyword, the default address is used.

Example

The following example shows dynamic addressing assigned to interface async 6.

```
Interface ethernet 0
ip address 1.0.0.1 255.0.0.0
interface async 6
async dynamic address
```

Related Commands

async default ip address

async dynamic routing

To allow the use of routing protocols on an interface, use the **async dynamic routing** interface configuration command. The **no** form of this command disables the use of routing protocols.

async dynamic routing
no async dynamic routing

Syntax Description

There are no arguments to this command.

Default

None

Command Mode

Interface configuration

Usage Guidelines

The use of routing protocols is further controlled by the use of the **/routing** keyword in the **slip** and **ppp** EXEC command. See the *Communication Server and Protocol Translator Connection Guide* for more information about making SLIP and PPP connections.

Example

The following example shows how to enable asynchronous routing on asynchronous interface 6. The **ip tcp header-compression passive** command enables Van Jacobson TCP header compression and prevents transmission of compressed packets until a compressed packet arrives from the asynchronous link.

```
interface async 6
async dynamic routing
async dynamic address
async default ip address 1.1.1.2
ip tcp header-compression passive
ip unnumbered ethernet 0
```

Related Commands

async dynamic address
ip tcp header-compression

async mode dedicated

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

async mode dedicated
no async mode

Syntax Description

This command has no arguments or keywords.

Default

By default, asynchronous mode is disabled.

Command Mode

Interface configuration

Usage Guidelines

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 communication server 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 Internet 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 182.32.7.51
async mode dedicated
encapsulation slip

line 20
location Joe's computer
stopbits 1
speed 19200
!
```

Related Command

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. Use the **no** form of this command to prevent users from implementing SLIP and PPP at the EXEC level.

async mode interactive
no async mode

Syntax Description

There are no arguments to this command.

Default

By default, asynchronous mode is disabled.

Command Mode

Interface configuration

Usage Guidelines

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 on, according to the configuration.

Example

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

```
interface async 6
async default ip address 182.32.7.51
async mode interactive
ip unnumbered ethernet 0
```

Related Command

async mode dedicated

clear line

To return a line to its idle state, enter the **clear line** privileged EXEC command at the system prompt.

clear line *line-number*

Syntax Description

line-number	The asynchronous line port number assigned with the **interface async** command.

Default

None

Command Mode

Privileged EXEC

Usage Guidelines

Normally, this command returns the line to its conventional function as a terminal line, with the interface left in a “down” state.

Example

The following example shows how to use the **clear line** command to return serial interface number 5 to its idle state:

```
CS# clear line 5
```

debug async

To debug asynchronous interfaces, use the **debug async** privileged EXEC command. The **undebug** command turns off the debugging function.

debug async {**framing** | **state** | **packet**s}
undebug

Syntax Description

framing Displays errors that have occurred in the framing of asynchronous packets. Includes CRC errors and illegal sequence errors, as well as packet length errors.

state Displays changes in the asynchronous interface state, such as protocol mode being turned on and off.

packets Displays log message for each input and output packet on asynchronous interfaces. This keyword creates a lot of output; use with care.

Default

Disabled

Command mode

Privileged EXEC

debug ppp

To debug PPP, use the **debug ppp** privileged EXEC command. The **undebug** command turns off the debugging function.

debug ppp {**negotiation** | **error** | **packet** | **chap**}
undebug

Syntax Description

negotiation	Debugs the PPP protocol negotiation process.
error	Displays PPP protocol errors and error statistics.
packet	Displays PPP protocol messages sent and received.
chap	Displays errors encountered during remote or local system authentication.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

The command **debug ppp packet** creates a lot of output. Use with care.

encapsulation

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

encapsulation {**slip** | **ppp**}
no encapsulation {**slip** | **ppp**}

Syntax Description

slip	Specifies SLIP encapsulation for an interface configured for dedicated asynchronous mode or DDR.
ppp	Specifies PPP encapsulation for an interface configured for dedicated asynchronous mode or DDR.

Default

SLIP encapsulation is enabled by default.

Command Mode

Interface configuration

Usage Guidelines

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 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 **debug ppp negotiations** debugging 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 keepalives to detect line failure. The **no keepalive** command can be used to disable echo requests. See the“Configuring IP Routing Protocols” chapter of this publication and the *Communication Server Configuration Guide* for more information about the **no keepalive** command.

In order to use SLIP or PPP, the communication server 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, asynchronous interface 1 is configured for PPP encapsulation.

```
tarmac# config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
tarmac(config)# interface async1
tarmac(config-if)# encapsulation ppp
```

Related Commands

The dagger (†) indicates that the command is documented in another chapter.

keepalives†
debug ppp

hold-queue

To limit the size of the IP output queue, use the **hold-queue** interface configuration command. The **no** form of this command turns the output queue to the default size.

hold-queue *packets*
no hold-queue

Syntax Description

packets	Maximum number of packets. The range of values is 0 through 65535.

Default

Ten packets (default for asynchronous interfaces only)

Command Mode

Interface configuration

Usage Guidelines

The default of ten packets allows the communication server to queue a number of back-to-back routing updates. This is the default for asynchronous interfaces only; other media types have different defaults.

The hold queue stores packets received from the network that are waiting to be sent to the client. It is recommended that the queue size not exceed ten packets on asynchronous interfaces. For most other interfaces, queue length should not exceed 100.

Example

The following example changes the packet queue length of a line to five packets:

```
interface async 2
async default ip address 182.32.7.5
hold-queue 5
```

interface

To specify the interface you want to configure, use the **interface** global configuration command command. The **no** form of the command clears the interface configuration.

interface *type unit*
no interface

Syntax Description

type	Specifies the type of interface.
unit	Specifies interface number. A number from 1 to 8 for a 508-CS, 1-16 for the 516-CS, and 1 to 113 on a fully configured ASM-CS.

Default

None

Command Mode

Global configuration

Usage Guidelines

The syntax **interface async***unit* is also accepted, with no space between the type of interface and the interface number.

Example

The following example specifies interface async 1:

```
interface async 1
```

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. Use the **no ip access-group** command to disable control over packets transmitted to or from an asynchronous host.

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

Syntax Description

access-list-number — Specifies the assigned IP access list number.

in — Defines access control on packets transmitted *from* the asynchronous host.

out — Defines access control on packets being sent *to* the asynchronous host.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

With this command in effect, the IP destination address of each packet is run through the access list for acceptability and 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 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
!
! Specify the access list
interface async 6
async dynamic address
ip access-group 1 out
ip access-group 2 in
!
```

ip address

Use the **ip address secondary** interface configuration command to set multiple IP addresses for an interface. Use the **no ip address secondary** interface configuration command to remove the specified addresses.

ip address *address mask* [**secondary**]
no ip address *address mask* [**secondary**]

Syntax Description

address	The IP address.
mask	The network mask for the associated IP network.
secondary	Specifies additional IP addresses.

Default

None

Command Mode

Interface configuration

Usage Guidelines

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. Communication servers respond 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, 131.108.1.27 is the primary address and 192.31.7.17 and 192.31.8.17 are secondary addresses for async 1:

```
interface async 1
ip address 131.108.1.27 255.255.255.0
ip address 192.31.7.17 255.255.255.0 secondary
ip address 192.31.8.17 255.255.255.0 secondary
```

ip mtu

To specify the size of the largest Internet packet, use the **ip mtu** interface configuration command. Use the **no** form of this command to return to the default MTU size of 1500 bytes.

ip mtu *bytes*
no ip mtu

Syntax Description

bytes	The maximum number of bytes. The range of values is 64 to 1000000.

Default

1500 bytes

Command Mode

Interface configuration

Example

The following example sets the packet MTU size to 200 bytes:

```
interface async 5
async default ip address 182.32.7.5
ip mtu 200
```

ip tcp header-compression

To configure Van Jacobson TCP header compression on the asynchronous link, use the **ip tcp header-compression** line configuration command. The **no** form of this command disables header compression.

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

Syntax Description

on	Turns header compression on.
off	Turns header compression off.
passive	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 **on** option.

Default

By default, header compression is on.

Command Mode

Interface configuration

Usage Guidelines

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 it 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 compression status used by the host at the other end of the asynchronous line.

```
Server> slip 1.0.0.1
Password:
Entering SLIP mode.
Interface IP address is 1.0.0.1, 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." See the *Communication Server and Protocol Translator Connection Guide* for more information about making SLIP and PPP connections.

Example

The following example illustrates how to enable 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

See documentation of these commands in the *Communication Server and Protocol Translator Connection Guide*:

slip
ppp
slip /compressed
slip default

ip unnumbered

To conserve network resources, use the **ip unnumbered** line configuration command. The **no** form of this command disables unnumbered interfaces.

ip unnumbered *interface-name*
no ip unnumbered

Syntax Description

interface-name	The name of another interface on which the communication server has an assigned IP address.

Default

None

Command Mode

Interface configuration

Usage Guidelines

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:

- 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 a runnable image over an unnumbered serial interface.
- The argument *interface-name* must be the name of another interface in the network server that has an IP address, not another unnumbered interface.

Example

The following example shows how to configure asynchronous serial interface 6 as unnumbered:

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

Related Command

ip address

show async bootp

To display the parameters that have been configured for extended BootP requests, use the **show async bootp** privileged EXEC command.

show async bootp

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Sample Display

The following is sample output from the **show async bootp** command.

```
sloth# show async bootp

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
```

Table 1-2 describes significant fields shown in the display.

Table 1-2 Show Async BootP Field Descriptions

Field	Description
bootfile... “pcboot”	Indicates that the boot file for address 128.128.1.1 is named pcboot.
subnet-mask 255.255.0.0	Specifies the subnet mask.
time-offset -3600	Indicates that the local time is one hour (3600 seconds) earlier than Coordinated Universal Time (UTC).
time-server 128.128.1.1	Indicates the address of the time server for the network.

show async status

To display the status of activity on all lines configured for asynchronous support, use the **show async status** privilged EXEC command.

show async status

Syntax Description

This command has no arguments or keywords.

Command Mode

Privileged EXEC

Usage Guidelines

The display resulting from this command shows all asynchronous sessions, whether they are using SLIP or PPP encapsulation.

Sample Display

The following is sample output from the **show async status** command:

```
cs> show async status
Async protocol statistics:
  Rcvd: 5448 packets, 7682760 bytes
        1 format errors, 0 checksum errors, 0 overrun, 0 no buffer
  Sent: 5455 packets, 7682676 bytes, 0 dropped


 Tty           Local          Remote Qd InPack OutPac Inerr  Drops  MTU Qsz
   1     192.31.7.84         Dynamic  0      0      0     0      0 1500  10
*  3     192.31.7.98            None  0   5448   5455     1      0 1500  10
```

Table 1-3 describes significant fields shown in the display.

Table 1-3 Asynchronous Statistics Display Field Descriptions

Field	Description
Rcvd:	Statistics on packets received
5448 packets	Packets received
7682760 bytes	Total number of bytes
1 format errors	Spurious characters received when a packet start delimiter is expected.
0 checksum errors	Count of checksum errors
0 overrun	Number of giants received
0 no buffer	Number of packets received when no buffer was available
Sent	Statistics on packets sent
5455 packets	Packets sent
7682676 bytes	Total number of bytes
0 dropped	Number of packets dropped

Field	Description
Tty	Line number
*	Line currently in use
Local	Local IP address on the link
Remote	Remote IP address on the link; “Dynamic” indicates that a remote address is allowed but has not been specified; “None” indicates that no remote address is assigned or being used
Qd	Number of packets on hold queue (Qsz is the maximum)
InPack	Number of packets received
OutPac	Number of packets sent
Inerr	Number of total input errors; sum of format errors, checksum errors, overruns and no buffers
Drops	Number of packets received that would not fit on the hold queue
MTU	Current maximum transmission unit size
Qsz	Current output hold queue size

show line

Use the **show line** privileged EXEC command to display connection status for a line running in asynchronous mode.

show line [*line-number*]

Syntax Description

line-number	Specifies the particular line about which information will be displayed. If you do not specify a line number, information about all lines is displayed.

Command Mode

EXEC

Sample Display

```
The following is sample output from the show line command:
mosey> show line
 Tty Typ    Tx/Rx    A Modem  Roty AccO AccI  Uses    Noise   Overruns
*  0 CTY             -    -      -    -    -     0        0        0/0
A  1 TTY  9600/9600  -    -      -    -    1     0        0        0/0
   2 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   3 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   4 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   5 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   6 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   7 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   8 TTY  9600/9600  -    -      -    -    -     0        0        0/0
   9 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  10 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  11 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  12 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  13 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  14 TTY  9600/9600  -    -      -    -    -     0        0        0/0
15 TTY  9600/9600  -    -      -    -    -     0        0        0/0
  16 TTY  9600/9600  -    -      -    -    -     0        0        0/0
* 17 VTY  9600/9600  -    -      -    -    -    18        0        0/0
```

- An asterisk (*) preceding the number in the Tty field means the line is currently active, running a terminal-oriented protocol.
- An “A” preceding the number indicates the line is currently active in asynchronous mode.
- An “I” preceding the number indicates that the line is free and it can be used for asynchronous mode because it is configured for **async mode interactive**.

Table 1-4 describes significant fields shown in the display.

Table 1-4 Show Line Field Descriptions

Tasks	Descriptions
Tty	Indicates the absolute line number of the specified line.
Type	Type of line. Possible values follow: CTY—Console AUX—Auxiliary port TTY—Asynchronous terminal port VTY—Virtual terminal LPT—Parallel printer
Tx/Rx	Transmit rate of the line (baud)/receive rate of the line (baud).
A	Indicates whether or not autobaud has been configured for the line. A value of F indicates that autobaud has been configured; a hyphen (-) indicates that it has not been configured for the line.
Modem	Types of modem signal that has been configured for the line. Possible values include: callin callout cts-req DTR-Act inout RIisCD
Roty	Rotary Group configured for this line.
AccO	Output access list number configured for the specified line.
AccI	Input access list number configured for the specified line.
Uses	Number of connections established to or from this line since the system was restarted.
Noise	Number of times noise has been detected on the line since the system restarted.
Overruns	Hardware (UART) overruns/software buffer overflows, both defined as the number of overruns or overflows that have occurred on the specified line since the system was restarted. Hardware overruns are buffer overruns; the UART chip has received bits from the software faster than it can process them. A software overflow occurs when the software has received bits from the hardware faster than it can process them.

The following display shows **show line** output for a specified line:

```
cs> show line 1

 Tty Typ    Tx/Rx    A Modem  Roty AccO AccI  Uses    Noise Overruns
   1 TTY  9600/9600  -    -      -    -   10     0        0        0

Line 1, Location: "charnel console", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600, no parity, 2 stopbits, 8 databits
Status: Ready, Hardware XON/XOFF
Capabilities: none
Modem state: Ready
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               0:10:00        never         0:00:15      not imp   not set
Session limit is not set.
Allowed transports are telnet lat rlogin.  Preferred is lat
No output characters are padded
Characters causing immediate data dispatching:
   Char    ASCII
Group codes:    0
```

Related Commands

async dynamic routing
ip tcp header-compression
async dynamic address