Overview of Interfaces, Controllers, and Lines Used for Dial Access

This chapter describes the different types of calls, interfaces, software constructs, controllers, channels, and lines that are used for dial-up remote access.

The following sections are provided in this chapter:

- Call Types
- Cisco IOS Dial Universe
- Logical Constructs
- Logical Interfaces
- T1 and E1 Controllers
- Channelized T1 and Channelized E1 (non-ISDN)
- B Channels and D Channels
- Line Types

Call Types

Three basic call types are used for dial access. These call types are discussed in the following sections:

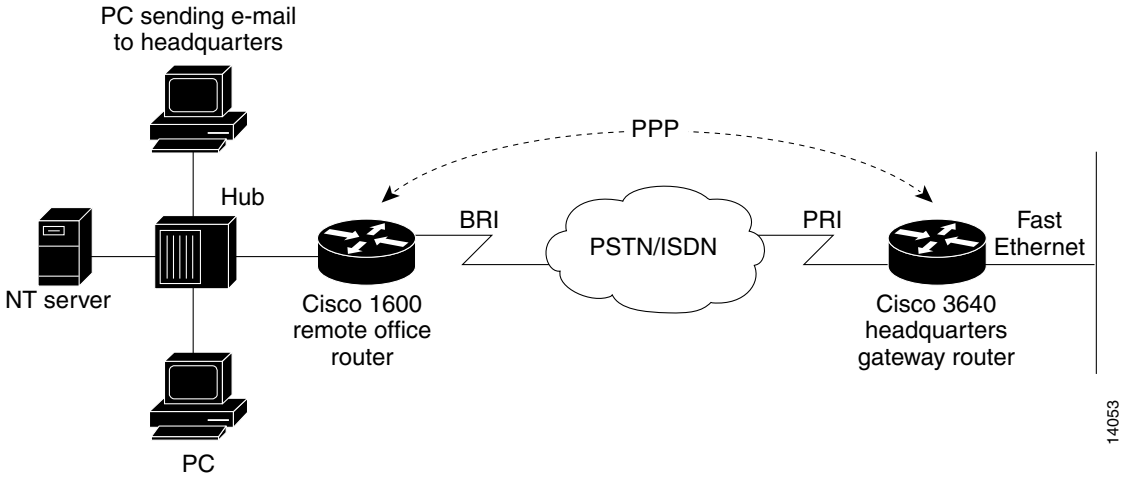
- Circuit Switched Digital Calls
- Analog Modem Calls
- Asynchronous Character Stream Calls

Circuit Switched Digital Calls

Circuit switched digital calls are usually ISDN 56-kbps or 64-kbps data calls that use the point-to-point protocol (PPP). These calls are initiated by an ISDN router, access server, or terminal adapter connected to a client workstation. Individual synchronous serial DS0s (B channels) are used to transport circuit switched digital calls across WANs. These calls do not transmit across old world POTS lines.

Figure 192 shows a Cisco 1600 series remote office router dialing into a Cisco 3640 router positioned at a headquarters gateway.

Figure 192 Remote Office LAN Dialing into Headquarters

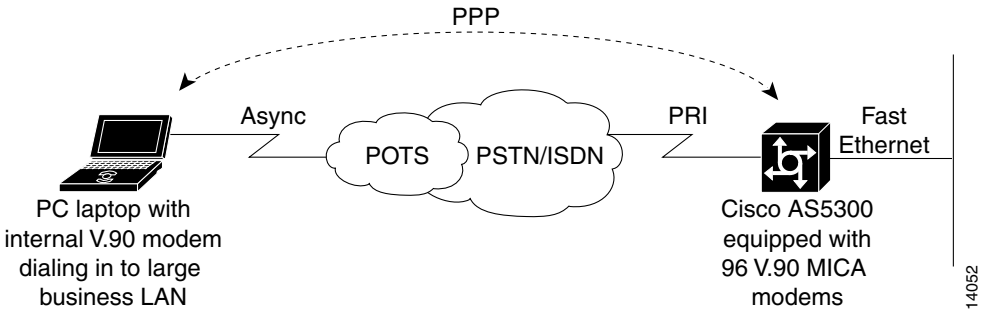


Analog Modem Calls

Analog modem calls travel through traditional telephone lines and ISDN lines. Regardless of the media used, theses calls are initiated by a modem and terminate on another modem at the remote end.

Figure 193 shows a remote laptop using a V.90 internal modem to dial into a Cisco AS5300 access server, which is loaded with 96 internal V.90 MICA modems.

Figure 193 Remote Node Dialing into a Cisco AS5300



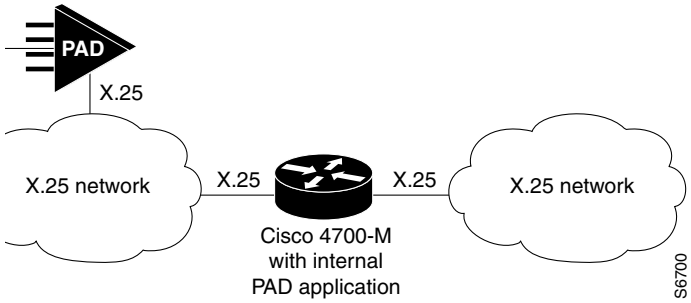
Asynchronous Character Stream Calls

Asynchronous character stream calls enter the router or access server through virtual terminal (VTY) lines and virtual asynchronous interfaces (VTY-async). These virtual lines and interfaces terminate incoming character streams that do not have a physical connection to the access server or router (such as a physical serial interface). For example, if you begin a PPP session over an asynchronous character stream, a VTY-async interface is created to support the call. The following types of calls are terminated on a virtual asynchronous interface: Telnet, LAT, V.120, TN3270, and PAD calls.

Figure 194 shows a dumb terminal using a modem and PAD to place a call in to an X.25 switched network. The Cisco 4700-M router is configured to support VTY lines and virtual asynchronous interfaces.

Figure 194 Standard X.25 Dial-Up Connection

erminal
a connection
X.25 host

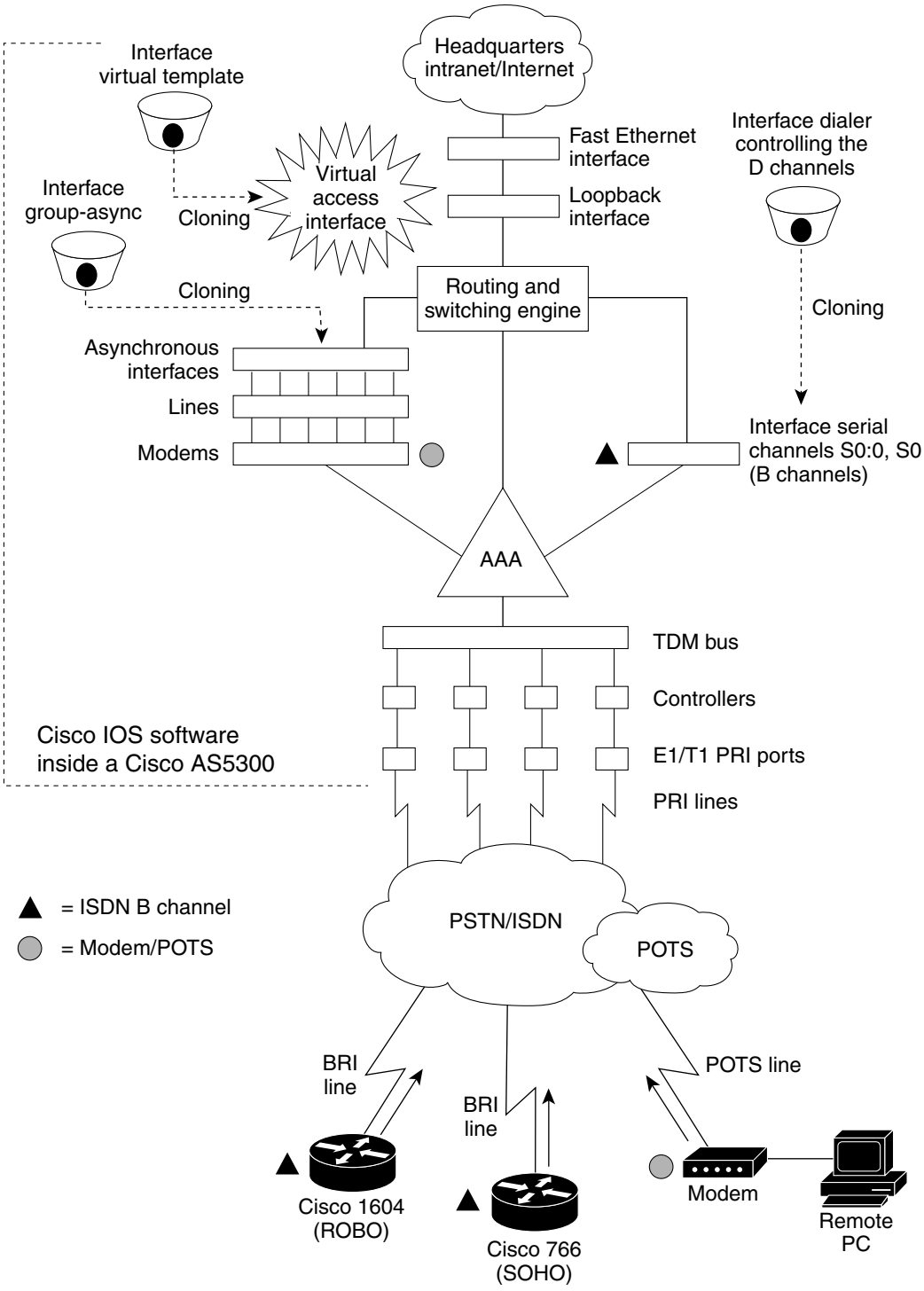


Cisco IOS Dial Universe

Different components inside the Cisco IOS software terrain work together to enable remote clients to dial-in and transmit packets. Figure 195 shows one Cisco AS5300 receiving calls from a remote office/branch office (ROBO), small office/home office (SOHO), and modem client.

Depending on your network scenario, you may or may not encounter all of the components in Figure 195. For example you might decide to create a virtual IP subnet by using a loopback interface. This step saves address space. Virtual subnets can exist inside devices that you advertise to your backbone. In turn, IP packets get relayed to remote PCs which route back to the central site.

Figure 195 Cisco IOS Dial Universe



For more information about the components in Figure 195, see the following sections:

- Logical Constructs
- Logical Interfaces
- T1 and E1 Controllers
- Channelized T1 and Channelized E1 (non-ISDN)
- B Channels and D Channels
- Line Types

Logical Constructs

A logical construct stores core protocol characteristics to assign to physical interfaces. No data packets are forwarded to a logical construct. Cisco uses three types of logical constructs in its access servers and routers:

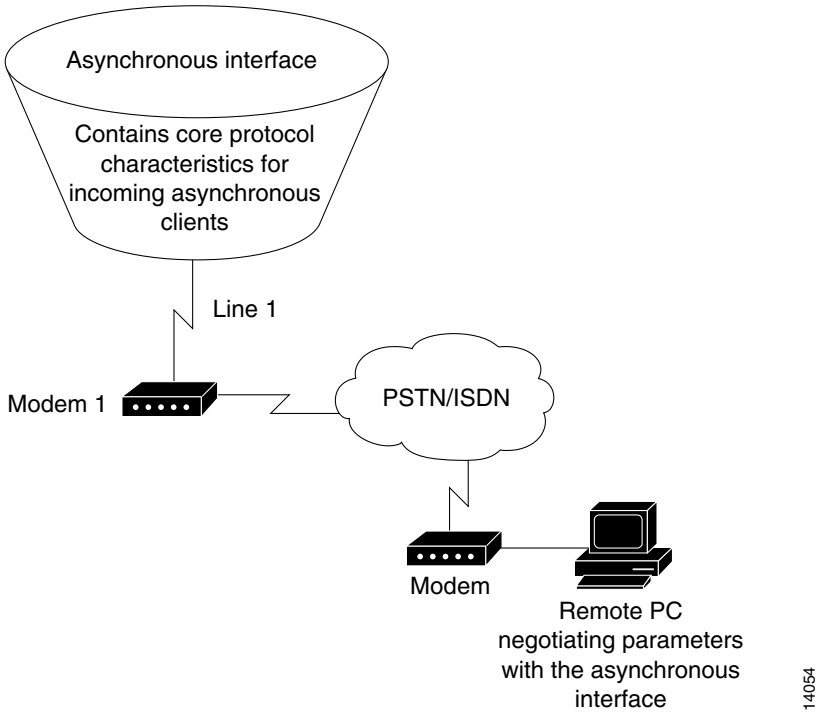
- Asynchronous Interfaces
- Group Asynchronous Interfaces
- Virtual Template Interfaces

Asynchronous Interfaces

An asynchronous interface assigns network protocol characteristics to remote asynchronous clients that are dialing in through physical terminal (TTY) lines and modems. See Figure 196.

Use the **interface async** command to create and configure an asynchronous interface.

Figure 196 Logical Construct for an Asynchronous Interface



To enable clients to dial in, you must configure two asynchronous components: async lines and async interfaces. Asynchronous interfaces correspond to physical terminal (TTY) lines. For example, asynchronous interface 1 corresponds to TTY line 1.

Commands entered in asynchronous interface mode enable you to configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode let you configure the physical aspects for the same port.

Specifically, you configure asynchronous interfaces to support PPP connections. An asynchronous interface on an access server or router can be configured to support the following functions:

- Network protocol support (such as IP, IPX, or AppleTalk)
- Encapsulation support (such as PPP)
- IP client addressing options (default and/or dynamic)
- IPX network addressing options
- PPP authentication
- ISDN BRI and PRI configuration

The following example shows how to configure an asynchronous interface on a Cisco 2509-RJ access server. This access server has nine asynchronous interfaces, as displayed by the **?** command. Only the first interface is configured in this example. The remaining eight are used for a different purpose.

```
as2509# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
as2509(config)# interface async ?
  <1-9>  Async interface number
as2509(config)# interface async 1
as2509(config-if)# description ASYNC LINE 5293731 TO HIGHWAY
as2509(config-if)# ip address 192.168.10.1 255.255.255.0
as2509(config-if)# encapsulation ppp
as2509(config-if)# async default routing
as2509(config-if)# async mode dedicated
as2509(config-if)# dialer in-band
as2509(config-if)# dialer map ip 192.168.10.2 name as2511 broadcast
as2509(config-if)# dialer-group 1
as2509(config-if)# ppp authentication chap
```

For additional information about configuring asynchronous interfaces, see the chapter "Configuring Modem Support and Asynchronous Devices."

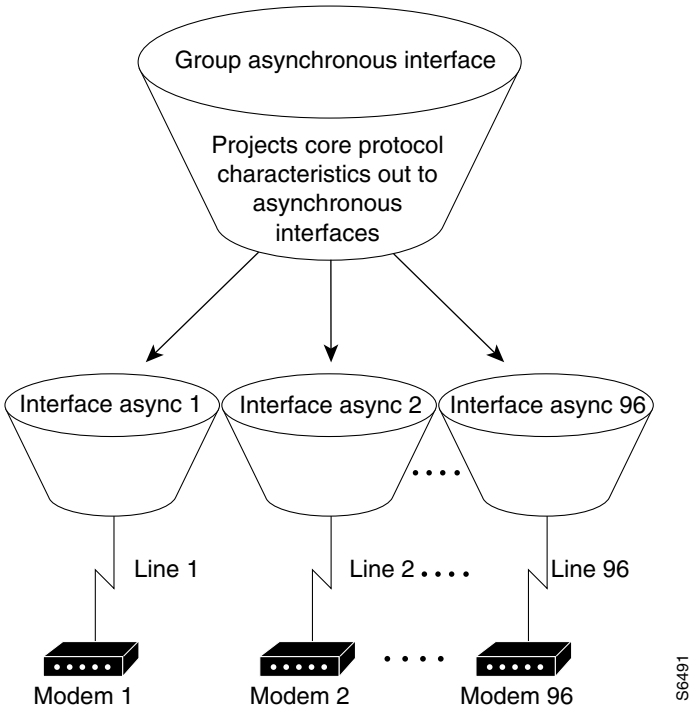
Group Asynchronous Interfaces

A group asynchronous interface is a parent interface that stores and projects core protocol characteristics to a specified range of asynchronous interfaces. Asynchronous interfaces clone protocol information from group asynchronous interfaces. No data packets arrive in a group asynchronous interface.

By setting up a group asynchronous interface, you also eliminate the need to repeatedly configure identical configuration information across several asynchronous interfaces. For example, on a Cisco AS5300 one group asynchronous interface is used instead of 96 individual asynchronous interfaces. See Figure 197.

Use the **interface group-async** command to create and configure a group asynchronous interface.

Figure 197 Logical Construct for Group Asynchronous Interface



The following example shows a group asynchronous configuration for a Cisco AS5300 loaded with one 4-port ISDN PRI card and 96 MICA modems.

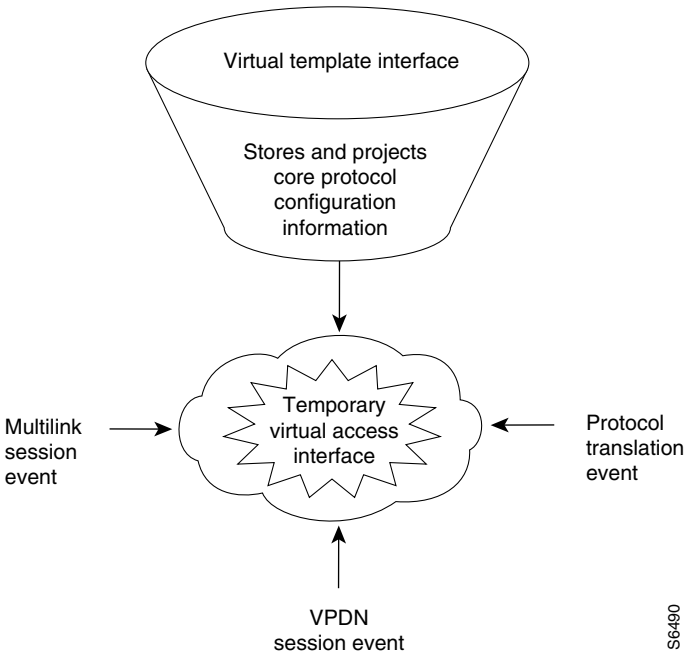
```
as5300(config)# interface group-async 1
as5300(config-if)# ip unnumbered loopback 0
as5300(config-if)# encapsulation ppp
as5300(config-if)# async mode interactive
as5300(config-if)# peer default ip address pool dialin_pool
as5300(config-if)# no cdp enable
as5300(config-if)# ppp authentication chap pap dialin
as5300(config-if)# group-range 1 96

Building configuration...

as5300(config-if)#
```

For more information, see the chapter “Configuring Modem Support and Asynchronous Devices.”

Virtual Template Interfaces

A virtual interface template stores protocol configuration information for virtual access interfaces and protocol translation sessions. See Figure 198.

Figure 198 Logical Construct for a Virtual Template Interface



Templates for Virtual Access Interfaces

Virtual templates project configuration information to temporary virtual access interfaces triggered by multilink or virtual private dial-up network session events. When a virtual access interface is triggered, the configuration attributes in the virtual template are cloned and the negotiated parameters are applied to the connection.

The following example shows a virtual template interface on a Cisco 7206, which is used as a home gateway router in a VPDN scenario.

```
c7206# configure terminal
c7206(config)# interface virtual-template 1
c7206(config-if)# ip unnumbered ethernet 2/1
c7206(config-if)# peer default ip address pool cisco-pool
c7206(config-if)# ppp authentication chap pap
c7206(config-if)# exit
c7206(config)# vpdn enable
c7206(config)# vpdn incoming isp cisco.com virtual-template 1
c7206(config)#
```

For more information, see the chapter “Configuring Virtual Template Interfaces.”

Templates for Protocol Translation

Virtual templates are used to simplify the process of configuring protocol translation to tunnel PPP or SLIP across X.25, TCP, and LAT networks. You can create a virtual interface template with the **interface virtual-template** command and use it for one-step and two-step protocol translation. When a user dials in through a virtual terminal (VTY) line and a tunnel connection is established, the router clones the attributes of the virtual interface template onto a *virtual access interface*. This virtual access interface is a temporary interface that supports the protocol configuration specified in the virtual interface template. This virtual access interface is created dynamically and lasts only as long as the tunnel session is active.

The virtual template in the following example explicitly specifies PPP encapsulation. The translation is from X.25 to PPP, which enables tunneling of PPP across an X.25 network.

```
c7206# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
c7206(config)# interface virtual-template 1
c7206(config-if)# ip unnumbered ethernet 0
c7206(config-if)# peer default ip address 162.18.2.131
c7206(config-if)# encapsulation ppp
c7206(config-if)# exit
c7206(config)# translate x25 5555678 virtual-template 1
c7206(config)#
```

For more information, see the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices."

Logical Interfaces

A logical interface receives and transmits data packets and controls physical interfaces. The Cisco IOS software provides three logical interfaces used for dial access:

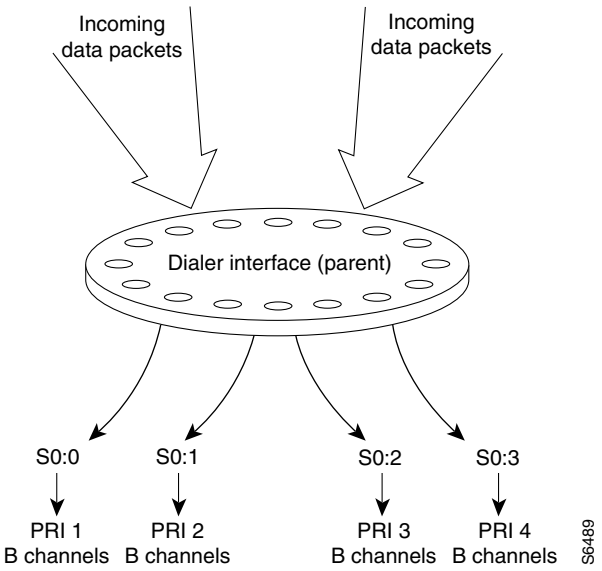
- Dialer Interfaces
- Virtual Access Interfaces
- Virtual Asynchronous Interfaces

Dialer Interfaces

A dialer interface is a parent interface that stores and projects protocol configuration information that is common to all D channels that are members of a dialer rotary group. Data packets pass through dialer interfaces, which in turn initiates dialing for inbound calls. In most cases, D channels get their core protocol intelligence from dialer interfaces.

Figure 199 shows packets coming into a dialer interface, which contains the configuration parameters common to four D channels (shown as S0:0, S0:1, S0:3, and S0:4). All the D channels are members of the same rotary group. Without the dialer interface configuration, each D channel must be manually configured with identical properties. Dialer interfaces condense and streamline the configuration process.

Figure 199 Dialer Interface and Its Neighboring Components



A dialer interface is user configurable and linked to individual B channels where it delivers data packets to their physical destinations. Dialer interfaces seize physical interfaces to cause packet delivery. If a dialer interface engages in a multilink session, a dialer interface is in control of a virtual access interface, which in turn controls S0:3 or chassis 2 S0:3 for example. A dialer interface is created with the **interface dialer** global configuration command.

The following example shows a fully configured dialer interface. All the D channels are members of rotary group 1.

```
as5300# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
as5300(config)# interface dialer 0
as5300(config-if)# ip unnumbered loopback 0
as5300(config-if)# no ip mroute-cache
as5300(config-if)# encapsulation ppp
as5300(config-if)# peer default ip address pool dialin_pool
as5300(config-if)# dialer in-band
as5300(config-if)# dialer-group 1
as5300(config-if)# no fair-queue
as5300(config-if)# no cdp enable
as5300(config-if)# ppp authentication chap pap callin
as5300(config-if)# ppp multilink
as5300(config-if)#
```

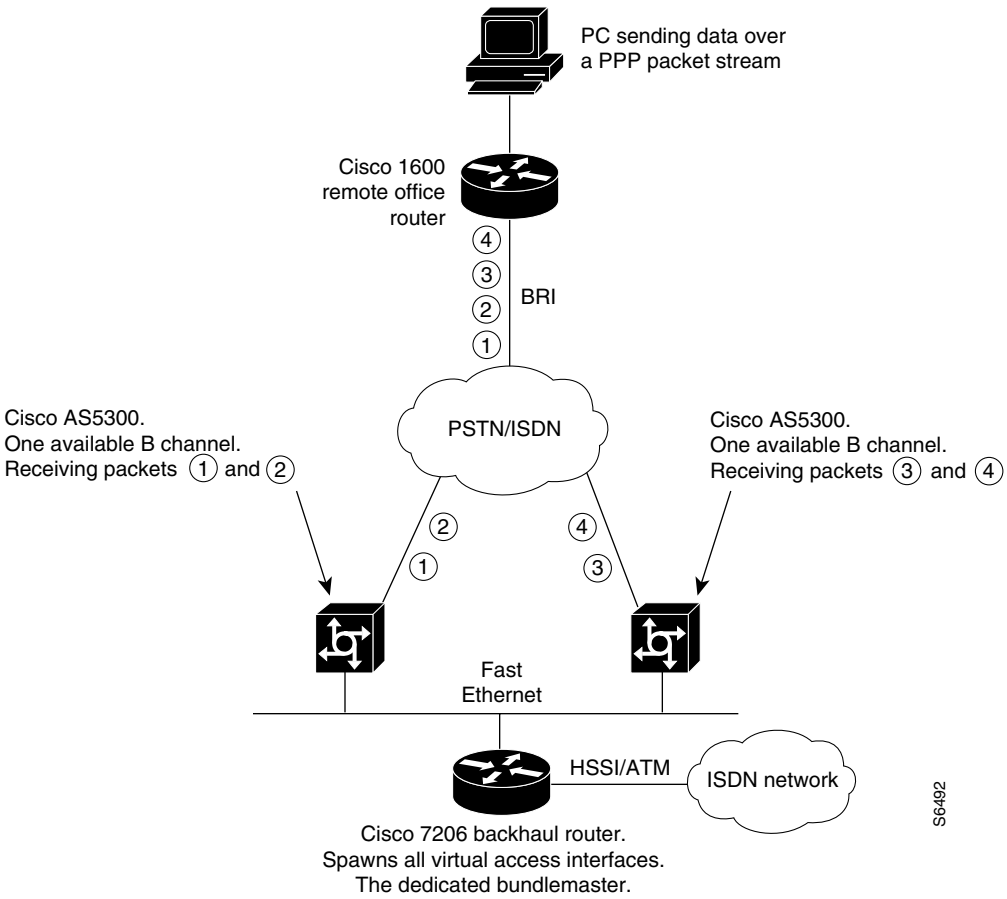
Virtual Access Interfaces

A virtual access interface is a temporary interface that is spawned to terminate incoming PPP steams that do not have physical connections. PPP streams and L2F/L2TP frames coming in on multiple B channels are reassembled on virtual access interfaces. These access interfaces are constructs used to terminate packets.

Virtual access interfaces gets their set of instructions from virtual interface templates. The attributes configured in virtual templates are projected or cloned to a virtual access interfaces. Virtual access interfaces are not directly user configurable. These interfaces are created dynamically on-the-fly and last only as long as the tunnels or multilink sessions are active. After the sessions are ended, the virtual access interfaces disappear.

Figure 200 shows how a virtual access interface functions to accommodate a multilink session event. Two physical interfaces on two different access servers are participating in one multilink call from a remote PC. However, each Cisco AS5300 only has one B channel available to receive a call. All other channels are busy. Therefore all four packets are equally dispersed across two separate B channels and two access servers. Each Cisco AS5300 receives only half the total packets. A virtual access interface is dynamically spawned upstream on a Cisco 7206 backhaul router to receive the multilink protocol, track the multilink frames, and reassemble the packets. The Cisco 7206 is configured to be the bundle master, which performs all packet assembly and reassembly for both Cisco AS5300s.

Figure 200 Virtual Access Interfaces Used for Multichassis Multilink Session Events

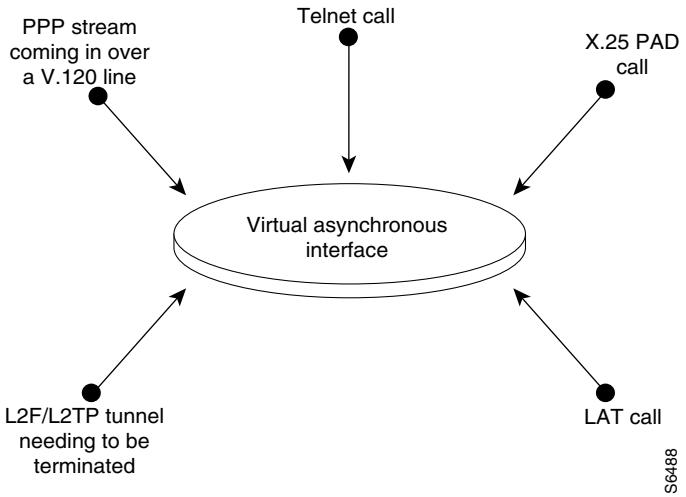


Virtual Asynchronous Interfaces

A virtual asynchronous interface (also known as VTY-async) is created on demand to support calls that enter the router through a non-physical interface. For example, asynchronous character stream calls terminate or land on non-physical interfaces. These types of calls include inbound Telnet, LAT, PPP over character-oriented protocols (such as V.120 or X.25), and PAD calls. A virtual asynchronous interface is also used to terminate L2F/L2TP tunnels, which are often traveling companions with multilink sessions. Virtual asynchronous interfaces are not user configurable; rather, they are dynamically created and torn down on demand. A virtual asynchronous line is used to access a virtual asynchronous interface.

Figure 201 shows a variety of calls terminating on a virtual asynchronous interface. After the calls end, the interface is torn down.

Figure 201 Async Character Stream Calls Terminating on a VTY Async Interface



T1 and E1 Controllers

Cisco controllers negotiate the following parameters between an access server and a central office: line coding, framing, clocking, DS0/timeslot provisioning, and signaling.

Timeslots are provisioned to meet the needs of particular network scenarios. T1 controllers have 24 timeslots. E1 controllers have 30 timeslots. For example to support traffic flow for one ISDN PRI line in a T1 configuration, use the **pri-group** command. To support traffic flow for analog calls over a channelized E1 line with E&M signaling, enter the **cas-group 1 timeslots 1-30 type e&m-fgb** command. Though supported on Cisco controllers, most telephone companies will not support provisioning one trunk for different combinations of timeslot services. For example, timeslots 1-10 running of PRI, timeslots 11-20 running channel associated signaling, and timeslots 21-24 supporting leased-line grouping.

The following example configures one of four T1 controllers on a Cisco AS5300. This example supports modem calls and circuit switched digital calls over ISDN PRI.

```
as5300# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
as5300(config)# controller t1 ?
  <0-3>  Controller unit number
as5300(config)# controller t1 0
as5300(config-controller)# framing esf
as5300(config-controller)# linecode b8zs
as5300(config-controller)# clock source line primary
as5300(config-controller)# pri-group timeslots 1-24
as5300(config-controller)#
```

Channelized T1 and Channelized E1 (non-ISDN)

A channelized T1 or channelized E1 line is an analog line that was originally intended to support analog voice calls, but has evolved to support analog data calls. ISDN does not transmit across channelized T1 or E1 lines. Channelized T1 and channelized E1 line are often referred to CT1 and CE1. These channelized lines are found in old world POTS networks (non ISDN).

The key difference between traditional channelized lines (analog) and non-channelized lines (ISDN) is that channelized lines do not have a built-in D-channel. For example, all 24 channels on a T1 line only carry data. The signaling is in-band or associated to the data channels. Traditional channelized lines do not support digitized data calls (for example, BRI with 2B+D). Channelized lines support a variety of in-band signal types, such as ground start, loop start, wink start, immediate start, E&M and R2.

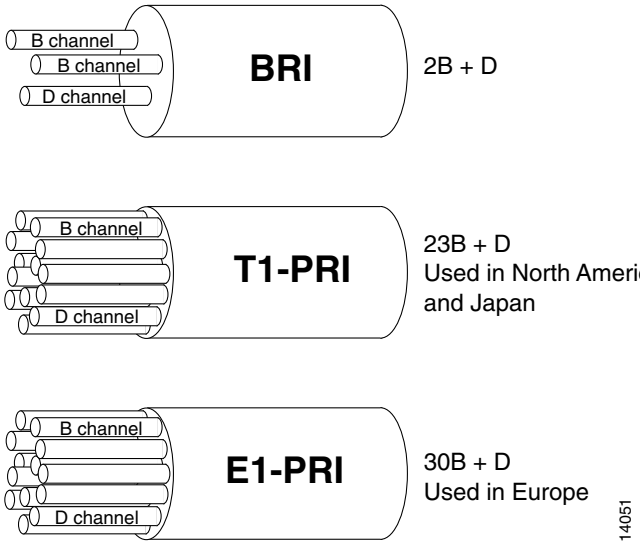
Signaling for channelized lines is configured with the **cas-group** controller configuration command. The following example configures E&M feature group B signaling on a T1 controller:

```
as5300# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
as5300(config)# controller t1 0
as5300(config-controller)# cas-group 1 timeslots 1-24 type ?
  e&m-fgb              E & M Type II FGB
  e&m-fgd              E & M Type II FGD
  e&m-immediate-start  E & M Immediate Start
  fxs-ground-start     FXS Ground Start
  fxs-loop-start       FXS Loop Start
  r1-modified          R1 Modified
  sas-ground-start     SAS Ground Start
  sas-loop-start       SAS Loop Start
as5300(config-controller)# cas-group 1 timeslots 1-24 type e&m-fgb
as5300(config-controller)# framing esf
as5300(config-controller)# clock source line primary
as5300(config-controller)#
```

B Channels and D Channels

Cisco routing devices support ISDN BRI and ISDN PRI, both media types use B channels and D channels. Figure 202 shows how many B channels and D channels are assigned to each media type.

Figure 202 Logical Relationship of B Channels and D Channels



ISDN BRI

ISDN BRI operates over most of the copper twisted-pair telephone wiring in place today. ISDN BRI delivers a total bandwidth of a 144-kbps via three separate channels. Two of the channels, called B (Bearer) channels, operate at 64-kbps and are used to carry voice, video, or data traffic. The third channel, the D (Data) channel, is a 16-kbps signaling channel used to tell the public switched telephone network how to handle each of the B channels. ISDN BRI is often referred to as "2 B+D."

Enter the **interface bri** command to bring up and configure a single BRI interface, which is the overseer of the 2 B+D channels. The D channel is not user configurable.

The following example configures an ISDN BRI interface on a Cisco 1600 series router. The **isdn spid** command defines the service profile identifier (SPID) number, which is assigned by the ISDN service provider, to both B channels. Not all ISDN lines have SPIDs.

```
1600# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
1600(config)# interface bri 0
1600(config-if)# isdn spid1 555987601
1600(config-if)# isdn spid2 555987602
1600(config-if)# ip unnumbered ethernet 0
1600(config-if)# dialer map ip 192.168.37.40 name hq 5552053
1600(config-if)# dialer load-threshold 70
1600(config-if)# dialer-group 1
1600(config-if)# encapsulation ppp
1600(config-if)# ppp authentication chap pap callin
1600(config-if)# ppp multilink
1600(config-if)# no shutdown
```

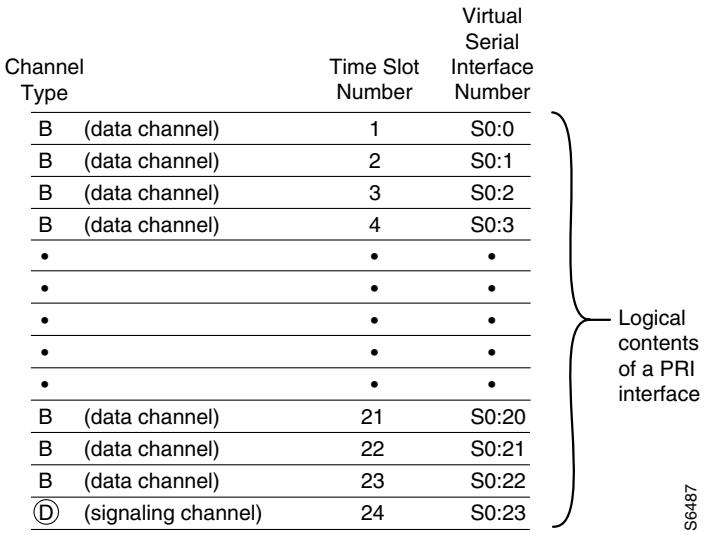
ISDN PRI

ISDN PRI is designed to carry large numbers of incoming ISDN calls at points-of-presences (POPs) and other large central site locations. All the reliability and performance of ISDN BRI applies to ISDN PRI, but ISDN PRI has 23 B channels running at 64-kbps each and a shared 64-kbps D channel that carries signaling traffic. ISDN PRI is often referred to as "23 B+D" (North America and Japan) or "30 B+D" (Europe).

The D channel notifies the central office switch to send the incoming call to particular timeslots on the Cisco access server or router. Each one of the B channels carries data or voice. The D channel carries signaling for the B channels. The D channel identifies if the call is a circuit switched digital call or an analog modem call. Analog modem calls are decoded and then get sent off to the onboard modems. Circuit switched digital calls are directly relayed off to the ISDN processor in the router. Enter the **interface serial** command to bring up and configure the D channel, which is user configurable.

Figure 203 shows the logical contents of a ISDN PRI interface used in a T1 network configuration. The logical contents includes 23 B channels, one D channel, 24 timeslots, and 24 virtual serial interfaces (total number of Bs + D)

Figure 203 Logical Relationship of ISDN PRI Components for T1



The following example is for a Cisco AS5300. It configures one T1 controller for ISDN PRI then goes on to configure the neighboring D channel (interface serial 0:23). Controller T1 0 and interface serial 0:23 are both assigned to the first PRI port. The second PRI port is assigned to controller T1 1 and interface serial 1:23, and so on. The second PRI port configuration is not shown in this example. This Cisco AS5300 is used as part of a stack group dial-in solution for an ISP.

```
as5300# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
as5300(config)# controller t1 0
as5300(config-controller)# framing esf
as5300(config-controller)# linecode b8zs
as5300(config-controller)# clock source line primary
as5300(config-controller)# pri-group timeslots 1-24
as5300(config-controller)# exit
as5300(config)# interface serial 0:23
```

```
as5300(config-if)# ip unnumbered Loopback 0
as5300(config-if)# ip accounting output-packets
as5300(config-if)# no ip mroute-cache
as5300(config-if)# encapsulation ppp
as5300(config-if)# isdn incoming-voice modem
as5300(config-if)# dialer-group 1
as5300(config-if)# no fair-queue
as5300(config-if)# compress stac
as5300(config-if)# no cdp enable
as5300(config-if)# ppp authentication chap
as5300(config-if)# ppp multilink
as5300(config-if)# netbios nbf
```

Line Types

This section describes the different line types used for dial access. It also describes the relationship between lines and interfaces.

Note Cisco devices have four types of lines: console, auxiliary, asynchronous, and virtual terminal lines. Different routers have different numbers of these line types. Refer to the hardware and software configuration guides that shipped with your device for exact configurations.

Table 14 shows the types of lines that can be configured.

Table 14 Available Line Types

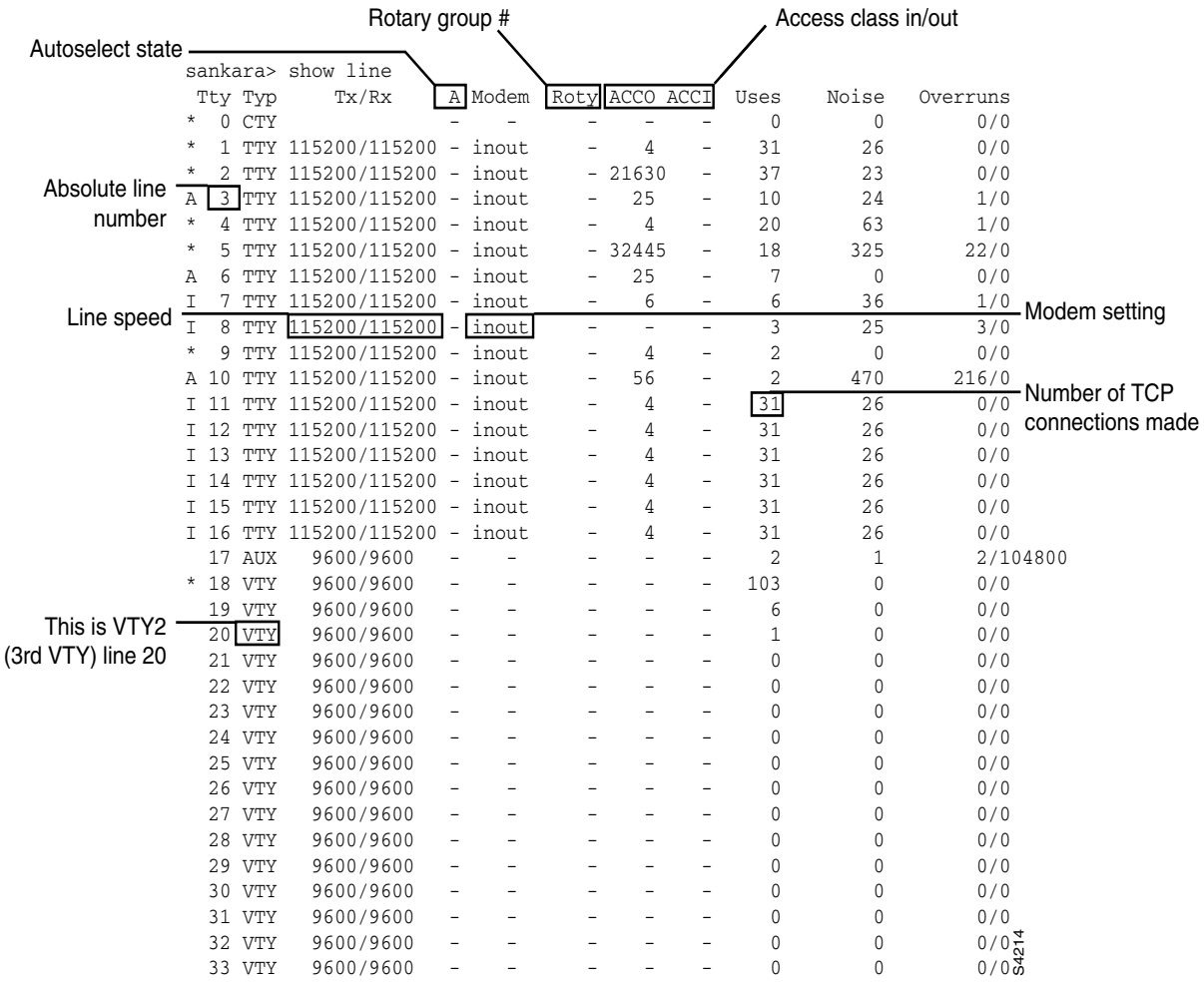
Line Type	Interface	Description	Numbering Rules
CON or CTY	Console	Typically used to log in to the router for configuration purposes.	Line 0.
AUX	Auxiliary	RS-232 DTE port used as a backup asynchronous port (TTY). Cannot be used as a Second console port.	Last TTY line number plus 1.
TTY	Asynchronous	Same as asynchronous interface. Used typically for remote-node dial-in sessions that use such protocols as SLIP, PPP, ARA, and XRemote.	The numbering widely varies between platforms. This number is equivalent to the maximum number of modems or asynchronous interfaces supported by your access server or router.[1]
VTY	Virtual asynchronous	Used for incoming Telnet, LAT, X.25 PAD, and protocol translation connections into synchronous ports (such as Ethernet and serial interfaces) on the router.	Last TTY line number plus 2 through the maximum number of VTY lines specified.[2]

1 Enter the **interface line tty ?** command to view the maximum number of TTY lines supported.

2 Increase the number of VTY lines on a router using the **line vty** command. Delete VTY lines with the **no line vty** *line-number* global configuration command. The **line vty** command accepts any line number larger than 5 up to the maximum number of lines supported by your router with its current configuration. Enter the **interface line vty ?** command to view the maximum number of VTY lines supported.

Use the **show line** command to see the status of each of the lines available on a router. See Figure 204.

Figure 204 Sample Show Line Output Showing CTY, TTY, AUX, and VTY Line Statistics




```
sankara> show line
 Tty Typ     Tx/Rx      A Modem  Roty ACCO ACCI  Uses   Noise   Overruns
*  0 CTY                -    -      -    -    -     0       0        0/0
*  1 TTY 115200/115200 - inout      -    4    -    31      26        0/0
*  2 TTY 115200/115200 - inout      - 21630   -    37      23        0/0
A  3 TTY 115200/115200 - inout      -   25    -    10      24        1/0
*  4 TTY 115200/115200 - inout      -    4    -    20      63        1/0
*  5 TTY 115200/115200 - inout      - 32445   -    18     325       22/0
A  6 TTY 115200/115200 - inout      -   25    -     7       0        0/0
I  7 TTY 115200/115200 - inout      -    6    -     6      36        1/0
I  8 TTY 115200/115200 - inout      -    -    -     3      25        3/0
*  9 TTY 115200/115200 - inout      -    4    -     2       0        0/0
A 10 TTY 115200/115200 - inout      -   56    -     2     470      216/0
I 11 TTY 115200/115200 - inout      -    4    -    31      26        0/0
I 12 TTY 115200/115200 - inout      -    4    -    31      26        0/0
I 13 TTY 115200/115200 - inout      -    4    -    31      26        0/0
I 14 TTY 115200/115200 - inout      -    4    -    31      26        0/0
I 15 TTY 115200/115200 - inout      -    4    -    31      26        0/0
I 16 TTY 115200/115200 - inout      -    4    -    31      26        0/0
  17 AUX   9600/9600    -    -      -    -    -     2       1        2/104800
* 18 VTY   9600/9600    -    -      -    -    -   103       0        0/0
  19 VTY   9600/9600    -    -      -    -    -     6       0        0/0
  20 VTY   9600/9600    -    -      -    -    -     1       0        0/0
  21 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  22 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  23 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  24 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  25 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  26 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  27 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  28 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  29 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  30 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  31 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  32 VTY   9600/9600    -    -      -    -    -     0       0        0/0
  33 VTY   9600/9600    -    -      -    -    -     0       0        0/0
```

Relationship between Lines and Interfaces

This section describes the relationship between lines and interfaces:

- Asynchronous Interfaces and TTY Lines
- Synchronous Interfaces and VTY Lines

Asynchronous Interfaces and TTY Lines

Asynchronous interfaces correspond to physical terminal (TTY) lines. Commands entered in asynchronous interface mode enable you to configure protocol-specific parameters for asynchronous interfaces; commands entered in line configuration mode let you configure the physical aspects of the line's port.

For example, to enable IP resources to dial in to a network through a Cisco 2500 series access server, configure the lines and asynchronous interfaces as follows:

Configure the physical aspect of a line that leads to a port. You might enter the following commands to configure lines 1 through 16 (asynchronous TTY lines on a Cisco 2511 access server):

```
line 1 16
 login local
 modem inout
 speed 115200
 flowcontrol hardware
 ! configures the line to autosense PPP; physical line attribute
 autoselect ppp
```

On asynchronous interface 1, you configure your protocol-specific commands. You might enter the following commands:

```
interface async 1
 encapsulation ppp
 async mode interactive
 async dynamic address
 async dynamic routing
 async default ip address 198.192.16.132
 ppp authentication chap
```

The remote node services SLIP, PPP, and XRemote are configured in asynchronous interface mode. AppleTalk Remote Access (ARA) is configured in line configuration mode on virtual terminal (VTY) lines or TTY lines.

Synchronous Interfaces and VTY Lines

Virtual terminal (VTY) lines provide access to the router through a synchronous interface. VTY lines do not correspond to synchronous interfaces in the same way that TTY lines correspond to asynchronous interfaces. This is because VTY lines are created dynamically on the router, whereas TTY lines are static physical ports. When a user connects to the router on a VTY line, that user is connecting into a *virtual* port on an interface. You can have multiple virtual ports for each synchronous interface.

For example, several Telnet connections can be made to an interface (such as an Ethernet or serial interface).

The number of VTY lines available on a router are defined using the **line vty** *number-of-lines* global configuration command.

Asynchronous Modem Lines

Asynchronous line configuration commands configure ports for the following options:

- Physical layer options (such as modem configuration)
- Security for EXEC mode
- ARA protocol configuration (PPP is configured in interface configuration mode)
- Autoselect to detect incoming protocols (ARA and PPP)

To enter line configuration mode, first connect to the console port of the access server and enter privileged EXEC mode. Then enter global configuration mode and finally enter line configuration mode for the asynchronous lines that you want to configure. The following example shows the process of entering line configuration mode for lines 1 through 16:

```
c2500> enable
c2500# configure terminal
c2500(config)# line 1 16
c2500(config-line)#
```

For additional information about configuring asynchronous modem lines, see the chapter "Configuring Modem Support and Asynchronous Devices."

Absolute versus Relative Line Numbers

When you enter line configuration mode, you can specify an *absolute line number* or a *relative line number*. For example, in Figure 204, absolute line number 20 is VTY2 (line 18 is VTY0). Referring to lines in a relative format is often easier than attempting to recall the absolute number of a line on a large system. Internally, the router uses absolute line numbers.

You can view all of the absolute and relative line numbers with the **show users all** EXEC command. In the following sample display, absolute line numbers are listed at the far left under the heading "Line." Relative line numbers are in the third column, after the line type. In this example, the second virtual terminal line, VTY 1, is absolute line number 3.

```
Line       User     Host(s)              Idle Location
0 con 0
1 aux 0
2 vty 0             incoming             0 SERVER.COMPANY.COM
3 vty 1
4 vty 2
5 vty 3
6 vty 4
```

Compare the line numbers in this sample display to the output from the **show line** command, as shown in Figure 204.