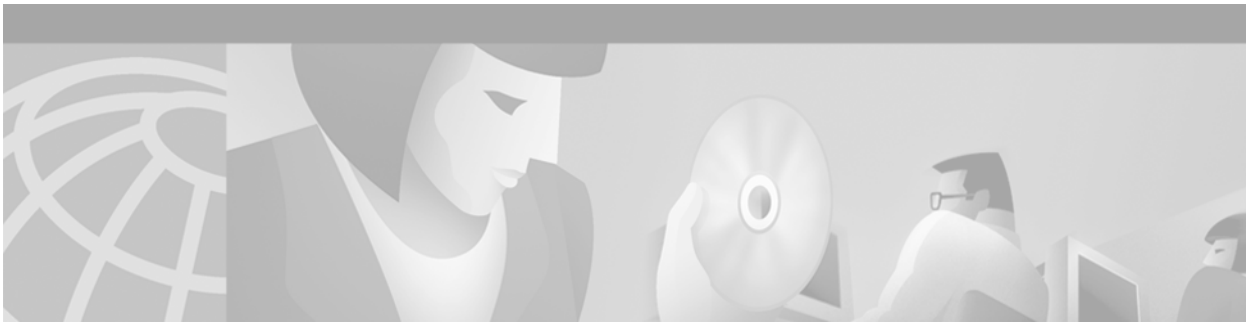


Preparing Asynchronous Interfaces

This chapter describes how to prepare asynchronous interfaces to allow connections such as analog modem calls to enter an access server. These tasks are presented in the following main sections:

- Modem and Asynchronous Interface Overview
- Asynchronous Interface Configuration Task List
- Asynchronous Interface Configuration Examples

See the section "Asynchronous Interfaces" in the chapter "Interfaces, Controllers, and Lines Used for Dial Access Overview" for more information about Cisco asynchronous serial interfaces.

For a complete description of the modem support commands in this chapter, see the *Cisco IOS Dial Services Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

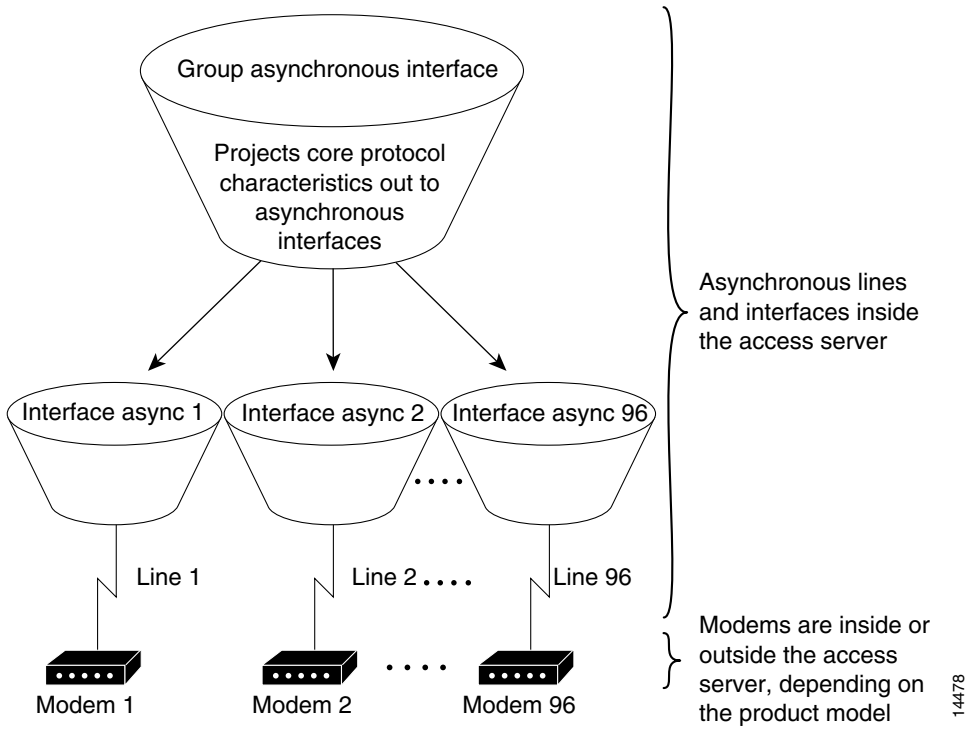
Modem and Asynchronous Interface Overview

Modems attach to asynchronous lines, which in turn attach to asynchronous interfaces. Depending on the type of access server you have, these components appear outside or inside the physical chassis. Figure 14 shows the logical relationship between modems, asynchronous lines, asynchronous interfaces, and group asynchronous interfaces. All these components work together to deliver packets as follows:

- Asynchronous calls come into the modems from the "plain old telephone service" (POTS) or Public Switched Telephone Network (PSTN).
- Modems pass packets up through asynchronous lines.
- Asynchronous interfaces clone their configuration information from group asynchronous interfaces.

Note The number of interfaces and modems varies among access server product models.

Figure 14 Modems, Lines, and Asynchronous Interfaces



Related Hardware Differences

Deciding which asynchronous features to use, to some degree, depends on your hardware configuration. All Cisco access servers must have their asynchronous interfaces and lines configured for network protocol support. Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode configure the physical and logical aspects for the same port.

Modems inside high-end access servers need localized modem country code. This code is projected from the Cisco IOS software to the onboard modems using the **modem country** {**mica** | **microcom_hdms**} *country* command. The following are high-end access servers: Cisco AS5800, Cisco AccessPath, Cisco AS5300, and the Cisco AS5200.

Modems externally attached to low-end access servers need to receive initialization strings from the **modem autoconfigure discovery** command. For troubleshooting tips, see the section "Attaching External Modems" in the chapter "Configuring Modems and Chat Scripts." The following are low-end access servers: Cisco AS2511-RJ, Cisco AS2509-RJ, Cisco 2509, Cisco 2511, and the Cisco 2512.

Figure 15 shows a Cisco AS2511-RJ access server. Figure 16 shows a Cisco AS5300 access server. Notice that modems are either inside or outside the chassis, depending on the product model.

Figure 15 Cisco AS2511-RJ Access Server

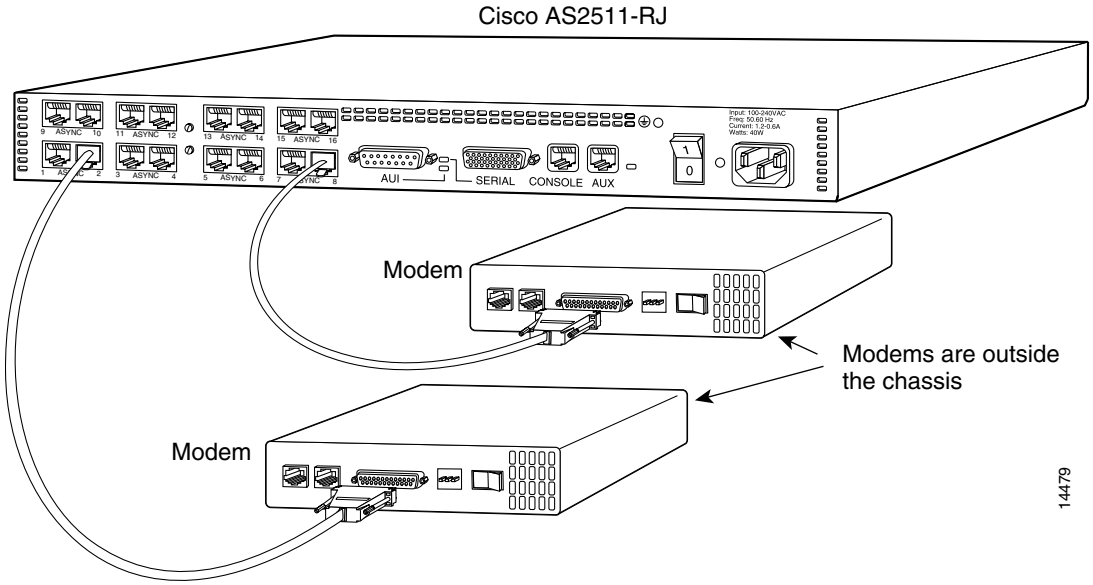
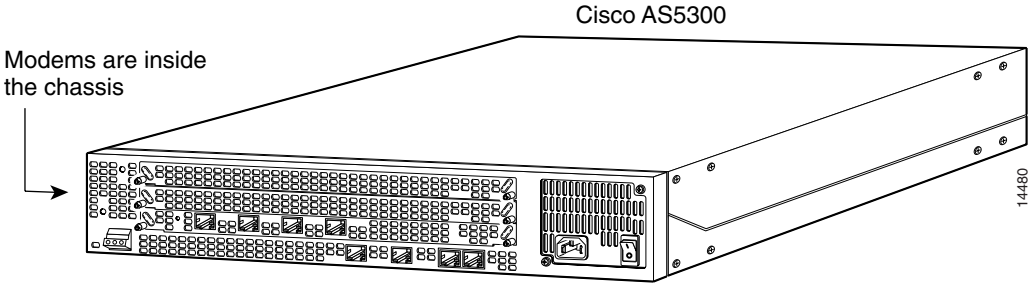


Figure 16 Cisco AS5300 Access Server



Asynchronous Modem Lines

Asynchronous line configuration commands configure ports for the following options:

- Physical layer options (such as modem configuration)
- Security for login
- AppleTalk Remote Access (ARA) protocol configuration
- 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.

Asynchronous Interfaces

Interfaces enable the Cisco IOS software to use routing functions. Specifically, you configure asynchronous interfaces to support PPP connections. Asynchronous interfaces are configured on access servers for the following functions:

- Network protocol support such as IP, Internet Protocol Exchange (IPX), or AppleTalk
- Encapsulation support such as PPP
- IP client addressing options (default or dynamic)
- IPX network addressing options
- PPP authentication

Group Asynchronous Interfaces

To configure multiple asynchronous interfaces at the same time (with the same parameters), you can assign each asynchronous interface to a group and then configure the group. Configurations throughout this guide configure group asynchronous interfaces, rather than each interface separately.

Note If you want to configure different attributes on different asynchronous interfaces, do not assign them to the group or assign different interfaces to different groups. After assigning asynchronous interfaces to a group, you cannot configure these interfaces separately. For example, on a Cisco AS5300 access server in a T1 configuration, you could assign asynchronous interfaces 1 to 48 as part of one group (such as group-async1) and asynchronous interfaces 49 to 96 as part of another group (group-async2). You can also use the **member** command to perform a similar grouping function.

Line and Modem Numbering Issues

The TTY line numbering scheme used by your access server or router is specific to your product and its hardware configuration. Refer to the product-specific documentation that came with your product for line numbering scheme information.

For example, the Cisco AS5200 access server has TTY lines that map directly to integrated modems, as shown in Table 4. Depending on the shelf, slot, and port physical architecture of the access server, the modem and TTY line number schemes will change.

As shown in Table 4, TTY lines 1 through 24 directly connect to modems 1/0 through 1/23, which are installed in the first chassis slot in this example. The TTY lines 25 through 48 directly connect to modems 2/0 through 2/23, which are installed in the second slot.

Table 4 TTY Lines Associated to Cisco AS5200 Modems

TTY Line	Slot/Modem Number	TTY Line	Slot/Modem Number
1	1/0	25	2/0
2	1/1	26	2/1
3	1/2	27	2/2
4	1/3	28	2/3
5	1/4	29	2/4
6	1/5	30	2/5
7	1/6	31	2/6
8	1/7	32	2/7
9	1/8	33	2/8
10	1/9	34	2/9
11	1/10	35	2/10
12	1/11	36	2/11
13	1/12	37	2/12
14	1/13	38	2/13
15	1/14	39	2/14
16	1/15	40	2/15
17	1/16	41	2/16
18	1/17	42	2/17
19	1/18	43	2/18
20	1/19	44	2/19
21	1/20	45	2/20
22	1/21	46	2/21
23	1/22	47	2/22
24	1/23	48	2/23

Asynchronous Interface Configuration Task List

To configure an asynchronous interface, perform the tasks in the following sections. The first task is required; the second task is optional.

- Configuring an Asynchronous Interface (Required)
- Creating a Group Asynchronous Interface (Optional)

Configuring an Asynchronous Interface

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

	Command	Purpose
Step 1	**interface async** *number*	Brings up a single asynchronous interface.
Step 2	**description** *description*	Provides a description for the interface.
Step 3	**ip address** *address mask*	Specifies an IP address.
Step 4	**encapsulation ppp**	Enables PPP to run on the asynchronous interfaces in the group.
Step 5	**async default routing**	Enables the router to pass routing updates to other routers over the AUX port configured as an asynchronous interface.
Step 6	**async mode dedicated**	Places a line into dedicated asynchronous mode using Serial Line Internet Protocol (SLIP) or PPP encapsulation.
Step 7	**dialer in-band**	Specifies that dial-on-demand routing (DDR) is to be supported.
Step 8	**dialer map** *protocol next-hop-address*	Configures a serial interface to call one or multiple sites or to receive calls from multiple sites.
Step 9	**dialer-group**	Controls access by configuring an interface to belong to a specific dialing group.
Step 10	**ppp authentication chap pap** *list-name*	Enables Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[1]
Step 11	**exit**	Return to global configuration mode.

1. To create a string used to name the following list of authentication methods tried when a user logs in, see the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, Kerberos, and so on.

See the sections “Configuring an Asynchronous Interface” and “Asynchronous Interface As the Only Network Interface” later in this chapter for examples of how to configure an asynchronous interface.

Creating a Group Asynchronous Interface

Create a group asynchronous interface to project a set of core protocol characteristics to a range of asynchronous interfaces. Configuring the asynchronous interfaces as a group saves you time. Analog modem calls cannot enter the access server without this configuration.

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

	Command	Purpose
Step 1	**interface group-async** *number*	Brings up a single group asynchronous interface.
Step 2	**ip unnumbered loopback** *number*	Configures the asynchronous interfaces as unnumbered, and assigns the IP address of the loopback interface to them, to conserve IP addresses.[1]
Step 3	**encapsulation ppp**	Enables PPP to run on the asynchronous interfaces in the group.
Step 4	**async mode interactive**	Configures interactive mode on the asynchronous interface.
Step 5	**ppp authentication chap pap** *list-name*	Enables CHAP and PAP authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[2]
Step 6	**peer default ip address pool** *poolname*	Assigns dial-in clients IP addresses from an address pool.[3]
Step 7	**no cdp enable**	Disables the Cisco Discovery Protocol (CDP) on the interface.
Step 8	**group-range** *low-end-of-range high-end-of-range*	Specifies the range of asynchronous interfaces to include in the group, which is usually equal to the number of modems you have in the access server.
Step 9	**exit**	Returns to global configuration mode.

1. You can also specify the Ethernet interface to conserver address space. In this case, enter the **ip unnumbered ethernet 0** command.
2. To create a string used to name the following list of authentication methods tried when a user logs in, see the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, Kerberos, and so on.
3. To create an IP address pool, see the **ip local pool** global configuration command.

See the section “Creating a Group Asynchronous Interface” later in this chapter for an example of how to configure a group interface.

Verifying the Group Interface Configuration

To verify the group interface configuration and check if one of the asynchronous interfaces is up, use the **show interface async** command:

```
NAS# show interface async 1
Async1 is up, line protocol is up
modem(slot/port)=1/0, csm_state(0x00000204)=CSM_IC4_CONNECTED, bchan_num=18
modem_status(0x0002): VDEV_STATUS_ACTIVE_CALL.

  Hardware is Async Serial
  Interface is unnumbered.  Using address of FastEthernet0 (10.1.1.10)
  MTU 1500 bytes, BW 115 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/5, 0 drops; input queue 1/5, 0 drops
  5 minute input rate 37000 bits/sec, 87 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     31063 packets input, 1459806 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     33 packets output, 1998 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

If you are having trouble, enter one of the following **debug** commands and then send a call into the access server. Interpret the output and make configuration changes accordingly.

- **undebug all**
- **debug ppp negotiation**
- **debug ppp authentication**
- **debug modem**
- **debug ip peer**

```
nas-1# undebug all
All possible debugging has been turned off
nas-1# debug ppp negotiation
PPP protocol negotiation debugging is on
nas-1# debug ppp authentication
PPP authentication debugging is on
nas-1# debug modem
Modem control/process activation debugging is on

nas-1# debug ip peer
IP peer address activity debugging is on
nas-1# show debug
General OS:
  Modem control/process activation debugging is on
Generic IP:
  IP peer address activity debugging is on
PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
nas-1#
```

```
*Mar  1 21:34:56.958: TTY4: DSR came up
*Mar  1 21:34:56.962: tty4: Modem: IDLE->READY
*Mar  1 21:34:56.970: TTY4: EXEC creation
*Mar  1 21:34:56.978: TTY4: set timer type 10, 30 seconds
*Mar  1 21:34:59.722: TTY4: Autoselect(2) sample 7E
*Mar  1 21:34:59.726: TTY4: Autoselect(2) sample 7EFF
*Mar  1 21:34:59.730: TTY4: Autoselect(2) sample 7EFF7D
*Mar  1 21:34:59.730: TTY4: Autoselect(2) sample 7EFF7D23
*Mar  1 21:34:59.734: TTY4 Autoselect cmd: ppp negotiate
*Mar  1 21:34:59.746: TTY4: EXEC creation
*Mar  1 21:34:59.746: TTY4: create timer type 1, 600 seconds
*Mar  1 21:34:59.786: ip_get_pool: As4: using pool default
*Mar  1 21:34:59.790: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:34:59.794: TTY4: destroy timer type 1 (OK)
*Mar  1 21:34:59.794: TTY4: destroy timer type 0
*Mar  1 21:35:01.798: %LINK-3-UPDOWN: Interface Async4, changed state to up
*Mar  1 21:35:01.834: As4 PPP: Treating connection as a dedicated line
*Mar  1 21:35:01.838: As4 PPP: Phase is ESTABLISHING, Active Open
*Mar  1 21:35:01.842: As4 LCP: O CONFREQ [Closed] id 1 len 25
*Mar  1 21:35:01.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:01.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:01.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:01.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:01.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.718: As4 LCP: I CONFREQ [REQsent] id 3 len 23
*Mar  1 21:35:02.722: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.726: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.726: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.730: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.730: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.738: As4 LCP: O CONFREJ [REQsent] id 3 len 7
*Mar  1 21:35:02.738: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.850: As4 LCP: I CONFREQ [REQsent] id 4 len 20
*Mar  1 21:35:02.854: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.854: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.858: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.862: As4 LCP: O CONFACK [REQsent] id 4 len 20
*Mar  1 21:35:02.866: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.870: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.870: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.874: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.842: As4 LCP: TIMEout: State ACKsent
*Mar  1 21:35:03.842: As4 LCP: O CONFREQ [ACKsent] id 2 len 25
*Mar  1 21:35:03.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.962: As4 LCP: I CONFACK [ACKsent] id 2 len 25
*Mar  1 21:35:03.966: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.966: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.970: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.974: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.974: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.978: As4 LCP: State is Open
*Mar  1 21:35:03.978: As4 PPP: Phase is AUTHENTICATING, by this end
*Mar  1 21:35:03.982: As4 CHAP: O CHALLENGE id 1 len 26 from "nas-1"
*Mar  1 21:35:04.162: As4 CHAP: I RESPONSE id 1 len 26 from "krist"
*Mar  1 21:35:04.170: As4 AUTH: Started process 0 pid 47
*Mar  1 21:35:04.182: As4 CHAP: O SUCCESS id 1 len 4
*Mar  1 21:35:04.186: As4 PPP: Phase is UP
*Mar  1 21:35:04.190: As4 IPCP: O CONFREQ [Not negotiated] id 1 len 10
*Mar  1 21:35:04.194: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
```

```
*Mar  1 21:35:04.202: As4 CDPCP: O CONFREQ [Closed] id 1 len 4
*Mar  1 21:35:04.282: As4 IPCP: I CONFREQ [REQsent] id 1 len 40
*Mar  1 21:35:04.282: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.286: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:04.290: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:04.294: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:04.298: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:04.302: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:04.306: As4 IPCP: O CONFREJ [REQsent] id 1 len 10
*Mar  1 21:35:04.310: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.314: As4 CCP: I CONFREQ [Not negotiated] id 1 len 15
*Mar  1 21:35:04.318: As4 CCP:    MS-PPC supported bits 0x00000001 (0x1206000000
01)
*Mar  1 21:35:04.318: As4 CCP:    Stacker history 1 check mode EXTENDED (0x11050
00104)
*Mar  1 21:35:04.322: As4 LCP: O PROTREJ [Open] id 3 len 21 protocol CCP
*Mar  1 21:35:04.326: As4 LCP:  (0x80FD0101000F12060000000111050001)
*Mar  1 21:35:04.330: As4 LCP:  (0x04)
*Mar  1 21:35:04.334: As4 IPCP: I CONFACK [REQsent] id 1 len 10
*Mar  1 21:35:04.338: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:04.342: As4 LCP: I PROTREJ [Open] id 5 len 10 protocol CDPCP (0x82
0701010004)
*Mar  1 21:35:04.342: As4 CDPCP: State is Closed
*Mar  1 21:35:05.186: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async4, ch
anged state to up
*Mar  1 21:35:05.190: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:35:05.190: As4 PPP: Trying to negotiate NCP for Link cdp
*Mar  1 21:35:05.194: As4 CDPCP: State is Closed
*Mar  1 21:35:05.198: As4 CDPCP: TIMEout: State Closed
*Mar  1 21:35:05.202: As4 CDPCP: State is Listen
*Mar  1 21:35:06.202: As4 IPCP: TIMEout: State ACKrcvd
*Mar  1 21:35:06.206: As4 IPCP: O CONFREQ [ACKrcvd] id 2 len 10
*Mar  1 21:35:06.206: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:06.314: As4 IPCP: I CONFACK [REQsent] id 2 len 10
*Mar  1 21:35:06.318: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:07.274: As4 IPCP: I CONFREQ [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.278: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:07.282: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:07.286: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:07.286: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:07.290: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:07.294: As4 IPCP: O CONFNAK [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.298: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.302: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.306: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.310: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.314: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.426: As4 IPCP: I CONFREQ [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.430: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.434: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.438: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.442: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.446: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.446: ip_get_pool: As4: validate address = 172.20.1.101
*Mar  1 21:35:07.450: ip_get_pool: As4: using pool default
*Mar  1 21:35:07.450: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:35:07.454: set_ip_peer_addr: As4: address = 172.20.1.101 (3) is redun
dant
*Mar  1 21:35:07.458: As4 IPCP: O CONFACK [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.462: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.466: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.470: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
```

```
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.478: As4 IPCP: State is Open
*Mar  1 21:35:07.490: As4 IPCP: Install route to 172.20.1.101
*Mar  1 21:35:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:36:12.614: TTY0: timer type 1 expired
*Mar  1 21:36:12.614: TTY0:  Exec timer (continued)
*Mar  1 21:36:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:37:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:38:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:39:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:40:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:41:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:42:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:43:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
```

Asynchronous Interface Configuration Examples

This section provides the following asynchronous interface configuration examples:

- Interface and Line Configuration
- Dedicated Asynchronous Interface Configuration
- Access Restriction on the Asynchronous Interface
- Group and Member Asynchronous Interfaces
- Asynchronous Interface Address Pool Examples
- IP and SLIP Using an Asynchronous Interface
- AppleTalk and PPP Asynchronous Interface Configuration
- IP and PPP Asynchronous Interface Configuration
- IPX and PPP Using a Loopback Interface
- IPX and PPP Using Dedicated IPX Network Numbers for Each Interface
- IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines
- Asynchronous Routing and Dynamic Addressing Configuration
- TCP Header Compression Configuration
- Network Address Conservation Using the IP Unnumbered Feature
- Asynchronous Interface As the Only Network Interface
- Routing on a Dedicated Dial-In Router
- IGRP Configuration

Interface and Line Configuration

The following is an example of one asynchronous interface configuration on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
interface async 1
 description ASYNC LINE 5293731 TO HIGHWAY
 encapsulation ppp
 async default routing
 async mode dedicated
 dialer in-band
 dialer map ip 192.168.10.2 name Router2 broadcast
 dialer-group 1
 ppp authentication chap
```

The following configuration shows interface and line configuration. The interface is configured with access lists, passive header compression, and a default address. The line is configured for TACACS authentication.

```
interface async 1
 ip access-group 1 in
 ip access-group 1 out
 ip tcp header-compression passive
 async default ip address 172.31.176.201

line 1
 login tacacs
 location 457-5xxx
 exec-timeout 20 0
 password XXXXXXXX
 session-timeout 20
 stopbits 1
```

The following example configures a Cisco AS5800 access server, which is used as a high-density dial-in solution:

```
configure terminal
line 1/2/00 1/9/71
 session-timeout 30
 exec-timeout 30 0
 absolute-timeout 240
 autoselect during-login
 autoselect ppp
 modem InOut
 transport preferred none
 transport input all
```

The following example configures one asynchronous line on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
configure terminal
line 1
 modem InOut
 speed 115200
 transport input all
 flowcontrol hardware
```

Dedicated Asynchronous Interface Configuration

The following example shows how to assign an IP address to an asynchronous interface and place the line in dedicated network mode. Setting the stop bit to 1 is a performance enhancement.

```
line 20
 location Department PC Lab
 stopbits 1
 speed 19200
!
interface async 20
 async default ip address 182.32.7.51
 async mode dedicated
```

Access Restriction on the Asynchronous Interface

The following example shows how to allow most terminal users access to anything on the local network, but restrict access to certain servers designated as asynchronous servers:

```
! access list for normal connections
access-list 1 permit 131.108.0.0 0.0.255.255
!
access-list 2 permit 131.108.42.55
access-list 2 permit 131.108.111.1
access-list 2 permit 131.108.55.99
!
line 1
 speed 19200
 flow hardware
 modem inout
interface async 1
 async mode interactive
 async dynamic address
 ip access-group 1 out
 ip access-group 2 in
```

Group and Member Asynchronous Interfaces

The following example shows how to create an asynchronous group interface 0 with group interface members 2 through 7, beginning in global configuration mode:

```
interface group-async 0
 group-range 2 7
```

The following example shows how you need to configure asynchronous interfaces 1, 2, and 3 separately if you do not have a group interface configured:

```
interface Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.1
 async mode interactive
 async dynamic routing
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.2
 async mode interactive
 async dynamic routing
!
interface Async3
 ip unnumbered Ethernet0
!
 encapsulation ppp
 async default ip address 172.30.1.3
 async mode interactive
 async dynamic routing
```

The following example configures the same interfaces, but from a single group asynchronous interface:

```
!
interface Group-Async 0
 ip unnumbered Ethernet0
 encapsulation ppp
 async mode interactive
 async dynamic routing
 group-range 1 3
 member 1 async default ip address 172.30.1.1
 member 2 async default ip address 172.30.1.2
 member 3 async default ip address 172.30.1.3
```

To configure a group asynchronous interface, specify the group async number (an arbitrary number) and the group range (beginning and ending asynchronous interface number). The following example shows the process of creating and configuring a group asynchronous interface for asynchronous interfaces 1 through 96 on a Cisco AS5300 access server, which is loaded with ninety-six 56K MICA technologies modems:

```
interface group-async 1
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 ppp authentication chap pap
 peer default ip address pool default
 group-range 1 96
```

The following example configures a Cisco AS5800 access server that is used as a high-density dial-in solution:

```
interface group-async 0
 ip unnumbered FastEthernet0/2/0
 encapsulation ppp
 async mode interactive
 peer default ip address pool default
 no cdp enable
 ppp authentication chap
 hold-queue 10 in
 group-range 1/2/00 1/9/71
```

Asynchronous Interface Address Pool Examples

The following sections provide examples of the use of Dynamic Host Configuration Protocol (DHCP) and local pooling mechanisms:

- DHCP Pooling
- Local Pooling
- Configuring Specific IP Addresses for an Interface

DHCP Pooling

The following global configuration example enables DHCP proxy-client status on all asynchronous interfaces on the access server:

```
ip address-pool dhcp-proxy-client
```

The following global configuration example shows how to specify which DHCP servers are used on your network. You can specify up to four servers using IP addresses or names. If you do not specify servers, the default is to use the IP limited broadcast address of 255.255.255.255 for transactions with any and all discovered DHCP servers.

```
ip dhcp-server jones smith wesson
```

The following interface configuration example illustrates how to disable DHCP proxy-client functionality on asynchronous interface 1:

```
async interface
interface 1
 no peer default ip address
```

Local Pooling

The following example shows how to select the IP pooling mechanism and how to create a pool of local IP addresses that are used when a client dials in on an asynchronous line. The default address pool comprises IP addresses 172.30.0.1 through 172.30.0.28.

```
! this command tells the access server to use a local pool
ip address-pool local
! this command defines the ip address pool.
! The address pool is named group1 and comprised of addresses
! 10.1.2.1through 10.1.2.5 inclusive
ip local-pool group1 10.1.2.1 10.1.2.5
```

Configuring Specific IP Addresses for an Interface

The following example shows how to configure the access server so that it will use the default address pool on all interfaces except interface 7, on which it will use an address pool called lass:

```
ip address-pool local
ip local-pool lass 172.30.0.1
 async interface
 interface 7
 peer default ip address lass
```

IP and SLIP Using an Asynchronous Interface

The following example configures IP and SLIP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running SLIP has been specified.

IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation slip
 async mode interactive
 async default ip address 172.18.1.128
```

AppleTalk and PPP Asynchronous Interface Configuration

The following example configures asynchronous interface 4 on the router so that users can access AppleTalk zones by dialing in to the router via PPP to this interface. Users accessing the network can run AppleTalk and IP natively on a remote Macintosh, access any available AppleTalk zones from Chooser, use networked peripherals, and share files with other Macintosh users. Routing is not supported on the asynchronous interface 6.

```
interface async 6
 encapsulation ppp
 appletalk virtual-net 12345 saivite
 appletalk client-mode
```

IP and PPP Asynchronous Interface Configuration

The following example configures IP and PPP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running PPP has been specified.

IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 peer default ip address 172.18.1.128
```

IPX and PPP Using a Loopback Interface

The following example shows how to configure IPX to run over PPP on an asynchronous interface. The asynchronous interface is associated with a loopback interface configured to run IPX. This example enables a nonrouting IPX client to connect to the router.

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

In this example, IPX client connections are permitted to asynchronous interface 3, which is associated with loopback interface 0. Loopback interface 0 is configured to run IPX. Routing updates have been filtered on asynchronous interface 3. Routing updates take up much of the bandwidth, and asynchronous interfaces have low bandwidth.

IPX and PPP Using Dedicated IPX Network Numbers for Each Interface

The following example shows how to configure IPX to run over PPP on an asynchronous interface. A dedicated IPX network number has been specified for each interface, which can require a substantial number of network numbers for a large number of interfaces. This example permits an IPX client with routing enabled to connect with the router.

```
ipx routing 0000.0c07.b509
interface async 6
 ip unnumbered ethernet0
 encapsulation ppp
 async mode interactive
 ipx network AC150E00
 ipx sap-interval 0
```

In this example, IPX client connections are permitted to asynchronous interface 6, which has a unique IPX network number. Routing updates have been filtered on asynchronous interface 6. Routing updates take up much of the bandwidth, and asynchronous interfaces have low bandwidth.

IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines

The following example shows how to enable IPX and PPP on vty lines. First, enable PPP to run on virtual terminal lines, then associate the virtual terminal line with a loopback interface configured to run IPX:

```
ipx routing 0000.0c07.b509
interface loopback0
 no ip address
 ipx network 544
vty-async ipx ppp-client loopback0
```

This example enables a nonrouting IPX client to connect to the router by permitting IPX client connections to vty lines, which have been associated with loopback interface 0. Loopback interface 0 is configured with an IPX network number that is used by the virtual terminal lines.

Asynchronous Routing and Dynamic Addressing Configuration

The following example shows a simple configuration that allows routing and dynamic addressing. With this configuration, if the user specifies **/routing** in the EXEC **slip** or **ppp** command, routing protocols will be sent and received.

```
interface async 6
 async dynamic routing
 async dynamic address
```

TCP Header Compression Configuration

The following example configures async interface 7 with a default IP address, allowing header compression if it is specified in the **slip** or **ppp** connection command entered by the user or if the connecting system sends compressed packets.

```
interface async 7
 ip address 172.31.79.1
 async default ip address 172.31.79.2
 ip tcp header-compression passive
```

Network Address Conservation Using the IP Unnumbered Feature

The following example shows how to configure your router for routing using unnumbered interfaces. The source (local) address is shared between Ethernet 0 and async 6 (172.18.1.1). The default remote address is 172.18.1.2.

```
interface ethernet 0
 ip address 172.18.1.1 255.255.255.0
!
interface async 6
 ip unnumbered ethernet 0
 async dynamic routing
! default address is on the local subnet
 async dynamic address
 async default ip address 172.18.1.2
 ip tcp header-compression passive
```

The following example shows how the IP unnumbered configuration works. Although the user assigned an address, the system response shows the interface as unnumbered, and the address entered by the user will be used only in response to BOOTP requests.

```
router> slip /compressed 10.11.11.254
Password:
Entering async mode.
Interface IP address is unnumbered, MTU is 1500 bytes.
Header compression is On.
```

Asynchronous Interface As the Only Network Interface

The following example shows how one of the asynchronous lines can be used as the only network interface. The router is used primarily as a terminal server, but is at a remote location and dials in to the central site for its only network connection.

```
ip default-gateway 10.11.12.2
interface ethernet 0
 shutdown
interface async 1
 async dynamic routing
 ip tcp header-compression on
 async default ip address 10.11.16.12
 async mode dedicated
 ip address 10.11.12.32 255.255.255.0
```

Routing on a Dedicated Dial-In Router

The following example shows how a router is set up as a dedicated dial-in router. Interfaces are configured as IP unnumbered to conserve network resources, primarily IP addresses.

```
ip routing
interface ether 0
 ip address 10.129.128.2 255.255.255.0
!
interface async 1
 ip unnumbered ethernet 0
 async dynamic routing
! The addresses assigned with SLIP or PPP EXEC commands are not used except
! to reply to BOOTP requests.
! Normally, the routers dialing in will have their own address and not use BOOTP at all.
 async default ip address 10.11.11.254
!
interface async 2
 ip unnumbered ethernet 0
 async default ip address 10.11.12.16
 ip tcp header-compression passive
 async mode dedicated
!
! Run RIP on the asynchronous lines, because few implementations of SLIP
! understand IGRP. Run IGRP on the Rthernet (and in the local network).
!
router igrp 110
 network 10.11.12.0
! Send routes from the asynchronous lines on the production network.
 redistribute RIP
```

```
! Don't send IGRP updates on the async interfaces.
 passive-interface async 1
!
 router RIP
 network 10.11.12.0
 redistribute igrp
 passive-interface ethernet 0
! Consider filtering everything except a default route from the routing
! updates sent on the (slow) asynchronous lines.
 distribute-list 1 out
 ip unnumbered async 2
 async dynamic routing
```

IGRP Configuration

In the following example, only the Interior Gateway Routing Protocol (IGRP) TCP/IP routing protocol is running; it is assumed that the systems that are dialing in to use routing will either support IGRP or have some other method (for example, a static default route) of determining that the router is the best place to send most of its packets:

```
router igrp 111
 network 10.11.12.0
interface ethernet 0
 ip address 10.11.12.92 255.255.255.0
!
interface async 1
 async default ip address 10.11.12.96
 async dynamic routing
 ip tcp header-compression passive
 ip unnumbered ethernet 0

line 1
 modem ri-is-cd
```