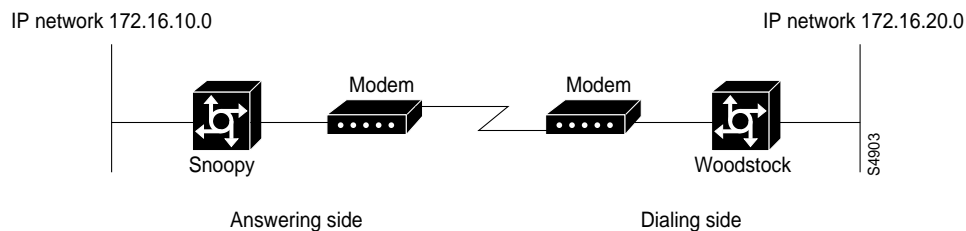


Routing across Modem Lines

Previous chapters in this guide have focused on configuring an access server to allow remote node computers to dial in to a network. This chapter describes how to configure two access servers so that one places an outgoing call and a second access server accepts. The access server initiating the call establishes a dial-on-demand routing (DDR) connection to the answering access server when packets that are considered “interesting” (IP unicast packets) pass through the interface configured to initiate a call.

Figure 6-1 shows a simple DDR scenario between two access servers. In this example, an IP host on network 172.16.20.0 opens a connection session with a host on IP network 172.16.10.0. The two access servers exchange routing information using the RIP routing protocol (although RIP broadcasts cannot initiate a call or keep the line active). This figure is referred to throughout this chapter and the sample configurations are based on this figure.

Figure 6-1 Asynchronous Dial-on-Demand Routing Network Design



In the preceding example, the answering access server is Snoopy on IP network 172.16.10.0, and the dialing access server is Woodstock on IP network 172.16.20.0. You must configure the answering access server first, then configure the dialing access server.

Configuring the Answering Access Server

In this configuration, the answering access server has the name Snoopy. This name is passed by this access server in a PPP authentication process. Each access server has the name of the other access server defined in its username database (with the **username** command). That is, the dialing access server—Woodstock—must have a username Snoopy defined, and the answering access server—Snoopy—must have the username Woodstock defined. Refer to the section “Configuring Security for the Access Server Answering the DDR Call.”

Note Names are case sensitive, so be sure that both the dialing and answering access servers use the same capitalization and spelling.

Perform the steps in the following procedures to configure the answering access server (Snoopy). The configuration is broken into procedural components (routing in global configuration mode, the Ethernet interface, asynchronous interfaces, security, and so on).

Defining Modem Parameters

Perform the following steps to configure modem support for an access server answering DDR calls (Snoopy):

Step 1 Configure the line speed. In the following example, line speed is set to 115200 bps. If you are configuring dialin on an AUX port, the maximum speed is 38400 bps.

```
Snoopy(config)# line tty 1
Snoopy(config-line)# speed 115200
```

Step 2 Configure flow control on the line accepting the incoming DDR call.

```
Snoopy(config-line)# flowcontrol hardware
```

Step 3 Because the answering access server is taking incoming calls on line 1 only, configure the modem to accept incoming calls on that line.

```
Snoopy(config-line)# modem dialin
```

Note You cannot establish a reverse Telnet session to the modem attached to line 1 if the **modem dialin** command is used. To use reverse Telnet, you must use the **modem inout** command. After a reverse Telnet session is completed, you can reissue the **modem dialin** command.

Configuring Routing and a Routing Protocol

Perform the following steps to configure RIP routing on the access server answering DDR calls (Snoopy).

Step 1 Configure RIP routing globally on the access server answering DDR calls (Snoopy):

```
Snoopy(config)# router rip
Snoopy(config-router)#
```

Step 2 Associate a network to the RIP routing process:

```
Snoopy(config-router)# network 172.16.0.0
Snoopy(config-router)# exit
Snoopy(config)#
```

Step 3 Create a static default route. A static default route is required because routes that are resolved dynamically are lost when the DDR link is down. If the access server receives a packet that is destined to a network not listed in its routing table, the access server forwards this packet to the access server on the other side of the dialup link (in this case to 172.16.20.1), which is the address of the opposite access server (Woodstock).

```
Snoopy(config)# ip route 0.0.0.0 0.0.0.0 172.16.20.1
```

- Step 4** Configure a second static route, because the asynchronous interface is unnumbered (refer to the section “Configuring the Asynchronous Interface Answering the DDR Call”). A second static route is needed to tell the local access server (Snoopy) which interface to use to get to the device at address 172.16.20.1. A mask of 255.255.255.255 is used to specify that this route is a host address.

```
Snoopy(config)# ip route 172.16.20.1 255.255.255.255 async1
```

Configuring Ethernet Interface 0

Perform the following task to configure Ethernet interface 0 on the access server answering incoming DDR calls (Snoopy):

Assign an IP address to Ethernet interface 0:

```
Snoopy(config-router)# interface Ethernet0
Snoopy(config-if)# ip address 172.16.10.1 255.255.255.0
```

Configuring the Asynchronous Interface Answering the DDR Call

Perform the following steps to configure the asynchronous interface answering DDR calls (Snoopy):

- Step 1** Configure the asynchronous interface through which you need to accept a call as IP unnumbered to conserve IP addresses and assign the IP address for Ethernet interface 0 to it.

```
Snoopy(config-if)# interface Async1
Snoopy(config-if)# ip unnumbered Ethernet0
```

- Step 2** Encapsulate PPP on the interface.

```
Snoopy(config-if)# encapsulation ppp
```

- Step 3** Specify asynchronous dynamic routing on the interface. The **async dynamic routing** command allows routing protocols to be run over the asynchronous interface to resolve IP routes dynamically. If the command is omitted, static routes can still be used.

```
Snoopy(config-if)# async dynamic routing
```

- Step 4** Specify the IP address of the opposite access server’s (Woodstock’s) Ethernet 0 interface as the default IP address:

```
Snoopy(config-if)# peer default ip address 172.16.20.1
```

- Step 5** Configure the asynchronous interface as dedicated to PPP mode, which means that the access server automatically uses a PPP session for this interface, and that the user will not see an EXEC prompt. The **async mode dedicated** command enables the configured session type to start automatically when the DDR link comes up.

```
Snoopy(config-if)# async mode dedicated
```

- Step 6** (Optional) Configure DDR support on the asynchronous interface using the **dialer in-band** command.

```
Snoopy(config-if)# dialer in-band
```

- Step 7** Set the number of seconds the connection remains open if no interesting traffic is being routed across this link. The timer is reset each time an interesting packet is forwarded across the DDR connection. You need to set the idle-timeout to the same value on both access servers. In this example, the line is closed after 5 consecutive minutes without interesting traffic.

```
Snoopy(config-if)# dialer idle-timeout 300
```

- Step 8** Specify that the name Woodstock be used to authenticate the dialin user. If authentication is successful, the IP address of the dialing access server's Ethernet interface (in this case, 172.16.20.1) is mapped to the remote user. Also, enable broadcast packets to be forwarded to this address (such as RIP or IGRP updates for IP).

Note There is no telephone number specified in the **dialer map** command, because Snoopy is not calling out. Snoopy is only accepting incoming DDR calls.

```
Snoopy(config-if)# dialer map ip 172.16.20.1 name Woodstock broadcast
```

- Step 9** Associate this interface with the dialer list 1 definition by using the **dialer-group 1** command. The interface now considers anything defined in dialer list 1 as interesting traffic.

```
Snoopy(config-if)# dialer-group 1  
Snoopy(config-if)# exit
```

Configuring Security for the Access Server Answering the DDR Call

To configure security on an access server answering DDR calls (Snoopy), perform these steps:

- Step 1** Specify the name of the dialing access server (Woodstock) in Snoopy's username database. This username is referenced in the **dialer map** command for authentication purposes. The username is case sensitive and must match the opposite access server's host name exactly. The password (peanuts) is used as the PPP authentication password for the user Woodstock. It is also case sensitive:

```
Snoopy(config)# username Woodstock password peanuts
```

Note If you enter the password **peanuts**, exit to privileged EXEC mode, and issue the **show running-config** command, the output of this command displays an encrypted password, similar to the following: `username Woodstock password 7 0215055500070C294D`. When you enter or make changes to the username command, always enter the password in its unencrypted form. Do not enter the encryption type (7). It is set automatically.

- Step 2** Create a PPP authentication list and a login authentication list:

```
Snoopy(config)# aaa authentication ppp default local  
Snoopy(config)# aaa authentication login default local
```

- Step 3** Apply the PPP authentication list to the asynchronous interface answering DDR calls and specify CHAP authentication (rather than PAP):

```
Snoopy(config)# interface async 1  
Snoopy(config-if)# ppp authentication chap default
```

- Step 4** Require login authentication on VTY lines 0 through 4. The login authentication default command uses the **aaa authentication default local** authentication list. The local keyword means that the local username database will be used for security. On this access server, only five VTY lines have been defined.

```
Snoopy(config-if)# line vty 0 4
Snoopy(config-line)# login authentication default
```

- Step 5** Create access list filters. In this example, the packets that the access list permits are referenced by the **dialer-list** command (in Step 6 of this procedure) to determine interesting packets.

```
Snoopy(config-line)# exit
Snoopy(config)# access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255
0.0.0.0
Snoopy(config)# access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0
255.255.255.255
```

In the preceding **access-list** command, the number 100 is the list identifier. All access-list commands with the same identifier define a single filter. Ordering of the access-list commands is very important. Statements in an access list are parsed one by one until a match is found. After a match is found, any access list definitions that follow are ignored. Although it is not displayed, an implicit “deny all” statement is always appended to the end of an access list. Therefore, if a packet reaches the end of an access list without matching a permit statement, the packet is denied automatically.

The line **access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0** specifies that all broadcast packets are uninteresting. Specifically, RIP updates cannot initiate a call, nor can they reset the dialer idle-timeout counter in this example.

The line **access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255** specifies that all other IP packets are interesting.

- Step 6** Reference the filter defined by access list 100. Packets permitted by access list 100 are considered interesting packets for a DDR interface belonging to dialer group 1.

```
Snoopy(config)# dialer-list 1 list 100
```

You have configured the answering access server (Snoopy). At this point, you can configure the dialing access server (Woodstock).

Configuring the Dialing Access Server

In this configuration, the dialing access server has the name Woodstock. This name is passed by this access server during the PPP authentication process, in the same way that the answering access server’s name is authenticated. (Refer to the section “Configuring Security for the Dialing Access Server.”)

Note Names are case sensitive, so be sure that both the dialing and answering access servers use the same capitalization and spelling.

Perform the steps in the following procedures to configure the access server that initiates the call (Woodstock). The configuration is broken into components (routing in global configuration mode, the Ethernet interface, asynchronous interfaces, security, and so on).

Defining Modem Parameters on the Dialing Access Server

Perform the following steps to configure modem support for an access server initiating DDR calls (Woodstock):

- Step 1** Configure the line speed. In the following example, line speed is set to 115200 bps. If you are configuring dialout on an AUX port, the maximum speed is 38400 bps.

```
Woodstock(config)# line tty 1
Woodstock(config-line)# speed 115200
```

- Step 2** Configure flow control on the line making the outgoing DDR call.

```
Woodstock(config-line)# flowcontrol hardware
```

- Step 3** Because the access server is making outgoing calls on line 1 only, configure the modem to make outgoing calls on that line.

```
Woodstock(config-line)# modem inout
```

- Step 4** Define a chat script to send commands to the modem (note that chat scripts are case sensitive).

```
Woodstock(config)# chat-script dialnum "" "atdt\T" TIMEOUT 60 CONNECT \c
```

This script, named `dialnum`, sends the `atdt` string to the modem. The `\T` in the script specifies that the phone number that appears in the **dialer map** statement be sent (see Step 8 in the section “Configuring the Asynchronous Interface Dialing the DDR Call”).

- Step 5** Create a chat script to initialize the modem making the DDR call. In this case, the name of the chat script is `rstusr`. When this script is executed, the modem string `at&fs0=1e0&r2&d2&c1&b1&h1&m0&k0` is sent.

```
Woodstock(config)# chat-script rstusr "" "at&fs0=1e0&r2&d2&c1&b1&h1&m0&k0" "OK"
```

Other modems require similar settings, but different syntax. In this example, this script is executed by the **script reset rstusr** command, as shown in the following step.

- Step 6** Assign the chat script `rstusr` to the asynchronous line.

```
Woodstock(config)# line tty 1
Woodstock(config-line)# script reset rstusr
```

The **reset** string causes the chat script `rstusr` to be sent to the modem when the line is reset.

- Step 7** Enable pulsing DTR signal intervals on the asynchronous interface to ensure that the modem properly disconnects by using the **pulse-time** command. This command is needed on the dialing access server only.

```
Woodstock(config-line)# interface async 1
Woodstock(config-if)# pulse-time 3
```

Configuring Routing and a Routing Protocol on the Dialing Access Server

Perform the following steps to configure RIP routing on the access server initiating DDR calls (Woodstock):

- Step 1** Configure RIP routing globally on the access server:

```
Woodstock(config)# router rip
Woodstock(config-router)#
```

Step 2 Associate a network to the RIP routing process:

```
Woodstock(config-router)# network 172.16.0.0
Woodstock(config-router)# exit
Woodstock(config)#
```

Step 3 Create a static default route. A static default route points to the answering access server's IP network number (in this case 172.16.10.0) via the next hop (in this case 172.16.10.1). Static default routes are required because dynamic routes are lost when the link is down.

```
Woodstock(config)# ip route 172.16.10.0 255.255.255.0 172.16.10.1
```

Step 4 Configure a second default route, because the asynchronous interface is unnumbered (refer to the section "Configuring the Asynchronous Interface Dialing the DDR Call"). A second static route is needed to tell the local access server (Woodstock) how to get to the device at address 172.16.10.1. A mask of 255.255.255.255 is used to specify that this route is a host address.

```
Woodstock(config)# ip route 172.16.10.1 255.255.255.255 async1
```

Configuring the Ethernet Interface 0 for the Dialing Access Server

Perform the following task to configure the Ethernet interface 0 on the access server initiating outgoing DDR calls (Woodstock):

Assign an IP address to the Ethernet interface 0:

```
Snoopy(config-router)# interface Ethernet0
Snoopy(config-if)# ip address 172.16.20.1 255.255.255.0
```

Configuring the Asynchronous Interface Dialing the DDR Call

Perform the following steps to configure the asynchronous interface initiating DDR calls:

Step 1 Configure the asynchronous interface through which you need to place calls as IP unnumbered to conserve IP addresses and assign the IP address for Ethernet interface 0 to it.

```
Woodstock(config-if)# interface async1
Woodstock(config-if)# ip unnumbered Ethernet0
```

Step 2 Encapsulate PPP on the interface.

```
Woodstock(config-if)# encapsulation ppp
```

Step 3 Specify asynchronous dynamic routing on the interface. The **async dynamic routing** command allows routing protocols to be run over the asynchronous interface to resolve IP routes dynamically. If the command is omitted, static routes can still be used.

```
Woodstock(config-if)# async dynamic routing
```

Step 4 Specify the IP address of opposite access server's (Snoopy's) Ethernet interface 0 as a default IP address:

```
Woodstock(config-if)# peer default ip address 172.16.10.1
```

- Step 5** Configure the asynchronous interface as dedicated to PPP mode, which means that the access server automatically uses a PPP session for this interface. The **async mode dedicated** command enables the configured session type to start automatically when the DDR link comes up.

```
Woodstock(config-if)# async mode dedicated
```

- Step 6** Configure DDR support on the asynchronous interface using the **dialer in-band** command.

```
Woodstock(config-if)# dialer in-band
```

- Step 7** Set the number of seconds the connection remains open if no interesting traffic is being routed across this link. The timer is reset each time an interesting packet is forwarded across the DDR connection. You need to set the idle-timeout to the same value on both access servers. In this example, the line is closed after 5 consecutive minutes without interesting traffic.

```
Woodstock(config-if)# dialer idle-timeout 300
```

- Step 8** Issue the **dialer map** command. In addition to authentication on the dialing access server, this command also provides the dial string and the modem script that are used to dial the number. The command essentially maps a name, modem script, and phone number to a destination IP address.

```
Woodstock(config-if)# dialer map ip 172.16.10.1 name Snoopy modem-script dialnum  
broadcast 14085554321
```

The address 172.16.10.1 is the IP address of the answering access server's asynchronous interface. Because IP unnumbered interfaces are being used, this address is the same as the central IP address assigned to the Ethernet interface 0.

The name Snoopy is the host name of the remote access server. The name is case sensitive and must be defined as a username.

The modem-script dialnum specifies that this chat-script (**dialnum**) be sent when the access server initiates a call.

The keyword **broadcast** enables broadcast packets to be forwarded to this address (such as RIP or IGRP updates for IP and RIP and SAP updates for IPX).

The number 14085554321 is the answering access server's telephone number. This is the number to dial to reach the remote access server.

- Step 9** Associate this asynchronous interface with the dialer list 1 definition by using the **dialer-group 1** command. The interface now considers anything defined in dialer list 1 as interesting traffic.

```
Woodstock(config-if)# dialer-group 1
```

Configuring Security for the Dialing Access Server

Perform the following steps to configure security on an access server initiating DDR calls (Woodstock):

- Step 1** Specify the name of the access server answering a call (Snoopy) in Woodstock's username database. This username is referenced in the **dialer map** command for authentication purposes. The username is case sensitive and must match the opposite access server's host name exactly. The password (peanuts) is used as the PPP authentication password for the user Snoopy. It is also case sensitive:

```
Woodstock(config)# username Snoopy password peanuts
```

Note If you enter the password peanuts, exit to privileged EXEC mode, and issue the **show running-config** command, the output of this command shows up with an encrypted password, similar to the following: `username Snoopy password 7 0215055500070C294D`. When you enter or make changes to the **username** command, always enter the password in its unencrypted form. Do not enter the encryption type (7). It is set automatically.

- Step 2** Create a PPP authentication list:

```
Woodstock(config)# aaa authentication ppp default local
Woodstock(config)# aaa authentication login default local
```

- Step 3** Apply the PPP authentication list to the asynchronous interface initiating DDR calls and specify CHAP authentication (rather than PAP):

```
Woodstock(config)# interface async 1
Woodstock(config-if)# ppp authentication chap default
```

- Step 4** Require login authentication on VTY lines 0 through 4. The **login authentication default** command uses the **aaa authentication default local** authentication list. The **local** keyword means that the local username database is used for security. On this access server, only five VTY lines have been defined.

```
Snoopy(config-if)# line vty 0 4
Snoopy(config-line)# login authentication default
```

- Step 5** Apply login authentication to TTY lines 1 to 16 on the access server.

```
Woodstock(config-if)# line 1 16
Woodstock(config-line)# login authentication default
```

- Step 6** Create access list filters. In this example, the packets that the access list permits are referenced by the **dialer-list** command (in Step 7 in this procedure) to determine interesting packets and activate a call. The access list you create depends on your particular network design.

```
Woodstock(config-line)# exit
Woodstock(config)# access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255
0.0.0.0
Woodstock(config)# access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0
255.255.255.255
```

The line `access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0` specifies that all broadcast packets are uninteresting.

The line `access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255` specifies that all other IP packets are interesting.

- Step 7** Reference the filter defined by access list 100. Packets permitted by access list 100 are considered interesting packets for a DDR interface belonging to dialer group 1.

```
Woodstock(config)# dialer-list 1 list 100
```

The **dialer-list** command points to the list of commands that belong to access list 100. Packets defined by access list 100 are interesting packets for any interface belonging to dialer-group 1. The **dialer-list** command is similar to the **dialer-list 1 protocol ip permit** command on the answering access server. However, the **dialer-list 1 list 100** command does not allow broadcast packets to keep the line up.

- Step 8** Specify a password (test in this example) on VTY lines 0 through 4. On this access server, only five VTY lines have been defined.

```
Woodstock(config-if)# line vty 0 4
Woodstock(config-line)# password test
```

- Step 9** Enable login to VTY lines 0 through 4:

```
Woodstock(config-line)# login
```

You have configured the dialing access server. To ensure the dial-on-demand function works, perform a task that requires your dialing access server to place a call to your answering access server.

Sample Configurations for Routing Across Modem Lines

This section shows sample output for access servers set up for unnumbered IP dial-on-demand routing on an asynchronous interface. These sample configurations are based on the steps you followed in the preceding sections of this chapter to configure the answering and dialing access servers.

Sample Configuration for the Answering Access Server

The following sample configuration is for the answering access server (Snoopy):

```
Current configuration:
!
version 12.0
!
hostname Snoopy
!
enable password test
!
aaa authentication ppp default local
!
username Woodstock password 7 kd345096ix09ghu934c=e
!
interface Ethernet0
 ip address 172.16.10.1 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Async1
 ip unnumbered Ethernet0
 encapsulation ppp
```

```

peer default ip address 172.16.20.1
async dynamic routing
async mode dedicated
dialer idle-timeout 300
dialer map ip 172.16.20.1 name Woodstock broadcast
ppp authentication chap
dialer-group 1
!
router rip
network 172.16.0.0
!
access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 100
!
ip route 0.0.0.0 0.0.0.0 172.16.20.1
ip route 172.16.20.1 255.255.255.255 async1
!
line con 0
line aux 0
modem dialin
speed 115200
flowcontrol hardware
line vty 0 4
password cisco
!
end

```

Sample Configuration for the Dialing Access Server

The following sample configuration is for the dialing access server (Woodstock):

```

Current configuration:
!
version 12.0
!
hostname Woodstock
!
enable password test
!
username Snoopy password peanuts
chat-script dialnum "" "atdt\T" TIMEOUT 60 CONNECT \c
chat-script rstusr "" "at&fs0=1e0&r2&d2&c1&b1&h1&m0&k0" "OK"
!
interface Ethernet0
ip address 172.16.20.1 255.255.255.0
!
interface Serial0
no ip address
!
interface Serial1
no ip address
!
interface Async1
ip unnumbered Ethernet0
encapsulation ppp
async default ip address 172.16.10.1
async dynamic routing
async mode dedicated
dialer in-band
dialer idle-timeout 300
dialer map ip 172.16.10.1 name Snoopy modem-script dialnum broadcast 14085554321
dialer-group 1

```

Sample Configurations for Routing Across Modem Lines

```
ppp authentication chap
pulse-time 3
!
router rip
 network 172.16.0.0
!
ip route 0.0.0.0 0.0.0.0 172.16.10.1
ip route 172.16.10.1 255.255.255.255 async 1
!
access-list 100 deny ip 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 100 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 100
!
line con 0
line aux 0
 modem InOut
 speed 115200
 script reset rstusr
 flowcontrol hardware
!
line vty 0 4
 password test
 login
!
end
```