CHAPTER 3

Dial Solutions

This chapter shows you how to configure and use the following network security features:

- **3-1: Modems**—Cisco routers and access servers can use internal or external modems to place and receive calls. Both digital and analog modems can be used. This section presents information on how to configure and manage all types of modems.
- **3-2: ISDN**—A router can provide dial-in or dial-out service over an ISDN PRI (23 or 30 channels) or an ISDN BRI (two channels).
- **3-3: Dial-on-Demand Routing (DDR)**—Calls can be placed dynamically from a router to a dialup destination based on specific traffic. DDR provides very flexible and scalable support for multiple dialing interfaces used to reach a destination. Connections can be made and taken down as needed, using dialup resources only when they are necessary.
- **3-4: Dial Backup**—Dialer interfaces can be used to back up other more permanent or robust interfaces on a router. When the interface goes down, DDR is used to bring up a backup connection automatically. Backup connections can also be made based on the availability of a remote IP route or IP address.
- **3-5: Routing Over Dialup Networks**—When dialup connections are used between two routers, routing protocol traffic might cause the connection to stay up indefinitely. Snapshot routing performs only periodic routing updates between routers. Routing table entries are frozen until the next periodic update, keeping the dialup connection down unless it is needed for regular traffic.
- **3-6: Point-to-Point Protocol (PPP)**—PPP can be used on serial interfaces to transport other network-layer protocols over a point-to-point link. PPP offers authentication between endpoints before a connection can be established. Multilink PPP allows multiple physical links to be made, bundling the links into a logical path. Packets are fragmented and distributed over the bundled links. Fragmentation and interleaving can also be used to allow deterministic transport of time-critical data such as voice traffic.

3-1: Modems

- Internal modems include Modem ISDN Channel Aggregation (MICA) and NextPort SPE digital modems and network module analog modems.
- Internal modems can be grouped into pools such that each pool can be used for a different purpose. Users dialing into a pool receive a Dialed Number Identification System (DNIS) number and a guaranteed maximum number of simultaneous connections.
- The Call Tracker feature can be used to gather and record detailed statistics about active and disconnected calls. The statistics can be retrieved through the command-line interface, Syslog, or SNMP.
- External modems can be used when they are connected to asynchronous lines (console, Aux, or line) on routers and access servers.

Configuration

1 (Optional) Use internal digital modems.

a. Set the country code for the modems:

```
(global) modem country {mica | microcom_hdms} country
```

-OR-

```
(global) spe country country
```

The digital modem type is given as **mica** (MICA), **microcom_hdms** (Microcom), or **spe** (NextPort). The *country* code must be one of **argentina**, **australia**, **austria**, **belgium**, **brazil**, **canada**, **chile**, **china**, **columbia**, **czech-republic**, **denmark**, **europe**, **finland**, **france**, **germany**, **greece**, **hong-kong**, **hungary**, **india**, **indonesia**, **israel**, **italy**, **japan**, **korea**, **malaysia**, **mexico**, **netherlands**, **norway**, **peru**, **philippines**, **poland**, **portugal**, **saudi-arabia**, **singapore**, **south-africa**, **spain**, **sweden**, **switzerland**, **taiwan**, **thailand**, **united-kingdom**, or **usa**. The MICA and NextPort modems also add country codes: **e1-default** (default E1, a-Law), **russia**, **t1-default** (default T1, u-Law), and **turkey**.

b. (Optional) Group modems into a logical pool.

— Create a modem pool:

```
(global) modem-pool pool-name
```

By default, all internal modems are members of a system default pool. Grouping modems into a logical or virtual pool allows each pool to be used for different purposes.

— Define the range of modems in the pool:

```
(modem-pool) pool-range low-high
```

The range is defined as *low* (the lowest numbered modem line), a dash, and *high* (the highest numbered modem line). To find the modem line numbers, you can use the **show modem** command.

— Define one or more DNIS numbers for the pool:

```
(modem-pool) called-number dnis-number [max-conn connections]
```

When a user dials into an access server, the router uses the DNIS number (the number that was called) to find the appropriate modem pool. Each user accessing the pool can be limited to *connections* simultaneous connections.

c. (Optional) Use Call Tracker to gather statistics from internal modems.

NOTE You can view the Call Tracker history from the command line by using the **show call calltracker active** (active calls) and the **show call calltracker history** (disconnected calls) commands. To see detailed information about the last call on a specific modem, use the **show modem calltracker** [*slot/port*] command.

— Enable Call Tracker:

```
(global) calltracker enable
```

— Specify the Call Tracker history:

```
(global) calltracker history max-size number
(global) calltracker history retain-mins minutes
```

The maximum number of call entries that are recorded can be set to *number* (0 to 10 times the number of DS0 channels supported on the router; the default is 1 times the maximum number of supported DS0s). A value of 0 prevents call entries from being recorded. Call entries can also be retained in memory for *minutes* (0 to 26000 minutes; the default is 5000). A value of 0 prevents the call history from being saved. When increasing the history size, choose the values carefully so that Call Tracker doesn't consume too much router memory.

— (Optional) Collect statistics on MICA modems:

```
(global) modem link-info poll time seconds
```

Call Tracker can poll MICA modems for statistics at intervals of *seconds* (10 to 65535 seconds).

— (SNMP only) Enable the Call Tracker SNMP trap:

```
(global) snmp-server enable traps calltracker
```

If SNMP is used to gather the Call Tracker history, you must enable the **calltracker** trap. See Section 1-6 for more information about SNMP and Syslog configuration.

d. (Optional) Busy out or disable modems.

— Disable a single modem:

```
(line) modem bad
```

-OR-

```
(line) modem shutdown
```

You can remove an idle modem from service with the **bad** keyword. A bad active modem can be abruptly shut down with the **shutdown** keyword.

— Use modem recovery to detect and recover from faults.

To detect a faulty modem, enter the following command:

```
(global) modem recovery threshold number
```

After *number* (1 to 1000 attempts; the default is 30 call attempts) attempts are made to use an unresponsive modem, the modem recovery process is started.

To define the amount of time before a modem is considered unresponsive, enter the following command:

```
(global) modem recovery-time minutes
```

A modem is considered locked and unresponsive after *minutes* (the default is 5 minutes) have passed since a call request was made.

To define the type of recovery action to take, enter the following command:

```
(global) modem recovery action {disable | download | none}
```

When modem recovery enters the maintenance window, it attempts to recover faulty modems. The recovery action can be **disable** (mark a faulty modem as bad and disable it), **download** (schedule the modem for a firmware download), or **none** (don't try to recover a faulty modem; keep using it).

To define the automatic modem recovery process, enter the following command:

```
(global) modem recovery maintenance {action {disable | drop-call |
  reschedule} | max-download number | schedule {immediate | pending} |
  time hh:mm | window minutes}
```

Every 24 hours, the router performs the modem recovery maintenance process on modems that have been marked as faulty. Faulty modems have their firmware reloaded in an attempt to bootstrap them. All

modems on the same module as a faulty modem are first flagged as being in the "recovery pending" state. This prevents some modems with active calls from being reinitialized with a firmware download. The **window** keyword defines a maintenance window for *minutes* (the default is 60 minutes); as soon as the window timer expires, the pending modems are recovered.

As soon as the window is expired, an **action** is taken: **disable** (mark the faulty modems as bad and return other modems to service), **drop-call** (drop any active calls and reload the firmware), or **reschedule** (reschedule the firmware reload until the next maintenance window; this is the default). The maintenance window can be set with the **schedule** keyword, as **immediate** (attempt modem recovery now) or **pending** (wait until the next window 24 hours later; this is the default). The time of day for the regular maintenance window can be set with the **time** keyword, as *hh:mm* (a 24-hour time format; the default is 3:00 a.m.).

To prevent a large number of modems from being disabled and reloaded at one time, you can use the **max-download** keyword to place a limit on the number of modems that will be recovered, as *number* (1 to 30; the default is 20 percent of the number of modems on a system).

2 Set modem parameters (internal or external modems).

a. Define a range of lines to use with modems:

```
(global) line start-line end-line
```

b. Define the connection protocols that can be used over the lines:

```
(line) transport {input | output} {all | none}
```

c. (Optional) Automatically start up an async protocol:

```
(line) autoselect {arap | ppp | slip}
```

When a connection is made to a modem line, one or more of the protocols **arap** (AppleTalk Remote Access Protocol), **ppp**, and **slip** can be started.

d. Select the modem control mode:

```
(line) modem {dialin | inout}
```

The modem can be configured to allow incoming connections only (**dialin**) or to allow both incoming and outgoing connections (**inout**).

e. Define the modem initialization string.

— (Optional) Use an existing modem definition.

To see if the router has a preconfigured entry for your modem, enter the following command:

```
(exec) show modemcap [modem-type]
```

A list of the preconfigured modem types is shown. As soon as you find a modem type in the list, you can view all the preconfigured attributes for it by adding the *modem-type* to the command.

To edit a preconfigured modem type if needed, enter the following optional command:

```
(global) modemcap edit modem-name attribute at-command
```

The *modem-name* must be one of the types listed from **show modemcap**. The *attribute* is the name of an attribute listed in the **show modemcap** *modem-type* command. The *at-command* is a string of characters that define the "AT-style" modem command you want to use for the attribute.

To apply the modem type to the line, enter the following command:

```
(line) modem autoconfigure type modem-name
```

For internal digital modems, the type of modem can be defined as **microcom_server** (Cisco Microcom V.34/56k on an AS5200), **cisco_v110** (Cisco NEC internal V.110 TA on an AS5200), **microcom_mimic** (Cisco Microcom internal analog modem on an NM-AM), **mica** (Cisco MICA), **nextport** (Cisco NextPort CSMV/6), or **microcom_hdms** (Microcom HDMS chassis).

For external modems, the type of modem can be defined as **default** (generic Hayes-compatible), **codex_3260** (Motorola Codex 3260), **usr_courier** (US Robotics Courier), **usr_sportster** (US Robotics Sportster), **hayes_optima** (Hayes Optima), **global_village** (Global Village Teleport), **viva** (Viva Rockwell ACF with MNP), **telebit_t3000** (Telebit T3000), **nec_v34** (NEC V.34), **nec_v110** (NEC V.110 TA), or **nec_piafs** (NEC PIAFS TA).

The modem initialization string is preconfigured in the IOS software and is reapplied to the modem every time the line goes down. To see a complete list of the preconfigured modem types, use the **show modemcap** command.

— (Optional) Automatically discover the modem type:

```
(line) modem autoconfigure discovery
```

Each time the line goes down, the router sends a string of commands to the modem in an attempt to discover the modem's type. The router can discover only modem types that are already configured (**show modemcap**). If the modem type can't be discovered, the router retries for 10 seconds. To see the results of the discovery, use the **show line** *line* command. Be aware that discovery mode is considerably slower than using a modem type that is manually configured.

f. (Optional) Place modem lines into a rotary group:

```
(line) rotary group
```

The line is identified with the rotary group numbered *group*. It becomes part of a hunt group for outgoing calls.

3 (Optional) Create a chat script for interaction with a modem or remote system.

a. Define the chat script:

```
(global) chat-script script-name expect-send-string
```

A chat script is created with the name *script-name* (text string). You can choose a script name that corresponds to a modem vendor, type, and modulation, as in *vendor-type-modulation*. When chat scripts are used, the name of the script can be wildcarded so that a matching chat script name will be selected.

The actual chat script consists of an *expect-send-string* (text string), which consists of pairs of strings. The *expect* string is expected to be received from the modem or remote system, and the *send* string is to be sent back in response.

A chat script can contain special escape sequences, as shown in Table 3-1.

Table 3-1 *Chat Script Escape Sequences*

Escape Sequence	Description
\value	Sends the ASCII character that has the value in octal (three digits of 0 to 7).
\\	Sends a backslash (\) character.
\"	Sends a double-quote (") character.
\c	Suppresses a new line at the end of the send string.
\d	Delays for 2 seconds.
\K	Inserts a BREAK.
\n	Sends a newline or linefeed character.
\N	Sends a null character.
\p	Pauses for 0.25 seconds.
\q	Reserved.
\r	Sends a return.
\s	Sends a space character.
\t	Sends a tab character.
\T	\T is replaced by the phone number from a dial string.
""	Expects a null string.
BREAK	Causes a BREAK.

continues

Table 3-1 *Chat Script Escape Sequences (Continued)*

Escape Sequence	Description
EOT	Sends an end-of-transmission character.
ABORT *string*	Expects a string and indicates an aborted condition.
TIMEOUT *time*	Sets the time to wait for an expected input for *time* seconds (the default is 5).

Here is a sample chat script:

```
chat-script DialRemote ABORT ERROR ABORT BUSY ABORT "NO ANSWER" ""
  "AT H" OK "ATDT \T" TIMEOUT 45 CONNECT \c
```

The script is named DialRemote, and it aborts if either **ERROR**, **BUSY**, or **"NO ANSWER"** is received from the modem. The script expects to see nothing (two double quotes), and then it sends **"AT H"** to force the modem to go on-hook. As soon as **OK** is received from the modem, the script sends **"ATDT \T"** to dial the digit string that is sent in place of the **\T** characters. The script waits for a maximum timeout of 45 seconds, expecting to see **CONNECT** from the modem. The **\c** causes the script to suppress a newline character as the final send string.

b. Use the chat script during a line event:

```
(line) script {activation | connection | dialer | reset | startup} regexp
```

A chat script that matches the *regexp* regular expression is used to communicate with a modem or remote system over the line when the specified line event occurs. Line events can be **activation** (when the line is activated with a new EXEC session), **connection** (when a network connection is made on the line), **dialer** (when DDR triggers an outbound call; used with a modem), **reset** (when the line is reset), or **startup** (when the router is started up).

NOTE If a chat script is not working properly, you can use the **debug chat line** *line-number* command to watch the interactive expect-send process.

3-2: ISDN

- Primary Rate Interface (PRI) has 23 B (bearer) channels and one D (data) channel in North America and Japan, and 30 B channels and one D channel in the rest of the world. These are usually known as the 23B+D and 30B+D formats.
- Each B channel carries a 64 kbps timeslot (data or voice).
- The D channel is also a 64 kbps timeslot, carrying signaling for all the B channels. On a 23B+D PRI, the D channel is found in channel 23. On a 30B+D PRI, it is in channel 15.

- Digital calls over a B channel are handled by the ISDN processor in the router, and analog modem calls are handled by the on-board modems in the router.
- Basic Rate Interface (BRI) has two B channels and one D channel. Each B channel is 64 kbps, and the D channel is 16 kbps, for a total of 144 kbps. This is known as the 2B+D format.
- BRI interfaces can use a service profile identifier (SPID) number for identification. It is assigned by the service provider.

PRI Configuration

1 Set the global ISDN switch type for all PRI interfaces:

```
(global) isdn switch-type switch-type
```

The ISDN *switch-type* must be set to match the switching equipment being used by the telephony provider. In North America, use **basic-5ess** (Lucent basic rate switches), **basic-dms100** (NT DMS-100 basic rate switches), or **basic-ni1** (National ISDN-1). In Australia, use **basic-ts013** (TS013). In Europe, use **basic-1tr6** (German 1TR6), **basic-nwnet3** (Norwegian NET3 phase 1), **basic-net3** (NET3), **vn2** (French VN2), or **vn3** (French VN3). In Japan, use **ntt** (NTT). In New Zealand, use **basic-nznet3** (New Zealand NET3).

NOTE To use QSIG signaling, use a *switch-type* of **basic-qsig**.

2 Configure the T1/E1 controller.

a. Select the controller:

```
(global) controller {t1 | e1} slot/port
```

-OR-

```
(global) card type {t1 | e1} slot
```

A T1/E1 controller is referenced by **controller** and *slot* and *port* number on 2600 and 3600 routers and by **card type** and *slot* number on 7200, 7500, and AS5x00 routers and access servers.

b. (Optional) Set the ISDN switch type for this PRI:

```
(global) isdn switch-type switch-type
```

The switch type can also be set on a per-PRI basis, overriding the global switch type.

c. Set the framing type:

```
(controller) framing {sf | esf | crc4 | no-crc4} [australia]
```

The T1 framing type can be **sf** (super frame, the default) or **esf** (extended super frame). The E1 framing type can be **crc4** (the default), **no-crc4**, and an optional **australia**.

d. Set the clock source:

```
(controller) clock source {line [primary | secondary] | internal}
```

The controller can derive its clock from **line** (CO or external source) or **internal** (the controller's internal clock). A line clock can be designated as **primary** (preferred over other controllers' line clocks) or **secondary** (used as a backup external clock source).

e. Set the line encoding:

```
(controller) linecode {ami | b8zs | hdb3}
```

For a T1, the line coding can be set to **ami** (the default) or **b8zs** (binary 8 zero substitution). For an E1, it can be set to **ami** or **hdb3** (the default; high-density bipolar 3).

3 Configure the PRI group:

```
(controller) pri-group [timeslots range]
```

The voice timeslots are identified as a *range* (numbers 1 to 23 or 1 to 30, separated by a dash or comma). If the **timeslots** and *range* keywords are not used, the default is a PRI with 23 B channels and one D channel.

NOTE To reference the serial interfaces corresponding to the PRI channels, use **interface serial** *controller:channel*, where *controller* is the T1 controller number for the physical connection and *channel* is 0 to 22 for B channels 1 to 23 and 23 for the D channel. (Interface channel numbering begins at 0, and T1/E1 channel numbering begins at 1.) For an E1 controller, the *channel* is 0 to 30 for B channels and 15 for the D channel.

When you are configuring a PRI for dial-related features, always configure the features on the D channel.

4 (Optional) Set the B channel ordering for outgoing calls.

a. Select the D channel interface:

```
(global) interface serial controller:[23 | 15]
```

The D channel is identified as channel 23 for a T1 PRI and channel 15 for an E1 PRI.

b. Set the order:

```
(interface) isdn bchan-number-order {ascending | descending}
```

The first available B channel can begin with B1 in **ascending** order or B23/B30 in **descending** order (the default). Make sure the order used matches that of your service provider.

5 Set other optional parameters.

a. (Optional) Use TEI negotiation:

```
(interface) isdn tei [first-call | powerup]
```

By default, TEI negotiation occurs when the router is powered up (**powerup**). In Europe or when connecting to a DMS-100 ISDN switch, you might need to perform the negotiation during the first active call (**first-call**).

b. (Optional) Send a calling number with outbound calls:

```
(interface) isdn calling-number calling-number
```

The *calling-number* (a string of digits) represents a telephone number to be used as a billing number by the service provider.

PRI Example

A T1 controller is configured for ISDN PRI use. ESF framing and B8ZS line coding are used. The clock source is the line, and controller T1 0 is the primary source. The PRI consists of timeslots 1 to 24 (the entire T1 format), with the D channel on timeslot 24 (or interface channel 23).

```
controller T1 0
       framing esf
       clock source line primary
       linecode b8zs
       pri-group timeslots 1-24
interface serial 0:23
```

BRI Configuration

1 Set a global ISDN switch type for all BRI interfaces:

```
(global) isdn switch-type switch-type
```

The ISDN *switch-type* must be set to match the switching equipment being used by the telephony provider. In North America, use **basic-5ess** (Lucent basic rate switches), **basic-dms100** (NT DMS-100 basic rate switches), or **basic-ni1** (National ISDN-1). In Australia, Europe, and the UK, use **basic-1tr6** (German 1TR6), **basic-net3** (NET3), or **vn3** (French VN3). In Japan, use **ntt** (NTT). All other areas should use **none** (no specific definition).

2 Select a BRI interface:

```
(global) interface bri number
```

The BRI *number* is the physical location on the router.

3 (Optional) Set the ISDN switch type for the BRI:

```
(global) isdn switch-type switch-type
```

The selected BRI interface can have its own switch type configured, overriding the global switch type.

4 (Optional) Use SPIDs:

```
(interface) isdn spid1 spid-number [ldn]
(interface) isdn spid2 spid-number [ldn]
```

If your ISDN service provider assigned SPIDs to your BRI, you must configure them on the BRI interface. One or two SPIDs can be assigned as *spid-number*, usually a seven-digit telephone number with additional optional numbers. DMS-100 and NI1 ISDN switch types require SPIDs, whereas they are optional on the 5ESS type. The service provider might also assign local directory numbers *(ldn)*, to be used when answering incoming calls.

5 Set other optional parameters.

a. (Optional) Use TEI negotiation:

```
(interface) isdn tei [first-call | powerup]
```

By default, TEI negotiation occurs when the router is powered up (**powerup**). In Europe or when connecting to a DMS-100 ISDN switch, you might need to perform the negotiation during the first active call (**first-call**).

b. (Optional) Screen incoming calls for one or more numbers:

```
(interface) isdn caller number
```

If the local ISDN switch can send caller ID (CLID) information, you can specify calling numbers that will be accepted. *number* (up to 25 digits; X is a wildcard digit) is an accepted calling telephone number.

c. (Optional) Verify the called party number:

```
(interface) isdn answer1 [called-party-number] [:subaddress]
(interface) isdn answer1 [called-party-number] [:subaddress]
```

If more than one device is attached to a BRI, the router answers only calls that are destined for either the *called-party-number* (up to 50 digits; X is a wildcard digit) or the ISDN :*subaddress* (a colon followed by a subaddress string of up to 50 digits; X is a wildcard), or both.

d. (Optional) Send a calling number with outbound calls:

```
(interface) isdn calling-number calling-number
```

The *calling-number* (a string of digits) represents a telephone number to be used as a billing number by the service provider.

e. (Optional) Set a fast rollover delay to release a B channel:

```
(interface) isdn fast-rollover-delay seconds
```

If a new ISDN call fails because a previous call hasn't yet been torn down, set the delay to *seconds* (usually 5 seconds is sufficient).

f. (Optional) Send a disconnect cause code to the ISDN switch:

```
(interface) isdn disconnect-cause {cause-code-number | busy |
  not-available}
```

By default, when a BRI connection ends, the ISDN application sends a default cause code. You can override this with a specific *cause-code-number* (1 to 127), **busy** (the USER-BUSY code), or **not-available** (the CHANNEL-NOT-AVAILABLE code).

g. (Optional) Bind a DNIS to an ISDN subaddress:

```
(interface) dialer called DNIS:subaddress
```

If it is necessary to identify a DNIS number with a specific ISDN subaddress, both can be specified as *DNIS*:*subaddress*. This is sometimes required in Europe and Australia.

BRI Example

Two ISDN BRI interfaces are used on a router. A global ISDN switch type is set for a 5ESS. Interface BRI 0 uses the default switch type, with no SPID numbers. Interface BRI 1, however, has another switch type defined for a DMS-100. Two SPIDs are configured for the two B channels:

```
isdn switch-type basic-5ess

interface bri 0
       ip address ...

interface bri 1
       isdn switch-type basic-dms100
       isdn spid1 555123401
       isdn spid2 555123402
       ip address 192.168.71.45 255.255.255.0
```

3-3: Dial-on-Demand Routing (DDR)

- DDR allows dialed calls to be made and closed dynamically, as needed.
- Outbound calls are triggered based on configurable traffic parameters.
- DDR can be configured through two methods, depending on the level of dialing flexibility that is needed. In this section, both methods are described as a single configuration procedure to make dialer configuration more intuitive and straightforward:
  - *Dialer profiles*—Logical dialer interfaces are used to reach a destination, through dial strings and dial maps. A dialer pool can be defined to make a call from one of a group of physical interfaces. Map classes define the call's characteristics and are applied to dialer profiles based on the call destination. Dialing becomes very flexible and scalable.

— *Hub and spoke*—Either physical or logical interfaces can be used to make calls. Logical dialer interfaces can be used to define rotary groups of physical interfaces. All dialer parameters are applied to the logical (or physical) interfaces, limiting the flexibility and scalability. Generally, the "spoke" DDR router makes calls to the centralized "hub" DDR router. The hub can both make and receive DDR calls.

Configuration

1 (Optional) Define a logical dialer interface for flexible dialing (Dialer Profiles).

a. Select an interface for dialing out:

```
(global) interface dialer number
```

A **dialer** is a logical interface used to support rotary groups of other physical dial interfaces. The *number* (1 to 255) is arbitrarily chosen. If a rotary group of physical interfaces is configured, the dialer *number* is also used to identify the rotary group number.

b. (Optional) Define the encapsulation for outgoing calls:

```
(interface) encapsulation ppp
```

See Section 3-6 for more configuration information on PPP encapsulation.

c. Define any network addresses needed:

```
(interface) ip address ip-address subnet-mask
```

d. Define dial destinations.

— (Optional) Call only a single destination.

Define a modem chat script. Refer to the **script dialer** command in Section 3-1 for more information.

Define the string of digits to dial by entering the following command:

```
(interface) dialer string string [class class-name]
```

The *string* is sent to the modem for dialing. It can include the digits 0 to 9, : (wait for a tone), < (pause), = (separator 3), > (separator 4), P (continue dialing in pulse mode), T (continue dialing with DTMF tones), and & (send a hookflash). If you are defining dialer map classes for more flexible per-destination characteristics, you can use the **class** keyword to apply the map class *class-name* (text string).

— (Optional) Define one or more destinations to call:

```
(interface) dialer map protocol next-hop-address [class class-name]
  [name host-name] [spc] [speed 56 | speed 64] [broadcast]
  [modem-script modem-regexp] [system-script system-regexp]
  [dial-string[:isdn-subaddress]]
```

As soon as the router determines the need to make a call, both the *protocol* (**appletalk**, **bridge**, **clns**, **decnet**, **ip**, **ipx**, **novell**, **snapshot**, **vines**, or **xns**) and the destination or *next-hop-address* are compared with the **dialer map** entries to find a match. If you are defining dialer map classes for more flexible per-destination characteristics, you can use the **class** keyword to apply the map class *class-name* (text string).

If needed, the **spc** keyword causes a semi-permanent connection to be used (Germany and Australia). If an ISDN interface is being configured, **speed 56** (56 kbps) or **speed 64** (64 kbps, the default) can be used to indicate the speed of the B channel. Use the **broadcast** keyword if broadcast packets should be forwarded to the destination.

For incoming calls, use the **name** keyword to authenticate the remote *host-name* using CHAP. See Section 3-6 for further configuration information about dial-in authentication.

For outgoing calls, the router finds a modem chat script (a **chat script** command) that matches the regular expression given by *modem-regexp* (a text string, including **.** and *****). The chat script name can consist of the **.** (match a character) and ***** (match any characters) wildcards so that a chat script can be selected according to modem or modulation type. See Section 3-1 for more information about chat scripts. If the remote system doesn't support CHAP authentication, you can use the **system-script** keyword to select a chat script that matches the regular expression given by *system-regexp* (a text string, including **.** and *****). The router sends and receives strings to navigate through a login procedure. A *dial-string* can also be given to define a string of digits to dial for this specific destination.

e. (Optional) Use a pool of physical interfaces for dialing:

```
(interface) dialer pool pool
```

Multiple physical interfaces can be assigned to a dialer pool so that they can be used for outgoing calls in a rotary fashion.

NOTE By default, the interface number of a logical dialer interface corresponds to a rotary group number. Physical interfaces can be assigned to this rotary group if desired. However, you can assign the physical interfaces to a dialer pool instead. This offers more flexibility.

f. (Optional) Define any queuing or traffic shaping.

See Chapter 10, "Quality of Service," for more information about queuing and traffic-shaping features.

g. Identify "interesting" traffic to trigger a call.

— (Optional) Use an access list to permit specific addresses and port numbers within a protocol:

```
(global) access-list acc-list-number {permit | deny} ...
```

Dialing can be triggered by packets containing a general protocol (IP, IPX, AppleTalk, and so forth) or by packets matching a more specific criteria. Any parameter that can be matched by an access list can be used to trigger a call. See Chapter 14, "Access Lists and Regular Expressions," for further configuration information about access lists for a specific protocol.

— Create a dialer list to identify the "interesting" protocol:

```
(interface) dialer-list dialer-group protocol protocol-name
  {permit | deny | list access-list-number | access-group}
```

One or more **dialer-list** statements are defined as a single arbitrary *dialer-group* (1 to 255). Each statement identifies a protocol by *protocol-name* (**appletalk**, **bridge**, **clns**, **clns_es**, **clns_is**, **decnet**, **decnet_router-L1**, **decnet_router-L2**, **decnet_node**, **ip**, **ipx**, **vines**, or **xns**). Dialing is triggered based on an action taken on the protocol: **permit** (trigger dialing on the protocol as a whole), **deny** (don't trigger dialing for the protocol as a whole), or **list** (trigger dialing if the access list numbered *access-list-number* permits the packet). The *access-group* parameter can be used to identify a filter list name for CLNS traffic.

— Apply the dialer list to a dialer interface:

```
(interface) dialer-group group
```

The dialer list numbered *group* is used to trigger an outbound call. When traffic is destined for a dialer interface, it is first filtered through the dialer list. If the packet is permitted, the call is placed.

— (IPX only) Use IPX spoofing to keep an idle connection from dialing. You have two options with IPX spoofing.

To use watchdog spoofing, enter the following command:

```
(interface) ipx watchdog {filter | spoof [enable-time-hours
  disable-time-minutes]}
```

Novell IPX servers send periodic watchdog packets to their clients every 5 minutes or so. This type of traffic can trigger a new call to be placed when the connection would otherwise be idle or disconnected. The **ipx watchdog** command causes the router to discard (**filter**) watchdog packets or answer (**spoof**) them on behalf of the clients.

Spoofing can occur for *enable-time-hours* (1 to 24 hours) and can be disabled for *disable-time-minutes* (18 to 1440 minutes). This prevents a new call from being triggered until true interesting traffic is detected.

Your other option is to use SPX spoofing. The command sequence is as follows:

```
(interface) ipx spx-spoof
(interface) ipx spx-idle-time delay
```

IPX servers send SPX keepalive packets to a client every 15 to 20 seconds after detecting that the client has been idle too long. With the **ipx spx-spoof** command, the router responds to SPX keepalive packets on behalf of an idle client so that a DDR call won't be triggered. The router can wait up to *delay* seconds (the default is 60 seconds) before starting to spoof the keepalive packets.

2 Define the dialer interface parameters.

a. (Optional) Apply the parameter settings to a map class:

```
(global) map-class dialer class-name
```

A dialer map class is used to define any parameters that can be set on a dialer interface. If you are configuring flexible dialing with the logical dialer interfaces (**interface dialer** or "Dialer Profiles"), you can use map classes to selectively configure the dialer interface on a per-destination basis. A map class is selected according to a matching **dialer-string** or **dialer map** command.

-OR-

b. (Optional) Apply the parameter settings to a physical interface:

```
(global) interface [async | serial | bri] number
```

-OR-

```
(global) interface serial controller/port:[23 | 15]
```

If flexible dialing and logical dialer interfaces are not used, the dialer parameters can be applied directly to a physical interface. The dial interface can be **async** (asynchronous line), **serial** (synchronous serial), or **bri** (ISDN BRI). For a PRI interface, use **serial** with the T1/E1 controller number and port, followed by the D channel (**23** for T1 or **15** for E1).

c. (Optional) Set the call idle timer:

```
(interface) dialer idle-timeout seconds [inbound | either]
```

As soon as an outbound call is made on a dialer interface, it is terminated if no traffic is detected for *seconds* (the default is 120 seconds) in the outbound direction. If desired, **inbound** or **either** (inbound or outbound) traffic can be used to measure the idle time.

d. (Optional) Set a fast idle timer to reuse an active interface:

```
(interface) dialer fast-idle seconds
```

Dialer contention occurs when a call is in progress on an interface, and the interface is then needed for another call to a different destination. To resolve the contention, the router can detect a shorter idle time and end the first call immediately. The fast idle time can be set to *seconds* (the default is 20 seconds).

e. (Optional) Set a line-down time before an interface can be used again:

```
(interface) dialer enable-timeout seconds
```

If you have problems with phone line availability, you can hold down an interface after a call is completed or fails before attempting to dial again. Set the line-down timer to *seconds* (the default is 15 seconds).

f. (Optional) Set the length of time to wait for a carrier signal:

```
(interface) dialer wait-for-carrier-time seconds
```

When an outbound call is made, the router must wait for any modem or system chat scripts to complete and for the carrier to be detected. If calls are being terminated before the remote system is successfully connected, increase the wait-for-carrier time to *seconds* (the default is 30 seconds).

g. (Optional) Place additional calls to increase the bandwidth on demand:

```
(interface) dialer load-threshold load [outbound | inbound | either]
```

When the traffic load on an interface in a rotary group reaches the *load* threshold (1 to 255, where 1 is unloaded and 255 is 100 percent), additional calls are placed from interfaces in the rotary group. The load threshold represents the total or cumulative load over all "up" interfaces to the destination. Links are brought up or torn down whenever the traffic load rises above or falls below the threshold. The traffic direction can be taken into account as **outbound**, **inbound**, or **either**.

NOTE Remember that an ISDN BRI is handled as a rotary group of two B channels. Setting a load threshold lets the router bring up the second B channel if needed.

h. (Optional) Set the size of the dialer hold queue:

```
(interface) dialer hold-queue packets timeout seconds
```

By default, any additional packets that are destined to a call that is being established are dropped. The router can place packets in a hold queue so that they will be sent after the call is made. The hold queue can contain up to *packets* (1 to 100 packets), which are held for a maximum of *seconds*.

i. (Optional) Redial if a call fails:

```
(interface) dialer redial interval time attempts number re-enable
  disable-time
```

By default, the router makes only a single attempt to dial a destination. The router can try to redial the destination every *time* (5 to 2147483 seconds) for up to *number* times (1 to 2147483 attempts). If all the configured redial attempts fail, the interface is disabled for *disable-time* (5 to 2147483 seconds) to prevent other redial failures in the near future.

3 Define additional parameters to the physical interface.

a. Select the interface:

```
(global) interface [async | serial | bri] number
```

-OR-

```
(global) interface serial controller/port:[23 | 15]
```

If flexible dialing and logical dialer interfaces are not used, the dialer parameters can be applied directly to a physical interface. The dial interface can be **async** (asynchronous line), **serial** (synchronous serial), or **bri** (ISDN BRI). For a PRI interface, use **serial** with the T1/E1 controller number and port, followed by the D channel (**23** for T1 or **15** for E1).

b. (Optional) Define the encapsulation needed for incoming calls:

```
(interface) encapsulation ppp
```

See Section 3-6 for more configuration information on PPP encapsulation.

c. (Optional) Define the authentication needed for incoming calls:

```
(interface) ppp authentication {pap | chap}
```

See Section 3-6 for more configuration information on PPP authentication.

d. (Optional) Use a group of interfaces to reach a common destination.

— (Optional) Make the interface a member of a dialer pool:

```
(interface) dialer pool-member number [priority priority]
  [min-link minimum] [max-link maximum]
```

If the **dialer pool** command has been used on a logical dialer interface, the physical interface becomes a part of the dialer pool *number* (1 to 255). A *priority* (0 to 255; 0 is the lowest, 255 is the highest, and the default is 0) can be assigned to the interface so that it will be chosen before other dialer pool members with lower priorities.

For ISDN, a number of B channels can be reserved for the dialer pool. You can specify the *minimum* (0 to 255; the default is 0) and *maximum* (0 to 255; the default is 255) number of channels for this purpose.

-OR-

— (Optional) Make the interface a member of a rotary group for dialing:

```
(interface) dialer rotary-group group
(interface) dialer priority priority
```

Multiple interfaces can be assigned to a rotary dialer group. As soon as a logical rotary group interface is configured with the **interface dialer** command, other physical interfaces can be assigned to it. Outbound calls are made on the first available interface in the rotary group, allowing more flexible and available calling. The *group* (0 to 255) must be the same number as the dialer interface number.

By default, a rotary group selects the next available interface in the order that they are configured. However, interfaces within a rotary group can be given a *priority* (0 to 255; 0 is the lowest and 255 is the highest) so that some interfaces are used before others.

NOTE Interfaces that are members of a dialer rotary group inherit the configuration commands that were entered for the corresponding **interface dialer**. ISDN BRI interfaces are inherently part of a rotary group so that outbound calls will rotate through the B1 and B2 channels if needed.

e. (Async or sync serial only) Select the type of dialing:

```
(interface) dialer dtr
```

-OR-

```
(interface) dialer in-band [no-parity | odd-parity]
```

An asynchronous or synchronous serial interface must send dialing information out-of-band (**dtr**; non-V.25bis modems) over the DTR signal or in-band (**in-band**; V.25bis modems) over the data stream.

Example

DDR is configured on a remote branch router to dial out to the Main Office router. Dialing occurs whenever there is traffic going from the 192.168.3.0 network (the remote Ethernet network) to the 192.168.209.0 network (the Main Office server farm). A static route is configured to define the next-hop address for network 192.168.209.0 as the Main Office router at 192.168.209.1. This address is matched in the **dialer map** statement, where the CHAP remote host name, the modem chat script, and the dial string are picked up.

Note that the dial string is defined in the dialer map. The modem chat scripts have an **"ATDT \T"** send string, where the dial string replaces **\T** during the dialing process. Interfaces async 1 and async 2 are members of a common dialer pool, allowing an additional call to be made if one of the lines is busy.

```
chat-script MainOffice ABORT ERROR ABORT BUSY ABORT "NO ANSWER" "" "AT H" OK
  "ATDT \T"
```

```
 TIMEOUT 45 CONNECT \c TIMEOUT 30

chat-script HotSite ABORT ERROR ABORT BUSY ABORT "NO ANSWER" "" "AT H" OK "AT M0
" OK "ATDT \T" TIMEOUT 45 CONNECT \c TIMEOUT 30

username southbranch password letmein

interface async 1
       no ip address
       encapsulation ppp
       ppp authentication chap
       dialer in-band
       dialer pool-member 7
       dialer redial interval 60 attempts 10 re-enable 5

interface async 2
       no ip address
       encapsulation ppp
       ppp authentication chap
       dialer in-band
       dialer pool-member 7
       dialer redial interval 60 attempts 10 re-enable 5

interface dialer 1
       ip address 192.168.4.1 255.255.255.0
       dialer remote-name tic
       dialer idle-timeout 432000
       dialer map ip 192.168.209.1 name mainoffice modem-script MainOffice 5987572
       dialer pool 7
       dialer-group 1

interface ethernet 0
       ip address 192.168.3.1 255.255.255.0

ip route 192.168.209.0 255.255.255.0 192.168.209.1

access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.209.0 0.0.0.0

dialer-list 1 protocol ip permit list 101

line 1
       modem InOut
       transport input all
       speed 115200

line 2
       modem inout
       transport input all
       speed 115200
```


3-4: Dial Backup

- Dial backup monitors a specific router interface to determine the need for a backup connection. If the interface is down or if a traffic threshold is exceeded, the backup interface is brought up.
- Dialer Watch can be used to watch for the deletion of specific routes that correspond to a remote router interface or an advertised network. If no routes to the address are found, the backup interface is brought up, and a call is made.

- Dialer Watch is useful when the state of a local router interface does not reflect a failed connection.

Dial Backup Configuration

1. Identify the interface to be backed up:

```
(global) interface type slot/port
```

2. Specify the backup interface:

```
(interface) backup interface type slot/port
```

   The backup interface is a dialer interface that DDR uses to make a call.

3. (Optional) Set a traffic threshold to trigger the backup:

```
(interface) backup load {enable-threshold | never}
  {disable-threshold | never}
```

   Dial backup is triggered when the traffic load exceeds the *enable-threshold* percentage (1 to 100 percent) of the available link bandwidth. Dial backup is disabled when the traffic load falls under the *disable-threshold* percentage (1 to 100 percent) of the link bandwidth. The **never** keyword can be used to cause dial backup to never be enabled or never be disabled.

NOTE If the traffic threshold is not defined, dial backup is triggered only if the primary interface goes down.

4. (Optional) Wait until the primary interface changes state:

```
(interface) backup delay {enable-delay-period | never}
  {disable-delay- period | never}
```

   When the primary interface goes down, dial backup waits for *enable-delay-period* (the default is 0 seconds) before triggering the backup call. When the primary interface comes back up, dial backup waits for *disable-delay-period* (the default is 0 seconds) before disabling the backup call. The **never** keyword can be used to never enable or never disable dial backup based on the primary interface state.

5. Make sure DDR is configured to use the backup interface. Refer to Section 3-3 for more configuration details if needed.

Dial Backup Example

Dial backup is configured so that when interface serial 0 goes down, DDR uses interface dialer 1 to make a backup connection. Dialer 1 is a logical interface that looks to dialer pool 5 to find an available interface for a new call. Interface serial 0 identifies interface dialer 1 as the backup interface, with an enable delay of 10 seconds and a disable delay of 20 seconds. Once dial backup is triggered, DDR uses dialer list 1 to specify the "interesting" traffic to actually trigger the call. Here, the dialer list triggers by permitting any IP traffic:

```
interface dialer 1
       ip address 192.168.1.1 255.255.255.0
       encapsulation ppp
       dialer string 5551111
       dialer pool 5
       dialer-group 1

dialer-list 1 protocol ip permit

interface bri 0
       encapsulation ppp
       dialer pool-member 5

interface serial 0
       ip address 192.168.200.1 255.255.255.0
       backup interface dialer 1
       backup delay 10 20
```


Dialer Watch Configuration

1. Define one or more routes or IP addresses to watch:

   ```
   (global) dialer watch-list group-number [delay route-check initial
     seconds] ip ip-address mask
   ```

   A watch list is created with an arbitrary *group-number* (1 to 255). Each **watch-list** statement in the list specifies an IP address and a subnet mask to be watched. The IP address given must exactly match a route in the routing table, as shown by the **show ip route** command. When the primary route to that address is removed from the routing table, the Dialer Watch feature is triggered. The **delay route-check initial** keywords can be used to force the router to wait for *seconds* (1 to 2147483 seconds) after the local router powers up and initializes before considering that the primary route is missing.

2. Select a backup interface:

   ```
   (global) interface type slot/port
   ```

   This selects the interface to be used by DDR (a logical dialer interface or a physical interface).

3. Enable Dialer Watch on the backup interface:

   ```
   (interface) dialer watch-group group-number
   ```

   Dialer Watch uses the Dialer Watch list numbered *group-number* (1 to 255) to determine when to trigger a DDR call.

4. (Optional) Wait to disable the backup interface:

   ```
   (interface) dialer watch-disable seconds
   ```

   After the primary route returns to the routing table, Dialer Watch can wait for *seconds* (the default is 0 seconds) before disabling the backup DDR call.

5. Make sure DDR is configured to use the backup interface. Refer to Section 3-3 for more configuration details if needed.

Dialer Watch Example

Dialer Watch is configured so that when the primary route to network 192.168.177.0 or network 192.168.178.0 is deleted from the routing table, DDR uses interface dialer 1 to make a backup connection. Dialer 1 is a logical interface that looks to dialer pool 5 to find an available interface for a new call. Dialer 1 also must have **dialer map** statements that exactly define the watched routes so that a call can be triggered.

```
interface dialer 1
       ip address 192.168.1.1 255.255.255.0
       encapsulation ppp
       dialer map ip 192.168.177.0 5551111
       dialer map ip 192.168.178.0 5551111
       dialer pool 5
       dialer-group 1
       dialer watch-group 10

dialer-list 1 protocol ip permit
dialer watch-list 10 ip 192.168.177.0 255.255.255.0
dialer watch-list 10 ip 192.168.178.0 255.255.255.0

interface bri 0
       encapsulation ppp
       dialer pool-member 5

interface serial 0
       ip address ...
```

3-5: Routing Over Dialup Networks

- Snapshot routing reduces the time a DDR connection is kept up by only periodically exchanging routing updates.
- Snapshot routing is based on the concept of *active periods*, when routing information can be exchanged, and *quiet periods*, when a snapshot of the routing information is kept frozen.

- During a quiet period, the routing entries are frozen and cannot be changed. They also are treated as static routes and are not aged until the next active period.
- A retry period can be defined to keep the client router from waiting another complete quiet period if a DDR interface is not available. In this case, the client router can retry the active period call after the retry period has elapsed.
- A snapshot client router can dial remote server routers only during an active period to collect routing information.
- Only distance vector routing protocols are supported: AppleTalk RTMP, Banyan Vines RTP, IP RIP and IGRP, and IPX RIP and SAP.
- On-Demand Routing (ODR) allows a centralized "hub" router to collect route advertisements from "spoke" routers on stub networks. The stub routers use only static or default routes and do not run any dynamic routing protocols.
- ODR uses the Cisco Discovery Protocol (CDP) to dynamically learn about the stub routers and their directly connected networks. Therefore, no routing information is exchanged by routing protocols.

Snapshot Routing Configuration

1 Define snapshot routing on the client router.

a. Select an interface to reach a server router:

```
(global) interface type slot/port
```

The interface is usually a logical **interface dialer** or a physical DDR interface.

b. Enable client mode snapshot routing:

```
(interface) snapshot client active-time quiet-time
  [suppress-statechange-updates] [dialer]
```

On the client router, the *active-time* (5 to 100 minutes; no default; typically 5 minutes) and *quiet-time* (8 to 100000 minutes) intervals are defined. The **suppress-statechange-updates** keyword can be used to disable routing exchanges each time the DDR interface state changes from "down" to "up" or from "dialer spoofing" to "fully up." In that case, routing information is exchanged only when the active period begins, regardless of whether the DDR interface just came up. The **dialer** keyword is used to cause the client router to dial the server routers even if there is no active traffic to trigger a call. Otherwise, the client router waits for the call to be made from normal DDR activity.

c. Define one or more server routers to contact:

```
(interface) dialer map snapshot sequence-number dial-string
```

A server router is listed with a unique *sequence-number* (1 to 254), specifying the order in which the routers are to be called. The *dial-string* (string of digits) required to reach the server router must also be given.

d. Make sure DDR is configured on the dialer interfaces. You don't have to configure the triggers for interesting traffic, because the snapshot routing triggers DDR and provides the dial string. Refer to Section 3-3 for more configuration details if needed.

2 Define snapshot routing on the server router.

a. Select an interface for incoming calls from the client router:

```
(global) interface type slot/port
```

The interface is usually a physical DDR interface that can accept incoming calls.

b. Enable server mode snapshot routing:

```
(interface) snapshot active-time [dialer]
```

The server router must have the same *active-time* (5 to 100 minutes) configured as that of the client router. The **dialer** keyword can be used to allow the server router to accept calls from the client router, even when regular DDR traffic is not present.

Snapshot Routing Example

Snapshot routing is configured on a client router at a central site. The snapshot client maintains periodic routing updates with a list of three server routers at branch sites. Snapshot routing is configured with an active period of 5 minutes and a quiet period of 480 minutes, or 8 hours. The server routers are dialed during the active period, even if a DDR call is not already in progress.

The following example shows the configuration of the client router:

```
interface dialer 1
       ip address 192.168.1.1 255.255.255.0
       encapsulation ppp
       dialer pool 5
       snapshot client 5 480 dialer

interface bri 0
       encapsulation ppp
       dialer pool-member 5

       dialer map snapshot 1 8598851234
       dialer map snapshot 2 8598855678
       dialer map snapshot 3 8598859999

router rip
       network 192.168.1.0
       network 192.168.100.0
```

The following example shows the configuration of one of the server routers:

```
interface bri 0
       encapsulation ppp
```

```
        ip address 192.168.1.2 255.255.255.0
        snapshot server 5 dialer

router rip
        network 192.168.1.0
        network 192.168.200.0
```

ODR Configuration

1 Enable ODR on the hub router:

```
(global) router odr process-id
```

ODR is started and is assigned a unique process number.

NOTE Routes can be filtered as they are collected by ODR. Refer to Section 8-4 for more information. Routes can also be redistributed into or out of ODR by using the IP route redistribution commands. Refer to Section 8-3 for more information.

2 (Optional) Tune the ODR timers.

a. Adjust the CDP timer on the stub routers:

```
(global) cdp timer seconds
```

ODR depends on CDP advertisements to collect IP route prefixes from stub routers. By default, CDP advertisements occur every 30 seconds. Set the CDP interval to *seconds* for more frequent advertisements and faster route convergence.

b. Adjust the ODR timers:

```
(router) timers basic update invalid holddown flush [sleeptime]
```

The frequency that routing updates are expected from stub routers can be adjusted to provide faster convergence. The *update* parameter (default 90 seconds) sets the rate in seconds at which updates are expected. This is the fundamental timing parameter of the ODR protocol. It should be set to match the CDP advertisement interval. The *invalid* parameter (default 270 seconds) sets the interval of time in seconds after which a route is declared invalid. It should be at least three times the value of *update*. The *holddown* parameter (default 280 seconds) sets the interval in seconds during which routing information regarding better paths is suppressed. It should be at least three times the value of *update*. The *flush* parameter (default 630 seconds) is the amount of time in seconds that must pass before the route is removed from the routing table. The interval specified must be at least the sum of *invalid* and *holddown*. If it is less than this sum, the proper holddown interval cannot elapse. This results in a new route's being accepted before the holddown interval expires. The *sleeptime* parameter (default 0 milliseconds) sets the interval in milliseconds for postponing routing

updates in the event of a Flash update. The *sleeptime* value should be less than the *update* time. If the *sleeptime* is greater than the *update* time, routing tables become unsynchronized.

c. Define a default route on the stub routers:

```
(global) ip route 0.0.0.0 0.0.0.0 interface-name
```

A stub router needs only a statically defined default route pointing to an interface that connects to the hub router.

3-6: Point-to-Point Protocol (PPP)

- PPP encapsulates network layer packets for transport over point-to-point links.
- PPP is supported on DDR interfaces as well as fixed point-to-point interfaces.
- Authentication is supported over PPP using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP).
- With CHAP or PAP authentication, users and other routers can be properly authenticated when they connect. In addition, a router host name is used during authentication and to prevent additional calls if it is already connected.
- CHAP offers a handshake procedure by using the local host name in a challenge to the remote user or router. The remote side responds with its host name and an encrypted password. The passwords and encryption methods used must match on both ends of the connection.
- PAP requires the remote user or router to send both a username and a password to be authenticated. There is no challenge or shared secret passwords, and passwords are also sent in the clear with no encryption.
- Link Quality Monitoring (LQM) compares the PPP packets exchanged between two peers. When the percentage of good packets falls below a threshold, the PPP connection is torn down.
- IP addresses can be negotiated using the IP Control Protocol (IPCP). A router can assign addresses to dial-in peers by proxying DHCP requests and replies, from a locally defined IP address pool, or from a static IP address configuration.
- Multilink PPP (MLP) uses multiple PPP links to provide load balancing and packet fragmentation. The multiple links can be brought up in succession, depending on traffic thresholds.
- PPP callback can be used to have a calling (client) router request that the destination (server) router call it back. The client router's authentication information is given to the server router. If it can be authenticated, and a dial string is found for the return call, the call is reversed.

Configuration

1. Enable PPP encapsulation on an interface:

```
(interface) encapsulation ppp
```

PPP encapsulation can be enabled on both logical and physical dialer interfaces, asynchronous, synchronous serial, HSSI, and ISDN interfaces.

2. (Async interfaces only) Select the interface mode:

```
(interface) async mode {dedicated | interactive}
```

In **dedicated** mode, the interface uses PPP encapsulation continuously. A user connected to the interface does not receive a login prompt or an EXEC session with the router. In **interactive** mode, a user receives normal login and password prompting (if configured) and is presented with an EXEC session on the router. The user is free to use other router commands and must then issue the **ppp** command to start PPP encapsulation.

3. (Optional) Use PPP authentication:

```
(interface) ppp authentication {protocol1 [protocol2...]} [if-needed]
  [list-name | default] [callin] [one-time]
```

PPP can use one or more authentication protocols (**chap**, **pap**, or **ms-chap**), listed in the order that they are tried. The **ms-chap** method can be used for CHAP between a router and a Microsoft Windows device. If you are using AAA with TACACS+ authentication, the **if-needed** keyword prevents PAP or CHAP from being used when the user is already authenticated. If configured, either the *list-name* AAA method list (**aaa authentication ppp**) or the default method list is used to perform AAA authentication. The **callin** keyword performs authentication only on incoming calls. Rather than giving the username and password separately, the **one-time** keyword can be used to present both at once.

NOTE If AAA authentication is not used, you must configure the remote router host names and their shared secret passwords before CHAP will work. Use **username** *name* **password** *password* to define the router's host name as the username. For further information, refer to Section 1-1 for local username authentication, and refer to Section 13-1 for AAA authentication configuration.

4. (Optional) Use PPP callback to increase security or to lower toll costs.

   a. (Optional) Request a callback from a remote peer (callback client):

```
(interface) ppp callback request
```

As soon as a call is made to a peer router, the local router requests that it be called back to complete the PPP connection.

b. (Optional) Accept a callback request (callback server):

```
(interface) ppp callback accept
(interface) dialer callback-server [username] [dialstring]
(interface) dialer callback-secure
```

The router accepts incoming calls that request PPP callback service. If the incoming peer router is authenticated by PPP, the call is completed. Then the local router initiates a call back to the requesting router. If authentication is successful, the PPP connection is established. The **username** keyword (the default) can be given to look up and authenticate the router host name in a **dialer map** command. The **dialstring** keyword is used to identify the return call during callback negotiation. The **callback-secure** keyword can be used to drop callback requests if the username or host name cannot be authenticated and approved for callback.

5 (Optional) Use LQM on the PPP interface:

```
(interface) ppp quality percentage
```

The number of PPP packets sent and received is compared to the number collected by the remote router. If the percentage of successful packet transfers falls below the *percentage* (1 to 100) threshold, the PPP link is shut down.

6 (Optional) Assign IP addresses over PPP.

a. (Optional) Use DHCP proxy to relay an address.

— Enable DHCP proxy:

```
(global) ip address-pool dhcp-proxy-client
```

The router accepts DHCP requests from the far end and relays them to a DHCP server.

— Relay requests to one or more DHCP servers:

```
(global) ip dhcp-server [ip-address | name]
```

The router relays DHCP requests to the server identified by *ip-address* or the host name. Up to ten DHCP servers can be configured.

— Assign an address over the PPP interface:

```
(interface) peer default ip address pool dhcp
```

The far-end PPP peer receives an address from the DHCP server.

b. (Optional) Use an address from a local pool.

— Enable a local address pool:

```
(global) ip address-pool local
```

The router uses a pool of IP addresses from its own configuration.

— Define the address pool:

```
(global) ip local pool {default | pool-name} low-ip-address
  [high-ip-address]
```

The pool can be named either **default** or *pool-name* (a text string). The range of IP addresses starts at the lowest IP address, *low-ip-address*, and ends at the highest IP address, *high-ip-address*. If the upper limit is not given, the pool consists of a single address.

— Assign an address over the PPP interface:

```
(interface) peer default ip address pool pool-name
```

The router assigns the PPP peer an address from the locally defined pool called *pool-name*.

c. (Optional) Assign a specific IP address over the PPP interface:

```
(interface) peer default ip address ip-address
```

Use this method if there are few IP addresses to assign to dial-in peers.

NOTE If the remote dial-in PPP peer is another router, you must configure the remote PPP interface to accept a negotiated IP address. Use the (interface) **ip address negotiated** command.

d. (Optional) Permit routing protocol traffic to pass over an asynchronous interface:

```
(interface) async dynamic routing
```

By default, no dynamic routing protocol traffic is allowed to pass over an asynchronous interface. Use this command if you need to exchange routing information over an asynchronous PPP interface.

7 (Optional) Use Multilink PPP (MLP).

a. Enable MLP on one or more interfaces:

```
(interface) ppp multilink
```

Usually, MLP is configured on a logical dialer interface so that all physical interfaces used in a rotary group or dialer pool are added to the bundle.

b. (Optional) Use MLP interleaving to fragment large packets:

```
(interface) ppp multilink interleave
(interface) ppp multilink fragment delay milliseconds
```

Packets destined for an MLP interface are fragmented and distributed across the links in the MLP bundle. The size of the fragments is governed by the required fragment transmission time *milliseconds* (1 to 1000 milliseconds; the default is 30). This feature is especially useful for the delivery of time-critical protocols such as voice traffic. Voice traffic requires a maximum packet serialization delay of 10 milliseconds.

Example

A router is configured for DDR using two ISDN interfaces—BRI 0 and BRI 1. A logical dialer interface is configured for PPP encapsulation, CHAP authentication, and Multilink PPP. The dialer interface uses a dialer pool that consists of both BRI interfaces. Additional B channels are brought up when the overall traffic load of the interfaces in use reaches 50 percent (or a threshold value of 255 times 50 percent, or 128). Multilink PPP fragmentation and interleaving are also configured to allow time-critical traffic such as voice to receive a guaranteed transmission delay of 10 milliseconds.

```
username Remote password letmein

interface dialer 1
       ip address 192.168.254.1 255.255.255.0
       encapsulation ppp
       dialer in-band
       dialer load-threshold 128
       dialer-group 10
       ppp authentication chap
       ppp multilink
       ppp multilink fragment delay 10
       dialer pool 5

interface bri 0
       encapsulation ppp
       dialer pool-member 5
       dialer load-threshold 128

interface bri 1
       encapsulation ppp
       dialer pool-member 5
       dialer load-threshold 128
```

Further Reading

Refer to the following recommended sources for further information about the topics covered in this chapter:

Building Cisco Remote Access Networks, by Catherine Paquet, Cisco Press, ISBN 1578700914

CCNP Remote Access Exam Certification Guide, by Brian Morgan and Craig Dennis, Cisco Press, ISBN 1587200031

CCNP Semester Six Companion Guide, Remote Access, Cisco Network Academy Program, Cisco Systems, ISBN 1587130289

CCNP Semester Six Lab Companion Guide, Remote Access, Cisco Network Academy Program, Cisco Systems, ISBN 1587130327