CHAPTER 4

Session Initiation Protocol

The Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF) standard call control protocol, based on research at Columbia University by Henning Schulzrinne and his team. The first SIP RFC, number 2543, was published in 1999. Since then, much work has been done, and numerous RFCs have been published to solidify and extend SIP capabilities.

SIP is designed to provide signaling and session management for voice and multimedia connections over packet-based networks. It is a peer-to-peer protocol with intelligent endpoints and distributed call control, such as H.323. Gateways that use SIP do not depend on a call agent, although the protocol does define several functional entities that help SIP endpoints locate each other and establish a session.

In this chapter you will learn

- How SIP works
- SIP call flow
- SIP pros and cons
- Dial plan considerations
- How to implement SIP gateways
- Some ways to secure SIP gateways
- Allowing H.323 to SIP connections
- Troubleshooting tools

Description of SIP

SIP was designed as one module in an IP communications solution. This modular design allows it to integrate with and use the services of other existing protocols, such as Session Description Protocol (SDP), Real-Time Transport Protocol (RTP), Resource Reservation Protocol (RSVP), RADIUS, and Lightweight Directory Access Protocol (LDAP). SIP usually uses User Datagram Protocol (UDP) as its transport protocol, but it can also use TCP. The default SIP port for either TCP or UDP is 5060. To provide additional security, Transport Layer Security (TLS) support is included beginning with Cisco IOS Software

Release 12.3(14)T. SIP specifications do not cover all the possible aspects of a call, as does H.323. Instead, its job is to create, modify, and terminate sessions between applications, regardless of the media type or application function. The session can range from just a two-party phone call to a multiuser, multimedia conference or an interactive gaming session. SIP does not define the *type* of session, only its management. To do this, SIP performs four basic tasks:

- Locating users, resolving their SIP address to an IP address
- Negotiating capabilities and features among all the session participants
- Changing the session parameters during the call
- Managing the setup and teardown of calls for all users in the session

SIP is built on a client-server model, using requests and responses that are similar to Internet applications. It uses the same address format as e-mail, with a unique user identifier (such as telephone number) and a domain identifier. A typical SIP address looks like one of the following:

sip:1112223344@mycompany.com
sip:1112223344@10.1.1.1

This allows Domain Name System (DNS) to be used to locate users, and it also allows SIP to integrate easily with e-mail. SIP uses Multipurpose Internet Mail Extension (MIME) to describe the contents of its messages. Thus, SIP messages can contain information other than audio, such as graphics, billing data, authentication tokens, or video. Session Description Protocol (SDP) is used to exchange session capabilities and features.

One of the most unique parts of SIP is the concept of *presence*. The public switched telephone network (PSTN) can provide basic presence information—whether a phone is on- or off- hook—when a call is initiated. However, SIP takes that further. It can provide information on the *willingness* of the other party to receive calls, not just the ability, before the call is attempted. This is similar in concept to instant messaging applications—you can choose which users appear on your list, and they can choose to display different status types, such as offline, busy, and so on. Users who subscribe to that instant messaging service know the availability of those on their list before they try to contact them. With SIP, you can gather presence information from many devices, such as cell phones, SIP phones, personal digital assistants (PDA), and applications. A SIP *Watcher* subscribes to receive presence information about a SIP *Presentity.* SIP presence information is available only to subscribers.

SIP is already influencing the marketplace. A growing number of IP Telephony Service Providers (ITSP), such as Vonage, are already using it. Traditional telephony providers, such as AT&T, have created SIP-aware networks for both internal and customer use. Cellular phone providers use SIP to offer additional services in their 3G networks. The Microsoft real-time communications platform—including instant messaging, voice, video, and application-sharing—is based on SIP. Cisco applications such as MeetingPlace,

CallManager, and CallManager Express (CME) support SIP. Some hospitals are implementing SIP to allow heart monitors and other devices to send an instant message to nurses. You can expect to see its use increase as more applications and extensions are created for SIP.

SIP Functional Components

SIP endpoints are called user agents (UA) and can be various devices, including IP phones, cell phones, PDAs, Cisco routers, or computers running a SIP-based application. UAs can act as either clients or servers. The user agent client (UAC) is the device that is initiating a call, and the user agent server (UAS) is the device that is receiving the call. The SIP protocol defines several other functional components. These functional entities can be implemented as separate devices, or the same device can perform multiple functions.

- **Proxy server**—This server can perform call routing, authentication, authorization, address resolution, and loop detection. A UA sends its call setup messages through a proxy server. The proxy server can forward the messages if it knows where the called party is located, or it can query other servers to find that information. It then forwards the request to the next hop. When it receives a response to the request, it forwards that to the client UA. After the call is set up, the proxy server can elect to stay in the signaling path so that it also sees call change or termination messages, or it can withdraw from the path and let the UAs communicate directly. Cisco has a SIP proxy server product.
- **Redirect server**—UAs and proxy servers can contact a redirect server to find the location of an endpoint. This is particularly useful in a network that has mobile users whose location changes. The redirect server can let its clients know that a user has moved either temporarily or permanently. It can also return multiple possible addresses for the user, if necessary. When a UA has multiple addresses, the proxy server can *fork* the call, sending it to each address either simultaneously or sequentially. This allows "Find Me/Follow Me" type services. Cisco routers can act as SIP redirect servers.
- **Registrar server**—UAs register their location with a registrar server, which places that information into a location database. A registrar server responds to location requests from other servers. The server can maintain the location database locally, or it can employ a separate location server. Cisco routers and CallManager 5.x can act as SIP registrar servers.
- **Location server**—This server maintains the location database for registered UAs.
- **Back-to-back user agent (B2BUA)**—This server acts as a UA server and client at the same time. It terminates the signaling from the calling UA and then initiates signaling to the called UA. B2BUAs are allowed to change the content of requests, giving them more control over the call parameters. Cisco CallManager 5.x can function as a SIP B2BUA.
- **Presence server**—This server gathers presence information from Presentities and subscription information from Watchers, and sends status notifications.

All these functions work together to accomplish the goal of establishing and managing a session between two UAs. SIP servers can also interact with other application servers to provide services, such as authentication or billing.

You can configure Cisco routers as SIP gateways. As such, they can act as a SIP UAC or UAS, they can register E.164 numbers with a SIP registrar, and they can act as SIP registrar and redirect servers. In addition, they can set up SIP trunks to another SIP gateway or to CallManager.

A Cisco SIP gateway that is using Survivable Remote Site Telephony (SRST) can provide registration and redirection services to SIP phones when CallManager and proxy servers are unavailable. SRST is not on by default; you must configure it. Both SIP and SCCP phones can fail over to a router that is running SIP SRST. Cisco CME and SRST also support B2BUA functionality beginning in Cisco IOS 12.(4)T. SIP SRST is described in Chapter 13, "SRST and MGCP Gateway Fallback."

SIP Messages

SIP uses plain-text messages, following the format of standard Internet text messages. This helps in troubleshooting, because it is easy to read SIP messages. However, you must understand the types of messages and their formats to successfully troubleshoot them. This section helps you with that understanding.

SIP messages are either requests or responses to a request; the function that the request invokes on a server is called a *method*. Several types of SIP methods exist. The original SIP specification included the following six methods. Cisco gateways can both send and receive these methods, except where noted.

- **REGISTER**—A UA client sends this message to inform a SIP server of its location.
- **INVITE**—A caller sends this message to request that another endpoint join a SIP session, such as a conference or a call. This message can also be sent during a call to change session parameters.
- **ACK**—A SIP UA can receive several responses to an INVITE. This method acknowledges the final response to the INVITE.
- **CANCEL**—This message ends a call that has not yet been fully established.
- **OPTIONS**—This message queries the capabilities of a server. Cisco gateways receive these methods only.
- **BYE**—This message ends a session or declines to take a call.

Cisco gateways also support the following additional methods, but they only respond to them. They do not generate them.

- **INFO**—This message is used when data is carried within the message body.
- **PRACK**—This message acknowledges receipt of a provisional, or informational, response to a request.

- **REFER**—This message points to another address to initiate a transfer.
- **SUBSCRIBE**—This message lets the server know that you want to be notified if a specific event happens.
- **NOTIFY**—This message lets the subscriber know that a specified event has occurred. It can also transmit dual tone multifrequency (DTMF) tones.
- **UPDATE**—A UAC uses this to change the session parameters, such as codec used or quality of service (QoS) settings, before answering the initial INVITE.

SIP entities can send additional messages in response to a method; these responses are listed in Table 4-1. Responses to SIP methods fall into six categories. The **100 Series** designates informational or provisional responses, such as 100 for Trying, and 180 for Alerting. A **200 Series** response means that the request was successful; it includes 200 for OK, and 202 for Accepted. The **300 Series** redirects the user to a different location for the called endpoint. Examples include 301 for Moved Permanently and 302 for Moved Temporarily. The **400 Series** of responses indicate a request failure, such as 404 User Not Found and 480 Temporarily Unavailable. A **500 Series** response is received due to a server failure, such as 500 for Server Internal Error or 503 for Service Unavailable. The **600 Series** is used for a global failure, including 603 when the call is declined.

Table 4-1 *SIP Response Table*

Class of Response	Status Code	Explanation
Informational/provisional	100	Trying
	180	Ringing
	181	Call Is Being Forwarded
	182	Queued
	183	Session Progress
Success	200	OK
Redirection	300	Multiple Choices
	301	Moved Permanently
	302	Moved Temporarily
	305	Use Proxy
	380	Alternative Service
Client-error	400	Bad Request
	401	Unauthorized
	402	Payment Required

continues

Table 4-1 *SIP Response Table (Continued)*

Class of Response	Status Code	Explanation
	403	Forbidden
	404	Not Found
	405	Method Not Allowed
	406	Not Acceptable
	407	Proxy Authentication Required
	408	Request Timeout
	410	Gone
	413	Request Entity Too Large
	414	Requested URL Too Large
	415	Unsupported Media Type
	416	Unsupported URI* Scheme
	420	Bad Extension
	421	Extension Required
	423	Interval Too Brief
	480	Temporarily Not Available
	481	Call Leg or Transaction Does Not Exist
	482	Loop Detected
	483	Too Many Hops
	484	Address Incomplete
	485	Ambiguous
	486	Busy Here
	487	Request Terminated
	488	Not Acceptable Here
	491	Request Pending
	493	Undecipherable
Server-error	500	Internal Server Error
	501	Not Implemented
	502	Bad Gateway
	503	Service Unavailable
	504	Server Timeout
	505	SIP Version Not Supported
	513	Message Too Large

Table 4-1 *SIP Response Table (Continued)*

Class of Response	Status Code	Explanation
Global failure	600	Busy Everywhere
	603	Decline
	604	Does Not Exist Anywhere
	606	Not Acceptable

*URI = uniform resource identifier

Example 4-1 shows a SIP INVITE message and explains the different fields. This call is from an IP phone in a CME network to an IP phone in a CallManager network. Neither phone is a SIP endpoint—the IP addresses listed are for the gateway and CallManager. A SIP trunk exists between CallManager and the gateway/CME.

Example 4-1 *SIP INVITE Message*

```
SIP-GW#debug ccsip messages
Sent:
!Request-URI (Uniform Resource Identifier) field
!This is the SIP address, or SIP URL, that the INVITE is sent to
INVITE sip:3401@10.6.2.10:5060 SIP/2.0
!Each device that handles the packet adds its IP address to the VIA field
Via: SIP/2.0/UDP  10.6.3.1:5060;branch=z9hG4bKA1798
!The calling party. A tag identifies this series of messages
From: <sip:4105553501@10.6.3.1>;tag=105741C-1D5E
!The called party
To: <sip:3401@10.6.2.10>
Date: Fri, 06 Jan 2006 05:35:01 GMT
!Unique identifier for this call
Call-ID: E937365B-2C0C11D6-802FA93D-4772A3BB@10.6.3.1
!Extensions supported include reliable provisional responses and timer refreshes
Supported: 100rel,timer
!Minimum value for session interval
Min-SE:  1800
Cisco-Guid: 3892269682-738988502-2150410557-1198695355
!Identifies the device that originated the INVITE
User-Agent: Cisco-SIPGateway/IOS-12.x
!List of methods that are supported
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY,
INFO, UPDATE, REGISTER
!Identifies call sequence number and method for this call
CSeq: 101 INVITE
!Max number of proxies or gateways that can forward this message
Max-Forwards: 70
Remote-Party-ID: <sip:4105553501@10.6.3.1>;party=calling;screen=no;privacy=off
Timestamp: 1014960901
!Identifies the user agent client, for return messages
Contact: <sip:4105553501@10.6.3.1:5060>
Expires: 180
Allow-Events: telephone-event
```

continues

Example 4-1 *SIP INVITE Message (Continued)*

```
!This INVITE carries an SDP message
Content-Type: application/sdp
Content-Length: 202
```

SIP uses SDP to exchange information about endpoint capabilities and negotiate call features. This sample INVITE contains SDP information. The SDP part of a SIP message has standard fields, as shown in Example 4-2. This is the continuation of the INVITE message in Example 4-1. The SDP fields have the following meanings:

- **v**—Tells the SDP version
- **o**—Lists the organization of the calling party
- **s**—Describes the SDP message
- **c**—Lists the IP address of the originator
- **t**—Tells the timer value
- **m**—Describes the media that the originator expects
- **a**—Gives the media attributes

Example 4-2 *SIP SDP Message Contents*

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 7181 811 IN IP4 10.6.3.1
s=SIP Call
c=IN IP4 10.6.3.1
t=0 0
m=audio 18990 RT
SIP-CME#P/AVP 0 19
c=IN IP4 10.6.3.1
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20
```

Continuing the call, the called side (the UAS) returns a provisional response 100 Trying, shown in Example 4-3. Note that the call sequence number, 101, and the method type it is responding to, INVITE, are sent in each message.

Example 4-3 *SIP "Trying" Response*

```
Received:
!"Trying" indicates that the gateway has received the INVITE
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  10.6.3.1:5060;branch=z9hG4bKA1798
From: <sip:4105553501@10.6.3.1>;tag=105741C-1D5E
!A tag is added by the UAS to identify this series of messages
To: <sip:3401@10.6.2.10>;tag=16777231
Date: Fri, 06 Jan 2006 5:35:10 GMT
Call-ID: E937365B-2C0C11D6-802FA93D-4772A3BB@10.6.3.1
```

Example 4-3 *SIP "Trying" Response (Continued)*

```
Timestamp: 1014960901
CSeq: 101 INVITE
Allow-Events: telephone-event
Content-Length: 0
```

In Example 4-4, the UAS sends a 180 Ringing response to indicate that the remote phone is ringing.

Example 4-4 *SIP Ringing Response*

```
Received:
! Ringing indicates that the called phone is being alerted
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  10.6.3.1:5060;branch=z9hG4bKA1798
From: <sip:4105553501@10.6.3.1>;tag=105741C-1D5E
To: <sip:3401@10.6.2.10>;tag=16777231
Date: Fri, 06 Jan 2006 5:35:10 GMT
Call-ID: E937365B-2C0C11D6-802FA93D-4772A3BB@10.6.3.1
Timestamp: 1014960901
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK
Allow-Events: telephone-event
Remote-Party-ID: <sip:3401@10.6.2.10>;party=called;screen=no;privacy=off
Contact: <sip:3401@10.6.2.10:5060>
Content-Length: 0
```

The remote phone has picked up the call, so a 200 OK response is sent, as shown in Example 4-5.

Example 4-5 *SIP OK Response*

```
Received:
! OK indicates that the called phone has answered
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.6.3.1:5060;branch=z9hG4bKA1798
From: <sip:4105553501@10.6.3.1>;tag=105741C-1D5E
To: <sip:3401@10.6.2.10>;tag=16777231
Date: Fri, 06 Jan 2006 5:35:12 GMT
Call-ID: E937365B-2C0C11D6-802FA93D-4772A3BB@10.6.3.1
0Timestamp: 1014960901
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK
Allow-Events: telephone-event
Remote-Party-ID: <sip:3401@10.6.2.10>;party=called;screen=yes;privacy=off
Contact: <sip:3401@10.6.2.10:5060>
Content-Type: application/sdp
Content-Length: 221

v=0
o=CiscoSystemsCCM-SIP 2000 1000 IN IP4 10.6.2.10
s=SIP Call
```

continues

Example 4-5 *SIP OK Response (Continued)*

```
c=IN IP4 10.6.2.10
t=0 0
m=audio 24580 RTP/AVP 0 101
a=sendrecv
a=rtpmap:0 PCMU/8000
a=ptime:20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

The UAC responds to the OK message with an ACK method, shown in Example 4-6. Now the call is established.

Example 4-6 *SIP ACK Message*

```
Sent:
ACK sip:3401@10.6.2.10:5060 SIP/2.0
Via: SIP/2.0/UDP  10.6.3.1:5060;branch=z9hG4bKB1C57
From: <sip:4105553501@10.6.3.1>;tag=105741C-1D5E
T0o: <sip:3401@10.6.2.10>;tag=16777231
Date: Fri, 06 Jan 2006 5:35:13 GMT
Call-ID: E937365B-2C0C11D6-802FA93D-4772A3BB@10.6.3.1
Max-Forwards: 70
CSeq: 101 ACK
Content-Length: 0
```

SIP Call Flow

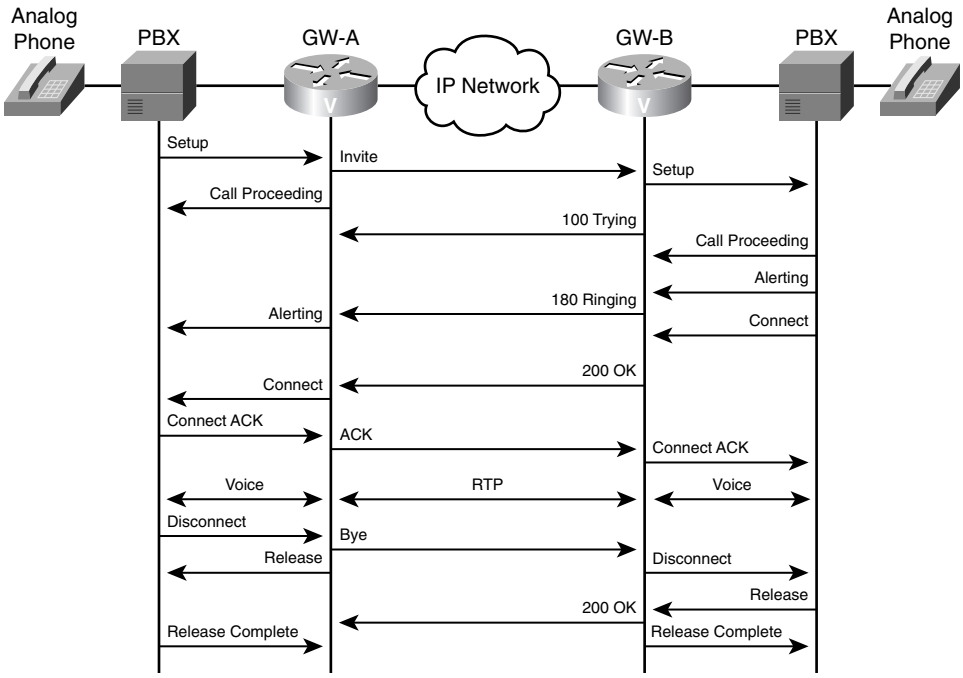
Basic SIP session setup involves a SIP UA client sending a request to the SIP URL of the called endpoint (UAS), inviting it to a session. If the UAC knows the IP address of the UAS, it can send the request. Otherwise, the UAC sends the request to a proxy or redirect server to locate the user. That server might forward the request to other servers until the user is located. After the SIP address is resolved to an IP address, the request is sent to the UAS. If the user takes the call, capabilities are negotiated and the call commences. If the user does not take the call, it can be forwarded to voice mail or another number. The following sections outline various scenarios in more detail.

Call Flow Between Two SIP Gateways

Cisco routers, including CME routers, can act as SIP gateways for calls that originate from non-SIP phones. The gateways function as SIP UAs and set up a SIP session between them for each call. Figure 4-1 shows two routers handling analog phones, using SIP between them. In this example, SIP GW-A originates the calls and acts as a UAC, and SIP GW-B acts as a UAS. The signaling from the PBX to the gateway is just normal analog call signaling. Only the two gateways exchange SIP messages.

In Figure 4-1, the analog phone on the left initiates a call to the analog phone on the right.

Figure 4-1 *Call Flow Between Two SIP Gateways*



After the first phone initiates the call, the call flow proceeds as follows:

1. The PBX sends a call setup signal to GW-A, which then sends a SIP INVITE message to GW-B. This INVITE contains SDP information for capabilities negotiation. GW-A also sends a Call Proceeding message to the PBX.
2. GW-B exchanges call setup message with its PBX and sends SIP responses 100 (Trying) and 180 (Ringing) to GW-A.
3. GW-A translates these messages into analog signaling messages for its PBX.
4. When the user on the right picks up the call, his PBX sends a Connect message to GW-B, which then forwards a SIP 200 (OK) response to GW-A. This OK response contains SDP information with the capabilities that both devices support.
5. GW-A delivers a Connect message to its PBX. When the PBX acknowledges that with a Connect ACK, it sends a SIP ACK message to GW-B.

6. GW-B sends a Connect acknowledgement to its PBX, and the call is active. At this point, normal voice steams exist between the two phones and the gateways, and RTP voice streams exist between the two gateways.
7. The user on the left hangs up the phone. His PBX sends a Call Disconnect message to GW-A. GW-A then sends a SIP BYE message to GW-B and a Release message to the PBX. The PBX responds with a Release Complete message.
8. GW-B sends a Call Disconnect message to its PBX, which responds with a Release message.
9. GW-B forwards a SIP 200 (OK) response to GW-A and a Release Complete message to its PBX. The call is now completely terminated.

Call Flow Using a Proxy Server

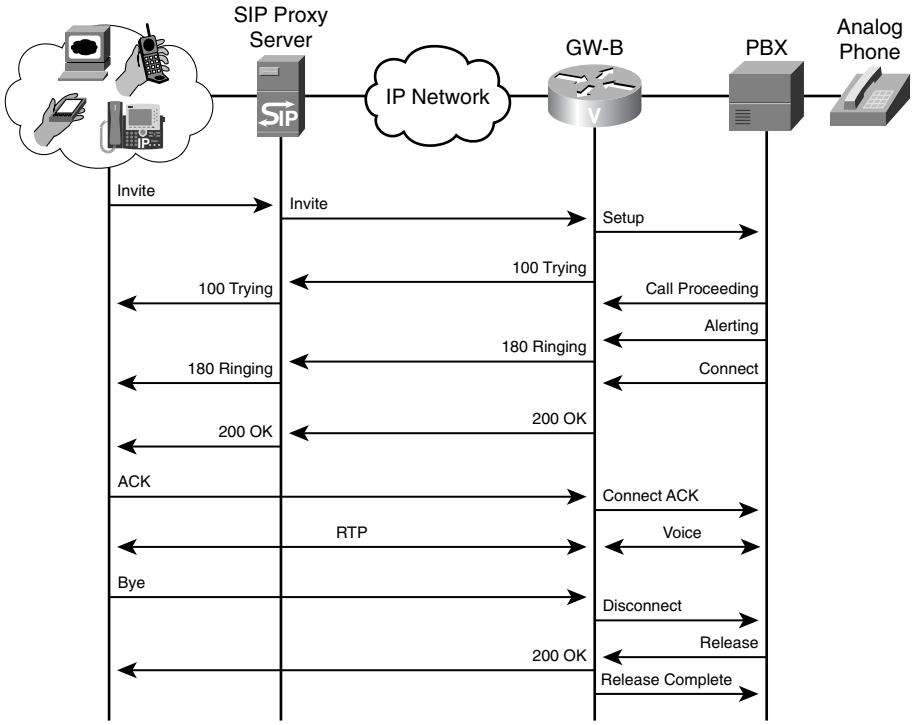
SIP UAs register with a proxy server or a registrar. Proxy servers then act as an intermediary for SIP calls. Cisco routers that are acting as SIP gateways can use the services of a SIP proxy server, either contacting the server or receiving requests from it. They can additionally register E.164 numbers with a proxy server or a registrar.

Proxy servers can either leave the signaling path when the call is connected or can enable "Record-route" to stay in the signaling path. If Record-route is disabled, the proxy server does not know of any changes to the call or when the call is disconnected. Figure 4-2 shows call flow when Record-route is disabled.

In Figure 4-2, a SIP endpoint places a call using a proxy server. The figure shows several types of endpoints:

- A PC and a PDA running a SIP application
- A SIP phone
- A cell phone that uses SIP

Figure 4-2 *SIP Call Flow Using a Proxy Server*



In Figure 4-2, one of these endpoints places a call to an analog phone behind SIP gateway GW-B. The call flow proceeds as follows:

1. The UAC sends an INVITE to its proxy server. In this INVITE, the Request-URI field contains the address of the called phone number as part of the SIP address. SDP information is included with this INVITE.
2. The proxy server creates a new INVITE, copying the information from the old INVITE, but replacing the Request-URI with the address of GW-B—the UAS.
3. When GW-B receives the INVITE, it initiates a call setup with the PBX. It sends a SIP response 100 (Trying) to the proxy server which, in this example, sends a 100 response to the SIP UAC. The proxy server is not required to send this response.
4. The PBX sets up an analog call with the end user and sends call progress messages to GW-B. When GW-B receives the Alerting message, it sends a SIP 180 (Ringing) message to the proxy server. The proxy server sends the same message to the UAC.

5 When the end user picks up the phone, the PBX sends a Connect message to GW-B. GW-B then sends a SIP 200 (OK) response to the proxy server, which sends it to the UAC. SDP information for the remote end is included in this OK response. The proxy server is not configured to be stateful—that is, Record Route is disabled. Therefore, the proxy server leaves the signaling path, and all further SIP signaling is directly between the UAC and GW-B.

6 The SIP UAC acknowledges the OK response, and a two-way RTP stream is established between the UAC and GW-B, the UAS. A two-way voice stream is established between GW-B and the PBX.

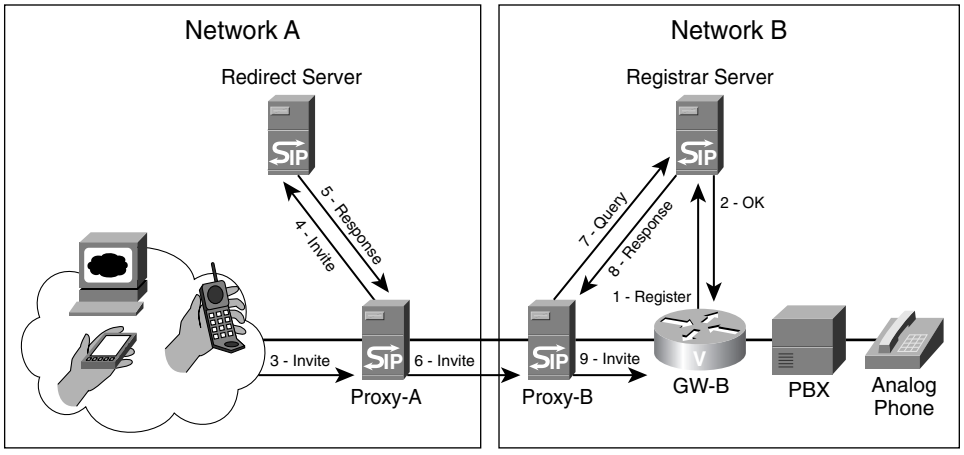
7 When the UAC hangs up, it exchanges SIP BYE and OK signals with GW-B. GW-B terminates the call with the PBX.

Call Flow Using Multiple Servers

SIP UAs and SIP proxy servers can contact a redirect server to determine where to send an INVITE. They typically do this when the called number is outside the local domain. The redirect server returns the most detailed information it has—either endpoint location(s) or the location of the next-hop server. Then it relies on the proxy server or UAC to route its INVITE appropriately.

Figure 4-3 shows the call flow in a more complex network with registrar, redirect, and proxy servers. (Recall that these are functional units and can all reside in the same device.) The figure shows the messages that are necessary to route the initial INVITE method to the UAS. After GW-B, the UAS, receives the INVITE, call flow is similar to the previous examples.

Figure 4-3 *SIP Call Flow with Multiple Servers*



In Figure 4-3, one of the SIP endpoints in Network A calls an analog phone behind gateway GW-B in Network B. The following steps take place:

1. The gateway, GW-B, registers the E.164 phone numbers of its analog phones with the registrar server.
2. The registrar server replies with a 200 (OK) response.
3. The UAC sends an INVITE method to its proxy server, Proxy-A.
4. The proxy server recognizes that the destination number is outside its domain. It sends the INVITE to the redirect server.
5. The redirect server replies with a 300-series message listing the SIP address of the next-hop proxy server, Proxy-B.
6. Proxy-A sends an INVITE message to Proxy-B.
7. Proxy-B requests the location of the called number from its registrar server.
8. The registrar server responds with the SIP address of GW-B.
9. Proxy-B sends an INVITE to GW-B.

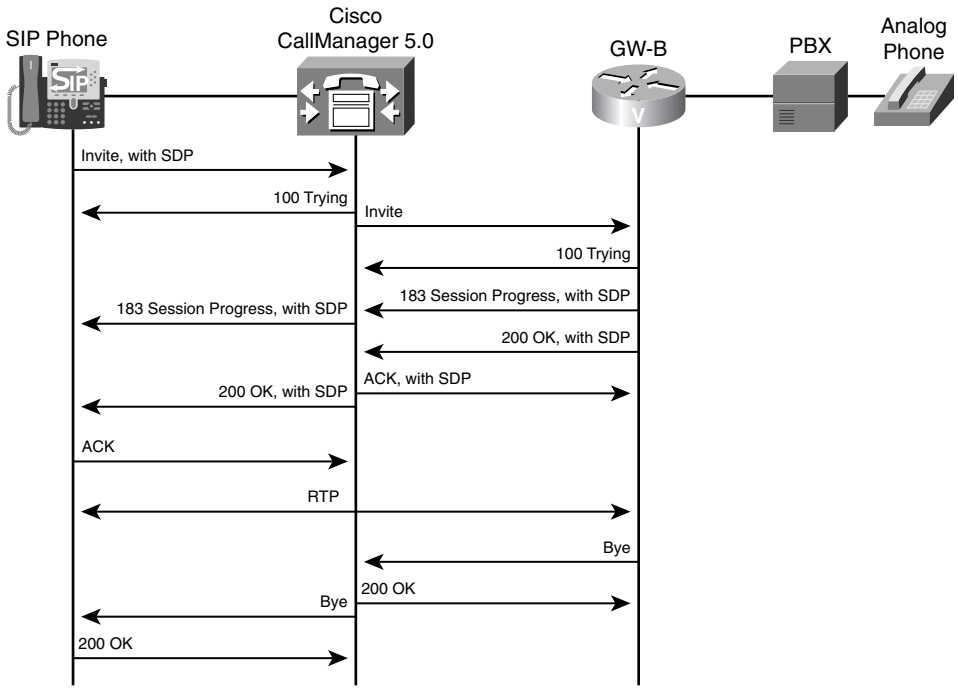
Following these steps, GW-B sets up the call with the PBX. It sends responses to Proxy-B, which forwards them through Proxy-A to the calling endpoint. If Record-route is enabled, all further signaling goes through the proxies. If not, call signaling proceeds as shown in Figure 4-2.

Call Flow Using Cisco CallManager 5.x

CallManager 5.x supports SIP phones and is an integral part of a SIP network. It can play different roles, such as registrar server and B2BUA.

Figure 4-4 illustrates a call flow scenario with CallManager acting as a B2BUA.

Figure 4-4 *Call Flow with CallManager 5.x*



In Figure 4-4, a SIP phone is registered to a CallManager. The SIP phone places a call to an analog phone off a PBX behind the router/gateway GW-B. A SIP trunk exists between CallManager and the gateway. CallManager acts as a B2BUA—it terminates each leg of the call during the signaling phases, yet it allows the RTP stream to go directly between the two endpoints. This is accomplished by the way SDP information is sent.

1. A SIP phone that is registered to CallManager calls the analog phone. It sends an INVITE containing standard SDP information to CallManager. CallManager responds with a 100 Trying message. In this step, CallManager is acting as a UAS.
2. CallManager sends an INVITE over its SIP trunk to the remote SIP gateway, GW-B. This INVITE has a different Call-ID number than the one from the phone. In addition, this INVITE does not contain SDP fields. CallManager acts as a UAC in this step.
3. GW-B answers with a 100 Trying message and initiates a call to the PBX. (That signaling is not shown.) GW-B sends its SDP parameters in a 183 Session Progress message to CallManager. Included in this are the session parameters that the gateway supports.

4. CallManager sends a 183 Session Progress message to the SIP phone. This message contains an SDP portion with the capabilities that both endpoints support. For instance, suppose that the original SDP message of the phone indicated that it supported G.711 and G.729 codecs, but the gateway SDP message said that it supported only G.729. In that case, the 183 message from CallManager to the phone would list only G.729. It would also list the IP address of GW-B as the originator address in SDP field 'c.'
5. When the analog phone picks up, GW-B sends a 200 OK message containing its SDP information. CallManager acknowledges it with an ACK that contains the SDP information that both endpoints support. The IP address of the SIP phone is also included as the originator address in the SDP field 'c.'
6. CallManager sends a 200 OK message with SDP information to the phone. The phone acknowledges that message. Now that each endpoint has the IP address of the other, the two can establish an RTP stream between them for the duration of the call.
7. In Figure 4-4, the analog phone hangs up, so GW-B sends a SIP BYE method to CallManager.
8. CallManager replies with a 200 OK response and then sends a BYE to the SIP phone. The phone responds with a 200 OK message.

SIP Pros and Cons

People have high expectations for SIP. They hope it will enable advanced, anywhere, anytime multimedia communication. However, SIP is basically just another call control protocol, with its pros and cons. The following sections list a few of each.

Pros

- SIP works independently of the type of session, or the media used, giving it flexibility.
- It is an open standard, allowing multivendor support and integration. Applications can be written to customize SIP uses.
- SIP messages are clear text, making troubleshooting easier.
- SIP can accommodate multiple users with differing capabilities. For instance, in a conference that has some users with video capability and some only with audio capability, the video users can see each other. They do not have to drop down to audio only, as with other protocols.

Cons

- Processing text messages puts a higher load on gateways. The router must translate that text into a language that the router can understand. Code for this must be in the Cisco IOS.
- SIP is a fairly new protocol, so fewer people understand it than the older protocols. Be sure you have trained support personnel if you intend to implement SIP within your network.
- When you are using both SIP and SCCP phones on the same network, you must convert between in-band and out-of-band DTMF tones.
- SIP features are still being developed, and many vendors have proprietary implementations of the protocol.

When to Use SIP

SIP is the protocol of choice when you want to integrate multiple types of media. For example, perhaps you get an instant message on your phone screen that you want to respond to verbally. You start to press the button to call the IM sender, but you see that the sender has set his status to "Don't call me," so you refrain. Later, you would like to have a videoconference with three other people, but you notice that one of them has his phone off-hook. The other two join you, one as video and the other as audio only. You send a text message to the IP phone of the third person, asking him to join when he can. This might sound like the office of the future, but it is becoming reality.

You can use SIP trunks even without native SIP phones. You might choose them because of the simple session setup mechanism for SIP. For instance, you can use a SIP trunk between gateways with toll bypass. When you use SIP with CallManager or CME, you are not limited to Cisco-brand IP phones. You can also use other commercially available, and perhaps lower-priced, SIP phones. Be aware that CallManager might not be able to provide all the features for these phones that it can for a Cisco-brand SIP phone.

Before you implement a SIP network, plan its integration into your existing network. Determine exactly what you want to accomplish and how you will accomplish it. Remember also that you need to either find people who are knowledgeable about SIP or train your staff.

Dial Plan Considerations

SIP dial plan considerations are similar to those for an H.323 gateway, because dial peers control gateway call routing. You must configure dial peers to forward calls out of the gateway. You can forward calls to CallManager using a Voice over Internet Protocol (VoIP) dial peer, to the PSTN as a plain old telephone service (POTS) dial peer, to another gateway using a VoIP dial peer, or to directly connected voice ports as POTS dial peers.

In a voice network that has CallManagers and SIP gateways, it is important to understand the interaction between the two, because different versions of CallManager have different SIP capabilities. CallManager versions before 5.x can only have a SIP trunk to a gateway or other servers. CallManager 5.x and above also acts as a SIP B2BUA and allows SIP phones to register to it. It also can do domain routing for SIP calls. A new menu, SIP Route Pattern, allows you to configure SIP URI dialing. Therefore, your dial plan must take into account the CallManager version. No matter which version of CallManager you use, you configure a dial plan to send calls to the SIP trunk when needed. CallManager appears to the SIP gateway as a SIP-enabled VoIP dial peer.

Another consideration in SIP networks is where the dial plan will reside. The default behavior of SIP is to push down the dial plan to each endpoint. When a user dials digits on the phone, the phone compares those numbers against its internal dial plan. If the phone finds a match, it sends an INVITE. Otherwise, it must wait for the interdigit timer to expire before playing a reorder tone. The alternative is to use the Key Press Markup Language (KPML). When you use KPML, the SIP phone sends each digit to CallManager, similar to the way SCCP phones behave. CallManager can instruct the phone to play a reorder tone immediately if an incorrect number is dialed, or it can route the call as soon as enough digits are dialed. If you do not use KPML, you must configure SIP dial rules.

In your dial planning, consider the need to configure the gateway for such options as number translations or other digit manipulations, or call restrictions. If you are using SRST, be sure that the dial plan will work both with and without CallManager and, if possible, any SIP servers in the network. You need at least one dial peer with a destination pattern for routing outgoing calls. Default incoming POTS and VoIP dial peers are available, but you should specifically configure dial peers for incoming calls if you need a nondefault configuration.

As with H.323, SIP gateway configuration can become complex in a large network. You must configure each gateway with the information you need to route calls. Proxy, registrar, redirect, and DNS servers can help the network scale by providing dial plan resolution. This simplifies the gateway configuration.

Implementing SIP Gateways

Configuring a SIP gateway can be as simple as configuring SIP VoIP dial peers or as complex as tweaking SIP settings and timers. Gateway SIP configuration is done in three basic places: on dial peers, under SIP UA configuration mode, and under voice service VoIP configuration mode. The following sections discuss the types of configuration you can accomplish in each mode.

SIP Dial Peer Configuration

A basic SIP gateway configuration consists of simply adding one line under a VoIP dial peer configuration: **session protocol sipv2**. You can use additional dial peer settings; the exact options and commands vary by Cisco IOS version. For instance, to allow SIP calls to be hairpinned from one VoIP dial peer to the other, use the **redirect ip2ip** command. SIP supports both consultative and blind call transfers from Cisco gateways. It also supports call forwarding from IP phones that are registered with the gateway as e-phones. This capability is enabled per dial peer with the **application session** command in dial-peer configuration mode.

To change the transport protocol (UDP is the default), use the **session transport** [**udp** | **tcp**] command. SIP can switch from using UDP to TCP when a voice packet gets within 200 bytes of the maximum transmission unit (MTU) to avoid UDP fragmentation. To configure this for a specific dial peer, use the **voice-class sip transport switch udp tcp** command. You can also configure this globally under voice service configuration mode.

Example 4-7 shows the configuration of two SIP dial peers. On the first dial peer, the transport protocol is changed from the default UDP to TCP (an optional step). On the second dial peer, the configuration shows enabling switching from UDP to TCP for large packets.

Example 4-7 *Configuring a SIP Dial Peer*

```
SIP-GW(config)#dial-peer voice 3401 voip
SIP-GW(config-dial-peer)#session target ipv4:10.6.2.1
SIP-GW(config-dial-peer)#session protocol sipv2
SIP-GW(config-dial-peer)#session transport tcp
!
SIP-GW(config)#dial-peer voice 4404 voip
SIP-GW(config-dial-peer)#session target ipv4:10.7.1.1
SIP-GW(config-dial-peer)#session protocol sipv2
SIP-GW(config-dial-peer)#voice-class sip transport switch udp tcp
SIP-GW(config-dial-peer)#destination-pattern 4404...
```

SIP UA Configuration

The SIP UA does not require configuration to function, but you might want to make some adjustments. Enter UA configuration mode by issuing the **sip-ua** command. As with dial peers, the options vary by Cisco IOS and device. Table 4-2 shows some common UA commands.

Table 4-2 *SIP UA Commands*

Command	Description
calling-info	Controls the calling ID treatment for calls to and from the PSTN.
max-forwards *number*	Configures the maximum hops for SIP methods. Values are 1–70; default is 70.

Table 4-2 *SIP UA Commands (Continued)*

Command	Description
nat	Configures NAT traversal settings.
retry	Controls the number of times that a SIP message will be sent. Options include the following: • **bye**—Default is 10. • **cancel**—Default is 10. • **comet**—Default is 10. • **invite**—Default is 6. • **notify**—Default is 10. • **prack**—Default is 10. • **refer**—Default is 10. • **register**—Default is 10. • **rel1xx**—Default is 6. • **response**—Default is 6. • **subscribe**—Default is 10.
registrar {**dns:***address* | **ipv4:***destination-address*} [**expires** *seconds*][**tcp**] [**secondary**]	Allows a gateway to register the E.164 numbers of non-SIP phones with a registrar or proxy server.
sip-server {**dns:**[*host-name*] | **ipv4:***ip-addr* [**:***port-num*]}	Specifies the name or IP address of a SIP server, usually a proxy server. When you use this, you can configure the dial-peer session target as **session target sip-server**.
timers	Changes SIP signaling timers. Options include the following: • **connect**—Time to wait for a 200 response to an ACK. Default is 500 ms. • **disconnect**—Time to wait for a 200 response to a BYE. Default is 500 ms. • **expires**—Valid time for an INVITE. Default is 180,000 ms. • **register**—Time that UA waits before sending REGISTER message. Default is 500 ms. • **trying**—Time to wait for a 100 response to an INVITE. Default is 500 ms.
[**no**] **transport** {**udp** | **tcp**}	Configures the SIP UA to listen for messages on port 5060 of either UDP or TCP. Both are enabled by default.

Example 4-8 shows a SIP UA configuration. The gateway is configured to register its analog phones with redundant servers, the IP address of the proxy server is specified, the maximum number of hops for SIP methods is reduced to 10, and the gateway is limited to listening for TCP SIP messages. The configuration is partially verified by using the **show sip-ua status** command. The configured E.164 phone number registration is verified with the **show sip-ua register status** command.

Example 4-8 *SIP UA Configuration*

```
SIP-GW(config)#sip-ua
SIP-GW(config-sip-ua)#registrar ipv4:10.30.25.250 tcp
SIP-GW(config-sip-ua)#registrar ipv4:10.30.25.251 tcp secondary
SIP-GW(config-sip-ua)#sip-server ipv4:10.30.25.252
SIP-GW(config-sip-ua)#max-forwards 10
SIP-GW(config-sip-ua)#no transport udp
!
SIP-GW#show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : DISABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 10
SIP DNS SRV version: 2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP: NONE
Check media source packets: DISABLED
Maximum duration for a telephone-event in NOTIFYs: 1000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
Reason Header will override Response/Request Codes: DISABLED

SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
!
SIP-GW#show sip-ua register status
Line          peer           expires(sec)  registered
============  =============  ============  ===========
4101          20001            118           yes
4102          20003            118           yes
4103          20005            118           yes
4104          20007            118           yes
```

SIP Voice Service Configuration

The voice service configuration mode is used for voice-related commands that affect the entire gateway. Enter this mode by issuing the **voice service voip** command. In this mode, you can allow hairpinned calls for all dial peers with the **redirect ip2ip** command. (You can also issue this command under dial-peer configuration mode. Then it affects only those dial peers.)

You can enter the SIP submode with the **sip** command from voice service mode. You can do several SIP-specific configurations from this mode. Table 4-3 lists some of these; specific options vary by Cisco IOS version and device.

Table 4-3 *SIP Voice Service Configuration Commands*

Command	Description
bind {**all** | **control** | **media**} **source-interface** *interface-id*	Sets the source IP address for control signaling or media, or both.
call service stop [**forced**]	Stops the SIP service. Active calls are not affected unless the **forced** option is used.
min-se *seconds*	Changes the SIP session expiration timer. Default (and recommended) value is 1800 sec. In the event of a timer mismatch, the higher value is used.
redirect contact order [**best-match** | **longest-match**]	Sends a SIP 300 Redirect message listing all the routes in the Contact header when the gateway knows of multiple routes to a destination. This command controls the order in which the routes are listed. **Longest-match** is default; it puts routes in order of number of digits matched in its destination patterns. **Best-match** uses the current system configuration.
registrar server [**expires** [**max** *seconds*] [**min** *seconds*]]	Configures the gateway to act as a registrar server; used when SRST is active.
session transport {**tcp** | **udp**}	Globally configures either TCP or UDP as the transport protocol.
subscription maximum {**accept** | **originate**} *number*	Controls the maximum number of subscriptions accepted or originated by the gateway. Default is twice the number of dial peers configured on the gateway.
transport switch udp tcp	Enables the router to use TCP instead of UDP when a SIP message exceeds the MTU size.

Example 4-9 shows a gateway that has been configured to hair-pin calls for all dial peers. The source IP address for all SIP signaling traffic has been set to Loopback 0, and this

gateway has been configured to act as a registrar server. You can verify the interface binding with the **show sip-ua status** command.

Example 4-9 *SIP Voice Service Configuration*

```
SIP-GW(config)#voice service voip
SIP-GW(conf-voi-serv)#redirect ip2ip
SIP-GW(conf-voi-serv)#sip
SIP-GW(conf-serv-sip)#bind control source-interface lo0
SIP-GW(conf-serv-sip)#registrar server expires max 1500 min 500
```

Toll Bypass

You use toll bypass to send calls between different sites over a packet network rather than over the PSTN. Because this bypasses the PSTN, it also bypasses any long-distance toll charges. Cisco routers, functioning as edge gateways, can use SIP to pass voice traffic between them. This traffic is typically from analog phones, such as those connected to a PBX, but it can be from IP or SIP phones. Figure 4-1 shows this type of setup.

Configuring the routers for toll bypass involves two components. First, you must configure connection(s) to the internal voice network. This might be a PRI to a PBX, for instance. You must configure both the physical links and the appropriate dial peers. Second, you must configure the connection to the other router. This involves configuring the physical link and at least one SIP VoIP dial peer pointing to the remote SIP router. You configure the dial peers and gateways as detailed in the previous sections of this chapter.

Registering with CallManager

CallManager versions 4.x can register a SIP trunk connecting to a remote gateway, a proxy server, or CallManager Express. The trunk is referred to as a *signaling interface*. CallManager 5.x can register SIP phones, in addition to SIP trunks. Trunks can point to other Cisco CallManager clusters, also.

Configuring a SIP Trunk with CallManager 4.x

The following steps show how to create a SIP trunk on CallManager 4.x. The menus on other versions of CallManager might vary slightly, but the process is similar.

Step 1 To add a new trunk, select the Device menu, and then select the **Trunk** link. At the Find and List Trunks screen, shown in Figure 4-5, click the link for **Add a New Trunk**.

Step 2 On the next screen, shown in Figure 4-6, select **SIP Trunk** as the Trunk type. Notice that Device Protocol is automatically set to SIP. Click the **Next** button.

Figure 4-5 *CallManager Find and List Trunks Screen*

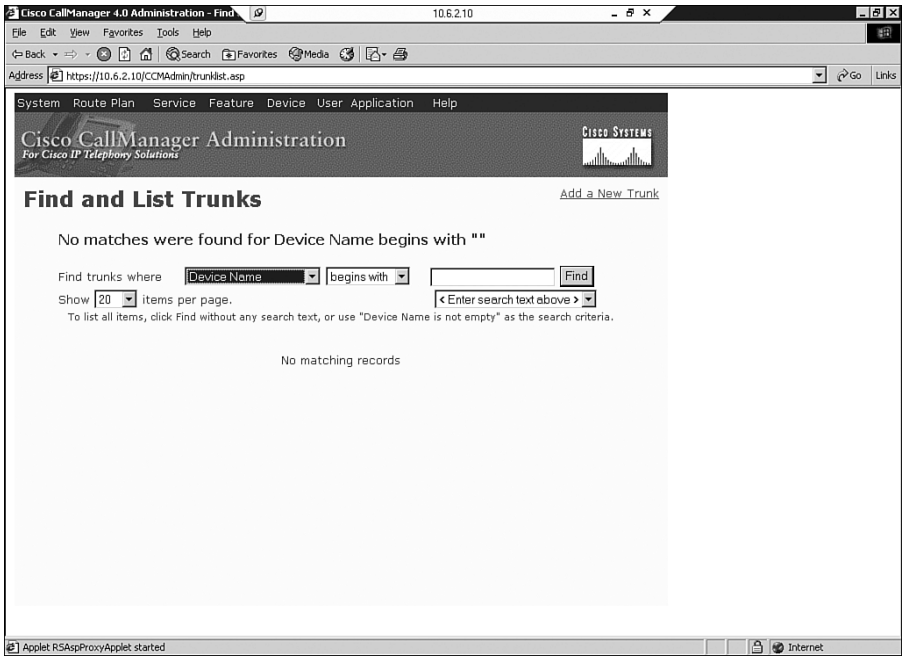
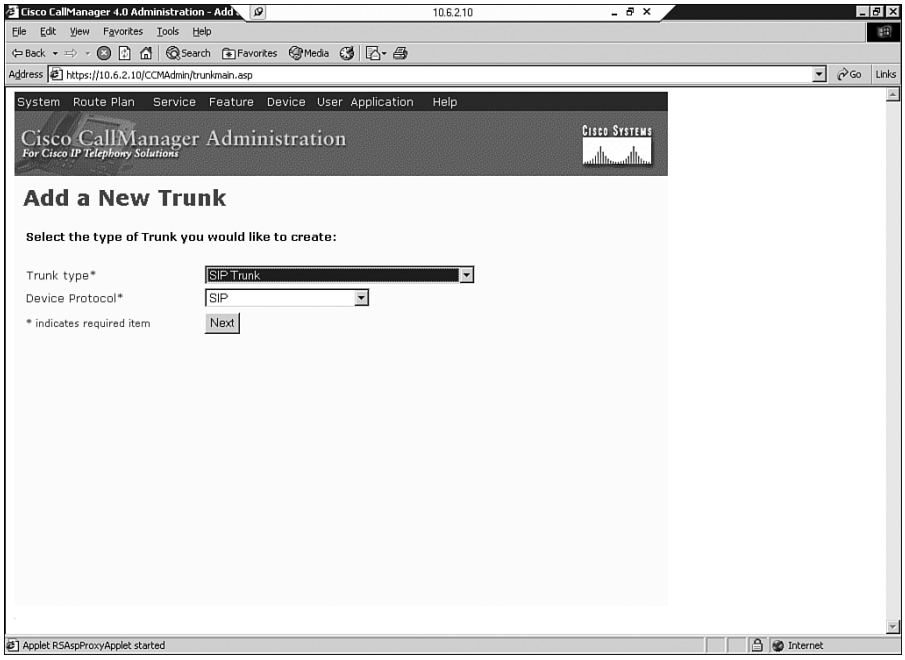
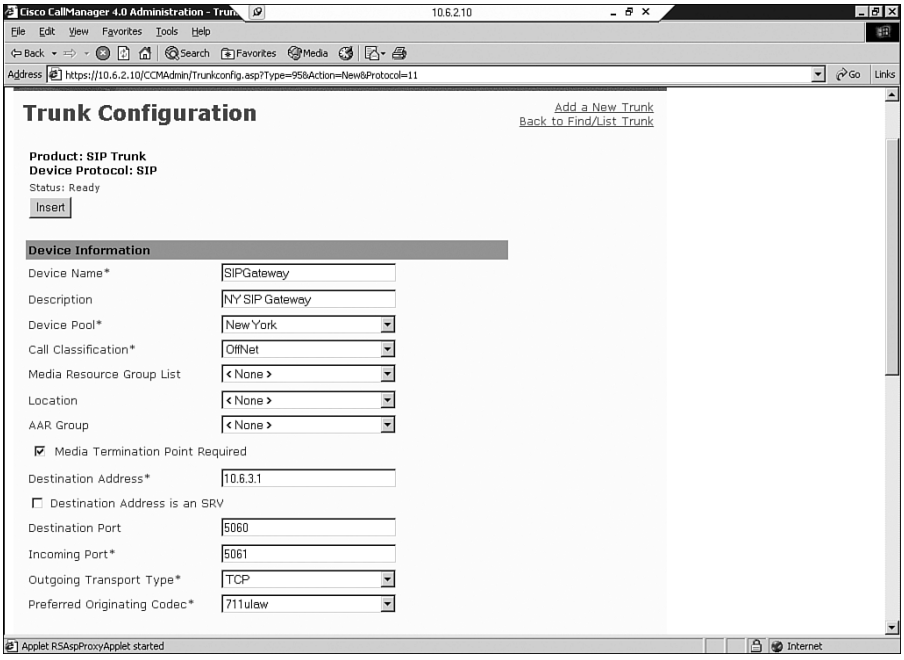


Figure 4-6 *CallManager SIP Trunk Creation*



Step 3 The Trunk Configuration screen, shown in Figure 4-7, appears next. Enter the name of the device at the other end of the trunk, and optionally a description. You must select a Device Pool. Then designate whether calls to this trunk are OnNet or OffNet, or let the system decide. Note that a media termination point (MTP) is required. This is to translate DTMF signals between SIP and SCCP phones. For more information, see the later section, "DTMF Relay."

Figure 4-7 *SIP Trunk Configuration in CallManager*



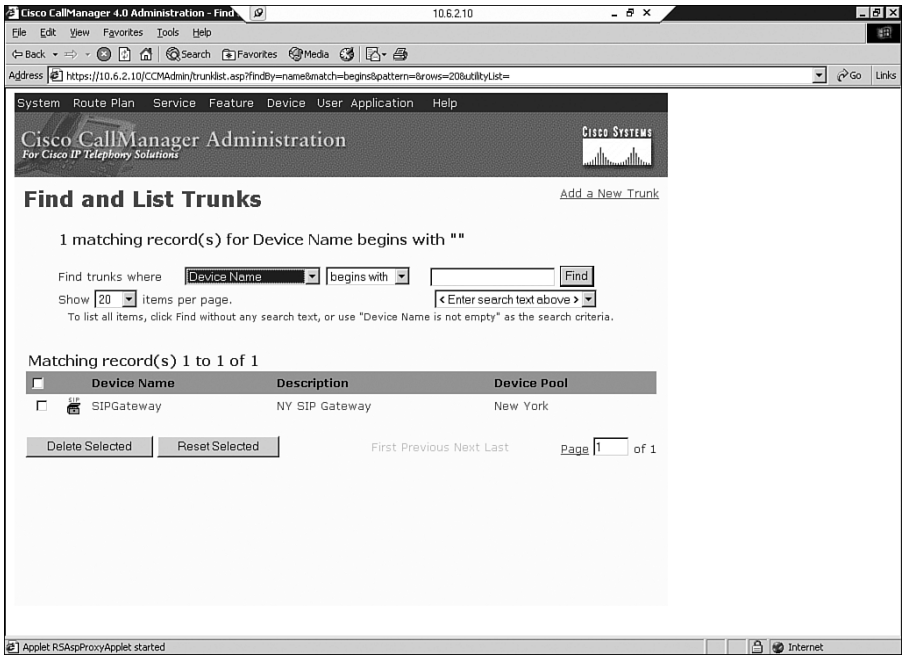
Step 4 On this same screen, enter the destination address of the trunk.

Step 5 The next options in the Trunk Configuration screen are for the Destination Port number, the Incoming Port number, and the Outgoing Transport Type (Layer 4 protocol). The default destination port for SIP is 5060, and it is typically left at that. CallManager can have multiple SIP signaling interfaces. You must use a unique incoming port for each signaling interface in CCM 4.x. Not shown in Figure 4-7 are settings for incoming and outgoing calls, further down on the screen.

Step 6 When you are finished with the configuration, click **Insert**. You are then prompted to reset the trunk.

Step 7 You can verify your trunk configuration by going back to the Find and List Trunks page. Click the **Find** button, and verify that your new trunk is there, as shown in Figure 4-8. After you have created the trunk, you can assign route patterns, list, or groups to it as normal.

Figure 4-8 *Verifying a SIP Trunk in CallManager*



Configuring a SIP Trunk with CallManager 5.x

The process that you use to register a SIP trunk with CallManager 5.x is slightly different from the CallManager 4.x process. For instance, an MTP is not required with CallManager 5.x if the endpoints can negotiate using out-of-band DTMF relay. Multiple trunks can have the same incoming port number.

To begin adding a SIP trunk in CCM 5, follow Steps 1 and 2 in the CallManager 4.x process. The next screen, the Trunk Configuration screen, has some different fields, and some fields are in different positions in CallManager 5.x. Figures 4-9, 4-10, and 4-11 show this Trunk Configuration screen. Some items to note are that the MTP might not be required and is not checked automatically. Some SIP-specific fields have been added, such as Presence Group, SIP Trunk Security Profile, SIP Profile, and SUBSCRIBE Calling Search Space.

Figure 4-9 *Configuring a SIP Trunk in CCM 5*

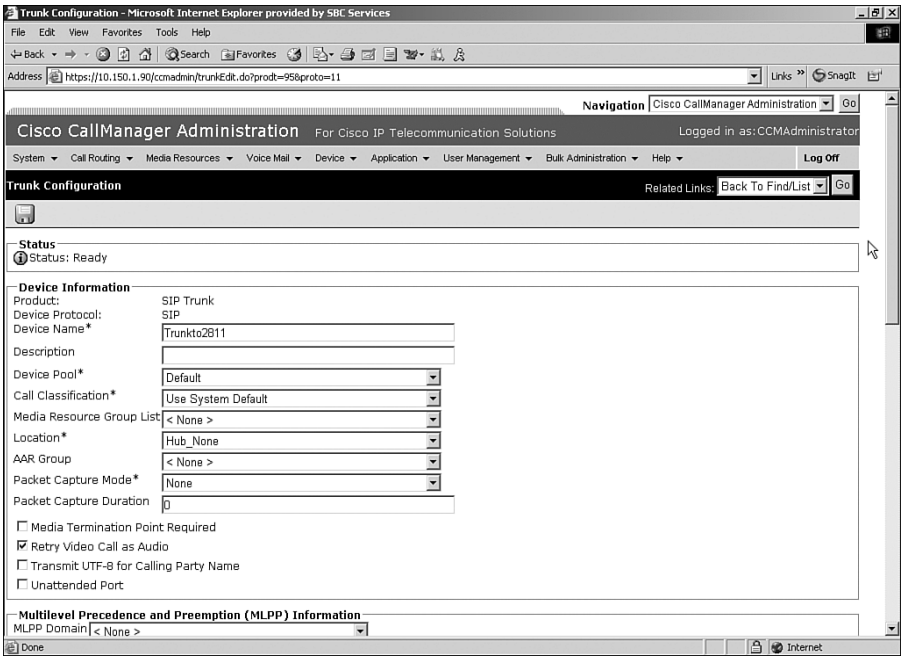


Figure 4-10 *Configuring a SIP Trunk in CCM 5 (continued)*

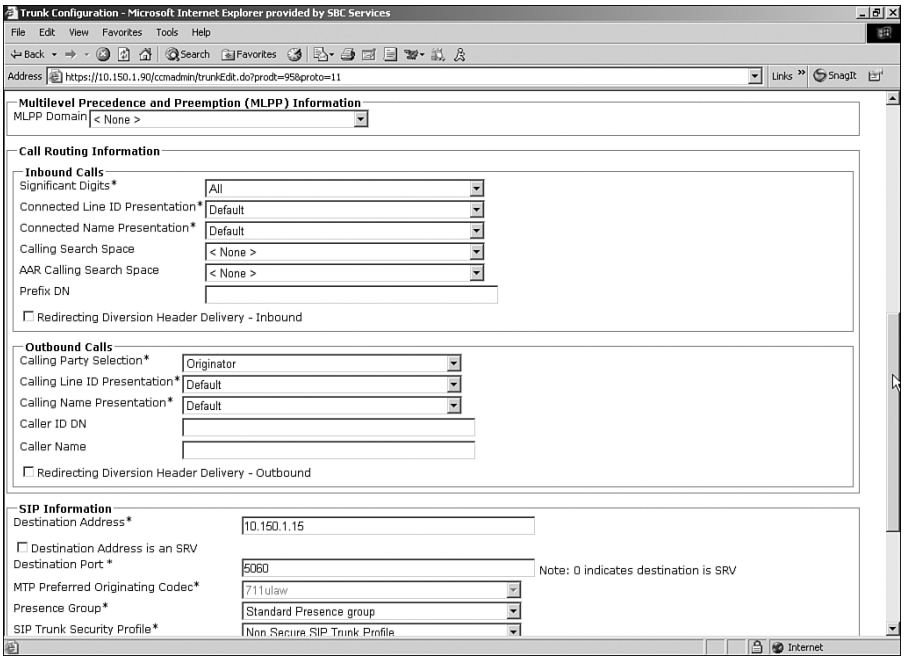
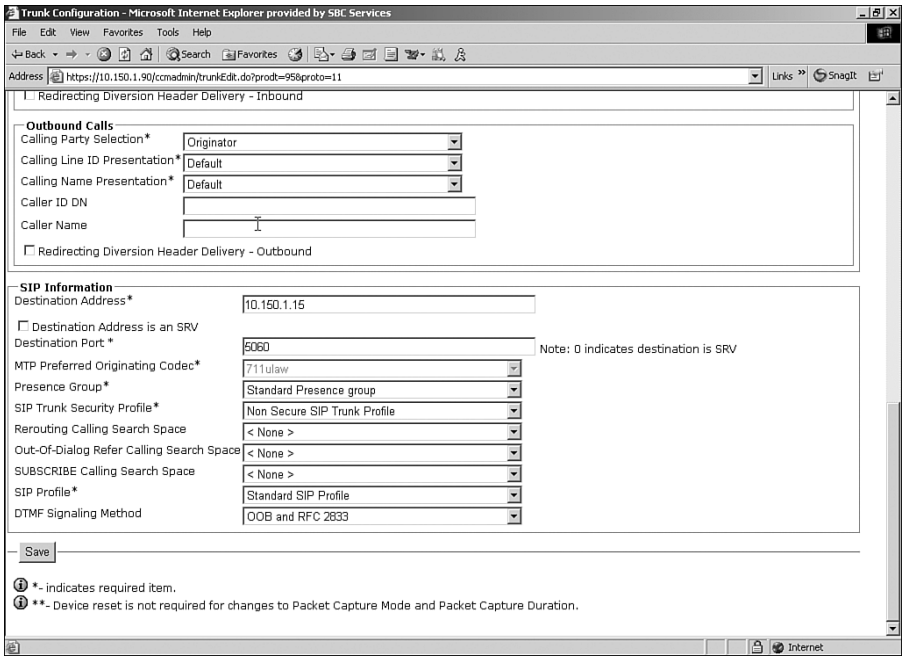


Figure 4-11 *Configuring a SIP Trunk in CCM 5 (continued)*



When you are finished configuring the trunk, you must save it by clicking the Save button on the bottom of the screen or the disk icon at the top of the screen. You can verify SIP trunk configuration for CallManager 5.x by using the same method that you used for CallManager 4.x—by using the Find button at the Find and List Trunks page.

Configuring the Gateway for a CallManager Trunk

Regardless of the CallManager version you use, the gateway sees it as a SIP dial peer. For redundancy, you can configure a dial peer for each CCM in the cluster, using preference to determine their order of use. Example 4-7 (shown previously) demonstrates the configuration for a basic SIP dial peer.

DTMF Relay

DTMF tones are the tones that are generated when a telephone key is pressed on a touchtone phone. Sometimes the called endpoint needs to hear those tones, such as when you enter digits during the call in response to a menu. Low-bandwidth codecs can distort the sound, however. *DTMF relay* allows that tone information to be reliably passed from one endpoint to the other. By default, SIP uses in-band signaling, sending the DTMF

information in the voice stream. However, you can configure it to use RTP-NTE, SIP INFO messages, SIP NOTIFY messages, or KPML for transmitting DTMF tone information.

RTP-NTE is an in-band DTMF relay method, which uses RTP Named Telephony Event (NTE) packets to carry DTMF information instead of voice. If RTP-NTE is configured, SDP is used to negotiate the payload type value for NTE packets and the events that will be sent using NTE.

RTP-NTE can cause problems communicating with SCCP phones, which use only out-of-band DTMF relay. In a CallManager 4.x network with SCCP phones, you must provision an MTP for calls that traverse the SIP trunk. This MTP translates between in-band and out-of-band DTMF signals. You must configure a separate MTP for each side of the SIP trunk. You can do this MTP in hardware, or in software on CallManager.

Cisco has two out-of-band procedures for DTMF relay. One uses SIP INFO methods, and the other uses SIP NOTIFY methods. The SIP INFO method sends DTMF digits in INFO messages. It is always enabled. When a gateway receives an INFO message containing DTMF relay information, it sends the corresponding tone.

NOTIFY-based out-of-band DTMF relay is negotiated by including a Call-Info field in the SIP INVITE and response messages. This field indicates an ability to use NOTIFY for DTMF tones and the duration of each tone in milliseconds. Using this method can help SIP gateways interoperate with Skinny phones. You can also use it for analog phones that are connected to Foreign Exchange Station (FXS) ports on the gateway.

When a DTMF tone is generated, the gateway sends a NOTIFY message to the terminating gateway. When that gateways receives the NOTIFY, it responds with SIP 200 OK and plays the DTMF tone. To configure the DTMF relay type, use the **dtmf-relay** command in dial-peer configuration mode. To optionally configure the interval between NOTIFY messages for a single DTMF event, use the **notify telephone-event max-duration** *millisecond*s command in SIP UA configuration mode. The default is 2000 msec; the lowest value between two SIP peers is the one chosen.

Example 4-10 shows these commands. Notice that two types of DTMF relay are configured. The router prefers SIP-NOTIFY but uses RTP-NTE if the other side does not support SIP-NOTIFY. If no DTMF relay method is configured, the tones are sent in-band.

Example 4-10 *Configuring NOTIFY-Based DTMF Relay*

```
!Setting the NOTIFY interval
SIP-GW(config)#sip-ua
SIP-GW(config-sip-ua)#notify telephone-event max-duration 1000
SIP-GW(config-sip-ua)#exit
!
!Setting NOTIFY-based out-of-band DTMF relay
SIP-GW(config)#dial-peer voice 3400 voip
SIP-GW(config-dial-peer)#dtmf-relay ?
  cisco-rtp          Cisco Proprietary RTP
  h245-alphanumeric  DTMF Relay via H245 Alphanumeric IE
```

Example 4-10 *Configuring NOTIFY-Based DTMF Relay (Continued)*

```
   h245-signal        DTMF Relay via H245 Signal IE
   rtp-nte            RTP Named Telephone Event RFC 2833
   sip-notify         DTMF Relay via SIP NOTIFY messages

SIP-GW(config-dial-peer)#dtmf-relay sip-notify rtp-nte
```

KPML is another way for SIP phones to send dialed-digit information, as described previously in the "Dial Plan Considerations" section. Like SIP-NOTIFY, KPML uses a NOTIFY message to transmit each digit.

Securing SIP Gateways

Your SIP gateway, as part of your IP network, should conform to your company security policy. Deployment of basic items, such as user control and authentication, access-control lists, and physical security, should be standard. The SIP network, like most of your user devices, should be on a LAN using private IP addresses, with strong perimeter security.

Because SIP messages contain IP addresses in several different locations, it is important to use a firewall that supports SIP. Cisco IOS firewalls, PIX firewalls, and Adaptive Security Appliance (ASA) devices are all able to inspect the SIP application data and maintain call flow information.

SIP supports some authentication, authorization, and accounting (AAA) mechanisms to help authenticate communications between UAs, servers, and gateways. You can use RADIUS to preauthenticate calls. The gateway forwards incoming call information to a RADIUS server, which must authenticate it before connecting the call. To enable AAA for SIP calls, you must use the normal AAA configuration on the gateway and the RADIUS server. In addition, at global configuration mode, issue the **aaa preauth** command to enter AAA preauthentication configuration mode. Specify the RADIUS server with the command **group** {**radius** | *groupname*}.

You can also use HTTP Authentication Digest. UAs, proxy servers, and redirect servers can request authentication before they process a SIP message. Gateways can respond to authentication challenges and can respond on behalf of non-SIP phones that they have registered to a SIP server. SIP defines authentication and authorization fields that can be present in the message header. A server that receives a message—such as an INVITE—without authentication credentials issues a challenge. The response includes an authorization field with an MD5 hash and other credentials. To configure a gateway to use HTTP Authentication Digest, give the following command in each dial peer or SIP-UA configuration mode: **authentication username** *username* **password** *password* [**realm** *realm*]. *Username* is the name of the user that will be authenticating, *password* is the shared password, and *realm* is an optional entry that lets you configure multiple

username/password combinations. The realm is included in the challenge, so the response will include credentials for that specific realm.

To provide a more secure, encrypted transport mechanism for SIP messages, Cisco IPT devices have added support for the TLS protocol.

Allowing H.323 to SIP Connections

SIP and H.323 dial peers can be configured on the same gateway, but call routing between the two types of dial peers is disabled by default. To enable this routing, enter voice service configuration mode and issue the command **allow-connections** *from-type* **to** *to-type*. Options for both the *from-type* and the *to-type* are **h323** and **sip**. Example 4-11 shows a router that is configured to allow multiple types of VoIP connections.

Example 4-11 *Configuring H.323 to SIP Connections*

```
H323-SIP-GW(config)#voice service voip
H323-SIP-GW (conf-voi-serv)# allow-connections h323 to h323
H323-SIP-GW (conf-voi-serv)# allow-connections h323 to sip
H323-SIP-GW (conf-voi-serv)# allow-connections sip to h323
H323-SIP-GW (conf-voi-serv)# allow-connections sip to sip
```

Troubleshooting Tools

If calls cannot be made between SIP gateways or over SIP trunks, dial peer configuration is one of the first places to check. Make sure that the dial peer is configured to use SIP and that both devices are using the same transport protocol and DTMF relay method. Make sure that destination patterns and session targets are correct, also.

Several **show** commands can troubleshoot and monitor the SIP UA function of the gateway. Example 4-12 lists them; options can vary by Cisco IOS and device.

Example 4-12 **show sip-ua** *Command Options*

```
SIP-GW#show sip-ua ?
  calls        Display Active SIP Calls
  connections  Display SIP Connections
  map          Display SIP status code to PSTN cause mapping table & vice versa
  min-se       Display Min-SE value
  mwi          Display SIP MWI server info
  register     Display SIP Register status
  retry        Display SIP Protocol Retry Counts
  service      Display SIP submode Shutdown status
  statistics   Display SIP UA Statistics
  status       Display SIP UA Listener Status
  timers       Display SIP Protocol Timers
```

The **show sip-ua connections** {**udp|tcp**} command gives you information on active connections, including those with errors. To stop a problem connection, use the **clear sip-ua** {**udp | tcp**} [**connection id** *number*] **[target ipv4:***ip-address***]** command.

To ensure that the SIP is enabled on the gateway, use the **show sip-ua service** command. You should get the following result:

```
SIP-GW#show sip-ua service
SIP Service is up
```

The **show sip-ua statistics** command provides statistics on each type of method and response, errors, and total SIP traffic information. You can reset these counters with the **clear sip-ua statistics** command.

The **show sip-ua status** command can be useful in troubleshooting, also. Output from this command was shown previously in Example 4-13.

To debug SIP messages, use the **debug ccsip** command. This command has several options, as Example 4-13 shows. Use **messages** to see the SIP method and response messages, as shown previously in Example 4-1. The **media** option shows RTP information. Your options might vary by Cisco IOS and device.

Example 4-13 debug ccsip *Command Options*

```
SIP-GW#debug ccsip ?
  all        Enable all SIP debugging traces
  calls      Enable CCSIP SPI calls debugging trace
  error      Enable SIP error debugging trace
  events     Enable SIP events debugging trace
  info       Enable SIP info debugging trace
  media      Enable SIP media debugging trace
  messages   Enable CCSIP SPI messages debugging trace
  preauth    Enable SIP preauth debugging traces
  states     Enable CCSIP SPI states debugging trace
  transport  Enable SIP transport debugging traces
```

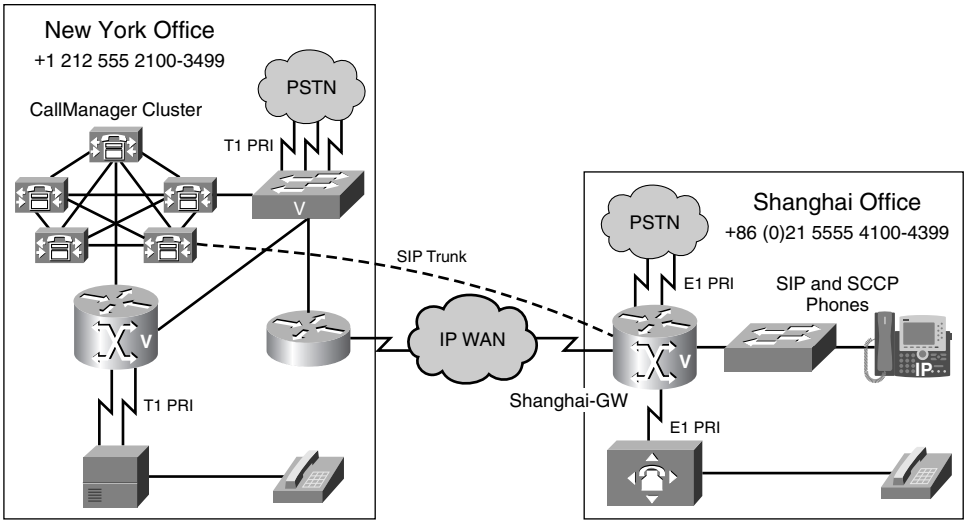
Case Study: Configuring SIP Between a Gateway and CallManager 5.x

In this case study, SIP is being introduced into the New York and Shanghai networks. The locations will use both SIP and SCCP phones, and a SIP trunk will be configured between the New York CallManager cluster and the Shanghai gateway, shown in Figure 4-12. CallManager 5.x is used to register the SIP phones. The preferred DTMF relay method will be SIP-NOTIFY, but RTP-NTE will be accepted as a second choice.

For redundancy, the Shanghai gateway has dial peers to two of the CallManager servers. One CallManager IP address is configured under the UA, to demonstrate that command. The gateway will register its network analog phones with registrar servers at 10.10.10.19 and 10.10.10.20. HTTP digest authentication is configured.

All SIP traffic will use interface Fa0/0 as its source IP address. Large packets will switch from UDP to TCP as their transport protocol to avoid UDP fragmentation.

Figure 4-12 *SIP Case Study Network Diagram*



Example 4-14 shows the configuration of SIP and the SIP trunk on the Shanghai gateway. POTS dial peers to connect to the PBX are not shown in this example. CallManager configuration is outside the scope of this case study.

Example 4-14 *SIP Case Study*

```
Shanghai-GW(config)#sip-ua
Shanghai-GW(config-sip-ua)#sip-server ipv4:10.10.10.11
Shanghai-GW(config-sip-ua)#registrar ipv4:10.10.10.19
Shanghai-GW(config-sip-ua)#registrar ipv4:10.10.10.20 secondary
Shanghai-GW(config-sip-ua)#authentication username NYCServ1 password C1sc0 realm
  NYC-SIP
!
Shanghai-GW(config)#voice service voip
Shanghai-GW(conf-voi-serv)#sip
Shanghai-GW(conf-serv-sip)#bind all source-interface fa0/0
Shanghai-GW(conf-serv-sip)#transport switch udp tcp
!
Shanghai-GW(config)#dial-peer voice 2121 voip
Shanghai-GW(config-dial-peer)#destination-pattern 21255521..
Shanghai-GW(config-dial-peer)#session target sip-server
Shanghai-GW(config-dial-peer)#session protocol sipv2
Shanghai-GW(config-dial-peer)#dtmf-relay sip-notify rtp-nte
Shanghai-GW(config-dial-peer)#preference 1
```

Example 4-14 *SIP Case Study (Continued)*

```
!
Shanghai-GW(config)#dial-peer voice 2122 voip
Shanghai-GW(config-dial-peer)#destination-pattern 21255521..
Shanghai-GW(config-dial-peer)#session target 10.10.10.12
Shanghai-GW(config-dial-peer)#session protocol sipv2
Shanghai-GW(config-dial-peer)#dtmf-relay sip-notify rtp-nte
Shanghai-GW(config-dial-peer)#preference 2
```

Review Questions

1. What do the acronyms UAC and UAS stand for? Define the difference between the two entities.
2. Name five types of SIP servers, and describe what they do.
3. Name the five types of SIP methods that Cisco routers can both generate and respond to. What is the purpose of each one?
4. What command configures a dial peer to use SIP in its communication with its VoIP peer?
5. How does CallManager 4.x interact with a SIP gateway?
6. What additional SIP capabilities does CallManager 5.x add?
7. Which Layer 4 protocol does SIP use by default, and what is the default port?
8. What is the function of the SDP in SIP call setup?

CHAPTER 5

Circuit Options

In this chapter, you will learn about the various circuit types that Cisco voice gateways support, in addition to signaling methods and available features of circuits.

NOTE For more information on the implementation details, see Chapter 6, "Connecting to the PSTN," and Chapter 7, "Connecting to PBXs."

This chapter helps you to understand the following:

- Analog circuit options and signaling methods
- Digital circuit options and signaling methods
- ISDN technology
- Voice tuning on different types of circuits

Circuit Signaling

All voice circuits use signaling methods to communicate. These signaling methods vary based on the type of circuit, but all circuits must communicate the same types of information. This information is grouped by the type of information and signaling method:

- **Supervisory signaling**—Supervisory signaling is used to indicate the state or current status of a circuit. A circuit, or circuit channel, can be either on-hook, which indicates that the circuit is available, or off-hook, which indicates that the circuit is in use.
- **Address signaling**—Address information consists of the digits that are assigned to an end station. Address signaling methods are required to transmit this information from the call originator.

  Multifrequency (MF) tones are commonly used for address signaling. MF signaling uses numerous reference frequencies sent two at a time to represent each digit. Each protocol defines the reference frequencies and how each pair of frequencies, or tone-pair, maps to a digit.

- **Informational signaling**—Informational signaling is used to provide feedback to the calling or called party. Common informational signaling includes dial tone, ring indication, and busy signal.

Various terms describe the address information. People often use these terms interchangeably, although they actually refer to specific standards or signaling methods.

- **Dialed number identification service (DNIS)**—The telephone number of the called party. Specifically, DNIS refers to the number that the service provider sends to the destination. DNIS might or might not be the digits that the calling party dials to reach the destination.
- **Automatic number identification (ANI)**—The telephone number of the calling party.
- **Calling number identification or calling name identification (CNID)**—CNID is used as a synonym to ANI.
- **Calling line identification (CLID)**—Similar to ANI. Refers to the number of the calling party.
- **Caller ID**—Generic term that describes the delivery of calling number or name. Also used as a marketing term by many service providers to refer to a chargeable feature offered on analog phone service.

Analog Circuits

Cisco voice gateways support three types of analog circuits:

- Foreign Exchange Station (FXS)
- Foreign Exchange Office (FXO)
- Ear and Mouth (E&M)

FXS/FXO

An FXO is a port type that connects subscribers to central office (CO) equipment or PBXs. You can use FXS to connect to regular analog phones or PBXs.

In general, an FXS port is a port that comes from the service offering equipment, whether it is a CO class five switch, a PBX, or a voice gateway. An FXO port is a port from an end device, such as a telephone, fax, or modem, that is connected to an FXS port to obtain telephony services.

FXS

FXS ports are the ports that you plug a telephone, fax, or modem into that provide telephony service. For example, an analog telephone that is connecting over twisted copper pairs to the port in a home or a port on a PBX is connecting to the FXS port of the telephone carrier or the private telephone exchange. FXS ports provide dial tone, battery current, and ring voltage to the subscriber device.

Dial tone is the current that the FXS port places on the analog circuit to inform the subscriber device that the line is ready to collect dual tone multifrequency (DTMF) tones for call routing.

Idle voltages range from 24V to 48V and are configurable. *Ring voltage* is the voltage that is placed on the analog circuit to inform subscriber devices of an incoming call.

Cisco routers provide 40Vrms 5 ring equivalence number (REN), which is sufficient to support on-premise equipment.

FXO

FXO ports are the ports on subscriber devices that connect to the CO or PBX to receive subscriber services. Examples are the port on a telephone, modem, or fax that connects to the FXS port in the wall, PBX, or voice gateway. PBX systems and voice gateways can also contain FXO ports that are connected to the CO. For example, you can connect an analog telephone FXO port to the FXS port from a PBX for voice service. This FXS port allows the analog phone to call other phones that are connected to the PBX. The PBX might also contain an FXO port that is connected to a CO FXS port. This FXO port allows the PBX to route calls from its FXS ports to the public switched telephone network (PSTN). On-hook/off-hook indication is delivered to the FXS port by a change in voltage on the analog line.

FXO Power Failover

If the power to the router fails, a metallic path is established between certain FXO and FXS ports. You can mark the analog phones that are connected to these FXS ports as emergency phones for calling out to the PSTN even when the router is down.

FXS/FXO Supervisory Signaling

A subscriber device that is connected to an FXS port initiates communications by delivering off-hook indication. An FXS port detects the off-hook state and prepares to gather DTMF tones used to determine how to route the call. The remote FXS port applies ring voltage to the remote subscriber FXO, and the remote FXO detects ring voltage and goes off-hook, indicating reception of the call and closing of the call circuit. The call is

ended by each local segment entering the on-hook state and the FXS port sending on-hook indication. You can accomplish supervisory signaling between FXS and FXO ports in two ways: loop-start and ground-start signaling.

Loop-Start Signaling

Loop-start signaling involves the breaking and connecting of the 48V circuit loop originating from the CO equipment. During the on-hook state, the subscriber equipment has a break on the loop, so no voltage is passing. When the subscriber equipment goes off-hook, it closes the loop, and current flows. At this point, the CO detects the closing of the loop and provides a dial tone. To seize the far-end subscriber, the CO provides an analog current across the line called ring voltage. This current causes the subscriber device to "ring," indicating an incoming call. When the subscriber answers the call, or goes off-hook, the 48V circuit loop is closed and the CO turns off the ringing voltage. The call is now connected.

The loop-start method has the advantage of not requiring a common ground between the CO and subscriber equipment. However, it does not provide far-end disconnect monitoring. There is no mechanism in place to determine when the remote party has returned to the on-hook state. There is also poor glare resolution with loop-start signaling. Glare occurs when both the CO and subscriber equipment attempt to seize the line at the same time.

Ground-Start Signaling

Ground-start signaling calls for the 48V circuit loop to be grounded on both sides of the connection. In the idle state, the subscriber has a break in the ring, and the CO has a break in the tip. When the subscriber seizes the line, it grounds the ring, allowing current to flow along the path. When the CO senses the ring ground, it grounds the tip. Upon detection of the tip ground, the subscriber closes the loop and removes the ring ground. This action connects the circuit. When the CO seizes the line, it grounds the tip, causing the subscriber to detect the current, close the loop, and remove the ring ground. Ground-start signaling provides far-end disconnect information because the CO can ground the tip when the remote subscriber has entered the on-hook state. It also minimizes instances of glare. However, it requires that both CO and subscriber equipment have a common ground.

Address Signaling

The FXO port sends address signaling to the FXS port to indicate the final destination of the call. Analog address signaling can use either pulsed digits or DTMF tones. DTMF is a common implementation of MF tones that defines eight reference frequencies and tone-pair to digit mapping.

Pulsed tones are rarely used today. Pulse tones were originally sent by spinning a dial wheel on a telephone. As this mechanical device spun, it opened and closed the circuit. The pulses, or duration of open and closed, vary by country but must be consistent within a specified tolerance.

Today, most analog circuits use DTMF tones to indicate the destination address. As you push the buttons on a touch-tone phone, two different frequencies are sent on the circuit. This combination of frequencies notifies the receiving side of the digits. Table 5-1 lists the frequencies that are associated with each button on a telephone keypad.

Table 5-1 *DTMF Frequencies*

	1209	**1336**	**1477**	**1633**
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

FXS-DID

Because address signaling is transmitted only from the FXO port to the FXS port, calls from the PSTN on a typical analog circuit do not indicate the end destination of a call. For a typical residential application, this is not an issue, because all phones on the associated line will ring. When the analog service is connected to a gateway or PBX, it would be beneficial to receive addressing signaling to allow calls to be routed to individual phones on the system. Some service providers offer this service by providing an FXO port as the demarcation. You can then connect an FXS port that is configured for direct inward dial (DID) service and receive address signaling from the PSTN. You cannot configure all FXS ports to support DID service.

NOTE You cannot use FXS ports that are configured for DID service to place outbound calls.

Informational Signaling

The FXS port provides informational signaling using call progress tones. Call progress (CP) tones are audible tones that the FXS device sends to indicate the status of calls. Functions are determined by the frequency of tone sent and the cadence of the tone. For example, in the United States, a dial tone is a continuous generation of 350 Hz and 440 Hz,

whereas a busy signal is 480 Hz and 620 Hz with a .5 second on/.5 second off cadence. A sample list of call progress tones is as follows:

- Busy signal
- Call waiting
- Dial tone
- Ring tone
- Congestion tone
- Audible ringback tone

CP Tones are country specific. Cisco voice gateways default to U.S. CP tones. You can select country-specific tones using the voice port command **cptone**, as shown in Example 5-1.

Example 5-1 *Configuring CP Tones*

```
Gateway#config t
Gateway(config)#voice-port 2/0/0
Gateway(config-voiceport)#cptone ?
  locale   2 letter ISO-3166 country code

AR Argentina          IS Iceland            PE Peru
AU Australia          IN India              PH Philippines
AT Austria            ID Indonesia          PL Poland
BE Belgium            IE Ireland            PT Portugal
BR Brazil             IL Israel             RU Russian Federation
CA Canada             IT Italy              SA Saudi Arabia
CN China              JP Japan              SG Singapore
CO Colombia           JO Jordan             SK Slovakia
C1 Custom1            KE Kenya              SI Slovenia
C2 Custom2            KR Korea Republic     ZA South Africa
CY Cyprus             LB Lebanon            ES Spain
CZ Czech Republic     LU Luxembourg         SE Sweden
DK Denmark            MY Malaysia           CH Switzerland
EG Egypt              MX Mexico             TW Taiwan
FI Finland            NP Nepal              TH Thailand
FR France             NL Netherlands        TR Turkey
DE Germany            NZ New Zealand        GB United Kingdom
GH Ghana              NG Nigeria            US United States
GR Greece             NO Norway             VE Venezuela
HK Hong Kong          PK Pakistan           ZW Zimbabwe
HU Hungary            PA Panama
```

Caller ID

Caller ID is a service provider feature that allows the receiving party to be notified of the directory number of the calling party. Caller ID is based on Telcordia Publications TR-TSY-000030 and TR-TSY-000031 and is officially known as Calling Number Delivery (CND).

CND sends the calling party information between the first and second ring of a call. The information is sent as ASCII-encoded text in a 1200 bps stream of data bytes and includes the date, time, and calling party number and name.

FXO ports support inbound Caller ID. You can also configure a name and number for FXS ports by using the **station-id** command. This information is used for calls within a system only. Calls to the PSTN present the official directory information stored in the service provider database.

NOTE CallManager-controlled Media Gateway Control Protocol (MGCP) gateways do not support inbound caller ID on FXO ports.

Supervisory Disconnect

Traditionally, there was no need for the FXS port to inform the FXO port of disconnect because the end user was a person. During a phone conversation, people indicate the end of the discussion or assume the end of a discussion by a prolonged silence. Even if the end user is a modem or fax, disconnect indication is not required because this type of device can detect when it loses the carrier from the far end. However, complications arise when the FXO port is part of another switching system, such as a PBX or voice gateway. During loop-start signaling, the status, or battery state, of the FXS device never changes. If the FXO port is on another switching device, the remote end might become disconnected, but the PBX or Voice over IP (VoIP) gateway has no indication of the disconnect. In this case, the FXO port can be left in a hung state.

The FXS port can signal call disconnect to the FXO port using several methods:

- If you use ground-start signaling, the FXO receives a disconnect indication when the voltage level changes.
- If you use loop-start signaling, you can configure the FXS port to temporarily deny power when the call disconnects. The FXO port can sense this short loss of power and disconnect the call. Power denial support is enabled by default on Cisco FXO ports.
- You can configure the FXS for battery reversal using the **battery-reversal** command under voice port configuration. FXS ports normally reverse battery polarity at the point of far-end answer. You can set the FXS to reverse the polarity again during far-end hang-up to indicate disconnect to the FXO.

- The FXS can send a tone-based supervisory disconnect to the FXO. After receiving this tone, the FXO port knows that the call has disconnected. The tone that is sent varies by country. Example 5-2 shows how to configure Supervisory Tone Disconnect on an FXO port.

Example 5-2 *Configuring Supervisory Tone Disconnect*

```
Gateway#config t
Gateway(config)#voice-port 2/0/0
Gateway(config-voiceport)#supervisory disconnect dualtone mid-call
Gateway(config-voiceport)#cptone us
Gateway(config-voiceport)#timeouts wait-release 5
Gateway(config-voiceport)#timeouts call-disconnect 5
Gateway(config-voiceport)#exit
```

If the default disconnect tone for a country does not work, you can specifically define the tone. You can modify settings for frequencies, power, delay, and cadence. You must obtain from the service provider the information that is required to configure the tone, or you must determine it using a frequency analyzer. Example 5-3 shows how to customize the disconnect tone.

Example 5-3 *Customizing the Supervisory Disconnect Tone*

```
Gateway #configure terminal
Gateway(config)#voice-port 3/1/1
Gateway(config-voiceport)#supervisory disconnect dualtone pre-connect voice-class
10
Gateway(config-voiceport)#end

 Gateway(config)#voice class dualtone 10
     Gateway(config-voice-class)#freq-pair 1 350 440
     Gateway(config-voice-class)#freq-max-deviation 5
     Gateway(config-voice-class)#freq-max-power 6
     Gateway(config-voice-class)#freq-min-power 25
     Gateway(config-voice-class)#freq-power-twist 15
     Gateway(config-voice-class)#freq-max-delay 16
     Gateway(config-voice-class)#cadence-min-on-time 100
     Gateway(config-voice-class)#cadence-max-off-time 250
     Gateway(config-voice-class)#cadence-list 1 100 100 300 300 100 200 200 200
     Gateway(config-voice-class)#cadence-variation 8
     Gateway(config-voice-class)#exit
```

CAMA

Centralized Automated Message Accounting (CAMA) is an old telephony protocol that was developed for long-distance billing but is widely used today in the United States for enhanced 911 (E911) services. The protocol allows for both the calling and called number to be carried using in-band signaling. This is needed in 911 calling because 911 calls are

routed to the appropriate service center based on the calling number. CAMA allows the telephone company to transmit the calling number to the public safety answering point (PSAP) dispatcher prior to establishing the audio connection. This is crucial information because it tells the dispatcher the exact location that the call is coming from and provides callback information if the call is disconnected.

Most E911 networks are standalone entities within their own region. Many states have laws requiring that businesses connect directly to the local E911 network, and this legislation is expected to be passed throughout the United States in the near term. More than 80 percent of the E911 networks in the United States consist of CAMA trunks, whereas some E911 regions allow alternative connections, such as ISDN. A CAMA trunk can either be an analog telephone line carrying MF signaling or a voice T1 with MF signaling. The enterprise device, whether it is a voice gateway or PBX, connects directly to the E911 device. This device routes the call to the appropriate PSAP based on the calling number of the incoming call.

You can configure VIC2 FXO ports to support CAMA signaling.

E&M

E&M signaling was developed to interconnect PBXs using dedicated circuits from the PSTN. Unlike the 2-wire FXS/FXO circuits, E&M circuits are 8-wire. The voice path uses either 2-wire or 4-wire. The remaining four wires are used for signaling. Table 5-2 lists the wires, or leads, that are used in E&M circuits.

Table 5-2 *E&M Leads*

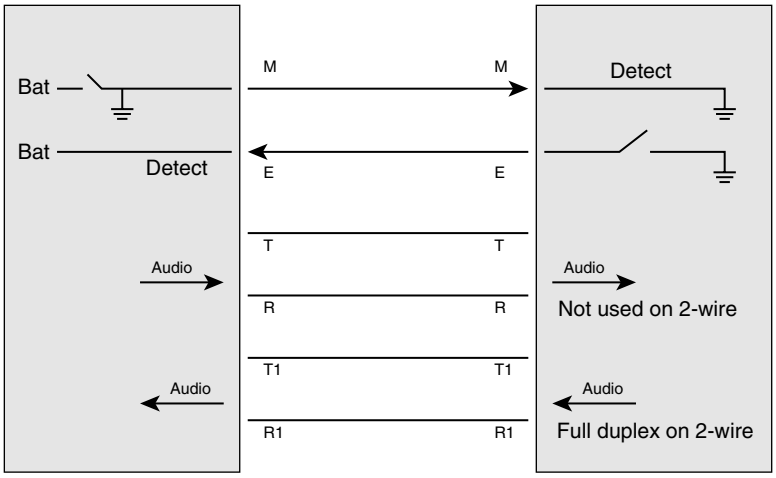
Lead	Description	Pin
SB	-48V signaling battery	1
M	Signaling input	2
R	Ring, audio input	3
R1	Ring, audio input/output	4
T1	Tip, audio input/output	5
T	Tip, audio input	6
E	Signaling output	7
SG	Signaling ground	8

E&M comes in six common variants: E&M Types I through V and E&M SSDC5. This section discusses E&M Types I through V. The Type indicates how the various signaling leads are used to indicate an off-hook or trunk seizure condition.

Type I Signaling

Type I signaling is the most common version seen in North America. Battery is provided on both the E and the M lead. During on-hook, both of the leads are open. The trunk circuit side indicates off-hook by connecting the M lead to the battery, while the signaling unit indicates off-hook by grounding the E lead, as shown in Figure 5-1.

Figure 5-1 *E&M Type I Signaling*



Condition	M Lead	E Lead
On-Hook	Ground	Open
Off-Hook	Battery	Ground

Type II Signaling

Type II signaling interfaces cause little interference and are typically used for sensitive environments. These interfaces use four leads for signaling: E, M, SB, and SG. During on-hook, both the E and M lead are open. The trunk side indicates off-hook by connecting the M lead to the SB lead connected to the battery of the signaling side. The signaling side indicates off-hook by connecting the E lead to the SG lead that is connected to the trunk circuit ground.

Type III Signaling

Type III signaling is not commonly used. It also uses four leads, but during on-hook, the E lead is open and the M lead is set to the ground that is connected to the SG lead of the signaling side. The trunk side indicates off-hook by moving the M lead connection from the

SG lead to the SB lead of the signaling side. The signaling unit indicates off-hook by grounding the E lead.

Type IV Signaling

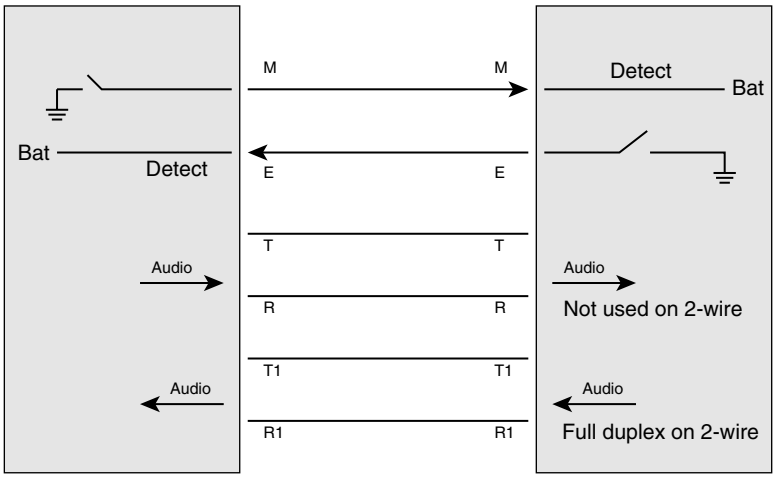
Type IV signaling is similar to Type II. It also uses four leads, and during on-hook, both the E and the M leads are open. The trunk side indicates off-hook by connecting the M lead to the SB lead that is connected to the ground of the signaling side. The signaling side indicates off-hook by connecting the E lead to the SG lead that is connected to the ground of the trunk side.

NOTE Cisco E&M voice interface cards (VIC) do not support E&M Type IV signaling.

Type V Signaling

Type V signaling is the most common mechanism used outside of North America. It is similar to Type I, and it only uses the E and M leads. During on-hook, both leads are open. The trunk side indicates off-hook by grounding the M lead, while the signaling side indicates off-hook by grounding the E lead, as shown in Figure 5-2.

Figure 5-2 *E&M Type V Signaling*



Condition	M Lead	E Lead
On-Hook	Open	Open
Off-Hook	Ground	Ground

Address Signaling

E&M supports three mechanisms of start dial signaling used between off-hook and digit collection: wink-start, delay-start, and immediate-start. Like FXS/FXO, both pulse dialing and DTMF can be used for address transmission.

Wink-Start

In wink-start operation, the originating device goes off-hook using the signaling leads as determined by the Type configuration. When the remote switch detects that the originating switch is off-hook, it transmits an off-hook pulse of approximately 140 to 290 ms in duration and then returns to the on-hook state. This is the "wink." The originating switch detects the wink, waits for at least 100 ms, and then outputs digits to the remote switch. The remote switch extends the call based on the receive digits. After the called party answers the call, the remote switch indicates call answer by transmitting off-hook.

Delay-Start

During delay-start, the originating switch waits a configurable time before inspecting the incoming signal from the remote switch. If the signal indicates on-hook, the originating switch outputs digits to the remote switch. If the signal is off-hook, the originating office waits until the signal returns to on-hook before forwarding digits. The remote switch indicates call answer by transmitting off-hook.

Immediate-Start

Immediate-start is the most basic of the trunk-signaling methods. The originating switch goes off-hook, waits for at least 150 ms, and then forwards digits.

Digital Circuits

Digital circuits provide significant advantages over analog circuits. Analog circuits require one port per conversation, which can result in port density issues and can be susceptible to interference.

Digital circuits overcome some of these limitations by digitizing audio waveforms and transmitting multiple calls over a single circuit. The process of digitizing an analog signal requires that the signal be sampled, quantified, and encoded. Henry Nyquist, who was an engineer at AT&T, developed a method for digitizing analog signals in 1924. What is known as Nyquist's theorem states that to digitize an analog signal, the signal must be sampled at a rate twice that of the highest frequency. Although human speech has a wide range, analog telephone channels typically carry frequencies in ranges between 200 and 3300 Hz. Because of variations in the frequency ranges supported at the time, Nyquist set

the maximum frequency at 4000 Hz, resulting in a sampling rate of 8000 times per second. Each of these 8000 samples is quantified, or assigned a numerical value, based on a reference scale. The scale uses 255 values, which can be encoded using 8 bits.

T1

T1 circuits evolved in early voice networks as a mechanism to transfer multiple calls across one copper transmission medium. The copper loop can carry significantly more bandwidth than the 4000 Hz required for voice transmission. By electronically changing the voice frequency of incoming calls, frequency division multiplexing (FDM) was used to carry 24 calls across one copper loop using 96000 Hz of spectrum. Today, digital transmission has changed the T1 into a TDM circuit transmitting 1s and 0s instead of analog signals.

One voice channel in digital form requires 64 kbps of bandwidth. This is calculated by multiplying the 8000 samples per second × 8 bits per sample = 64 kbps. This 64-kbps package is known as the *digital signal level zero*, or DS-0. At 64 kbps, the 24 voice channels represent 1.536 Mbps of data. An additional 8000 bps were added for framing, bringing the speed of the T1 circuit to 1.544 Mbps.

Framing is the mechanism used by transmitting devices to ensure the synchronization and organization of user data. A T1 time slot is an 8-bit segment for each DS-0. A frame consists of 24 time slots and one framing bit, for a total of 193 bits. The transmitting and receiving equipment use the framing bit; the bit does not contain user data. It must be recognizable in the transmission, so the receiver looks for a pattern that repeats every 12 frames: 100011011100. This 12-frame unit is called a *Super Frame* (SF), or D4. In some cases, the voice signaling is transmitted by using the least significant bit of the sixth and twelfth frame of the SF. The A-bit is robbed from the sixth frame, and the B-bit is robbed from the twelfth frame. Table 5-3 illustrates an SF. The bolded rows show the position of the signaling bits if robbed-bit signaling is used.

Table 5-3 *SF (D4)*

Frame Number	Framing Bit	Channel 1	Channel 2	Channel 3-23	Channel 24
1	1	12345678	12345678	12345678	12345678
2	0	12345678	12345678	12345678	12345678
3	0	12345678	12345678	12345678	12345678
4	0	12345678	12345678	12345678	12345678
5	1	12345678	12345678	12345678	12345678
6	**1**	**1234567A**	**1234567A**	**1234567A**	**1234567A**
7	0	12345678	12345678	12345678	12345678
8	1	12345678	12345678	12345678	12345678
9	1	12345678	12345678	12345678	12345678

continues

Table 5-3 *SF (D4) (Continued)*

Frame Number	Framing Bit	Channel 1	Channel 2	Channel 3-23	Channel 24
10	1	12345678	12345678	12345678	12345678
11	0	12345678	12345678	12345678	12345678
12	**0**	**1234567B**	**1234567B**	**1234567B**	**1234567B**

As digital technology arose, the need for more signaling states led to the development of the Extended Superframe (ESF). ESF doubled the number of frames in the SF from 12 to 24. This allowed for two more signaling bits, C and D, to be robbed from frames 18 and 24, respectively. As technology advanced, only 2 kbps were required for synchronization in a 6-bit framing pattern 001011. The other framing bits are used to carry 2 kbps of CRC error checking and 4 kbps of data link information, such as performance information or alarms. Table 5-4 illustrates an ESF frame. In the table, FPS indicates a synchronization bit, DL indicates a data link bit, and CRC indicates a CRC bit.

Table 5-4 *ESF Frame*

Frame Number	Framing Bit	Channel 1	Channel 2	Channel 3-23	Channel 24
1	1/0 DL	12345678	12345678	12345678	12345678
2	1/0 CRC1	12345678	12345678	12345678	12345678
3	1/0 DL	12345678	12345678	12345678	12345678
4	0 FPS	12345678	12345678	12345678	12345678
5	1/0 DL	12345678	12345678	12345678	12345678
6	**1/0 CRC2**	**1234567A**	**1234567A**	**1234567A**	**1234567A**
7	1/0 DL	12345678	12345678	12345678	12345678
8	0 FPS	12345678	12345678	12345678	12345678
9	1/0 DL	12345678	12345678	12345678	12345678
10	1/0 CRC3	12345678	12345678	12345678	12345678
11	1/0 DL	12345678	12345678	12345678	12345678
12	**1 FPS**	**1234567B**	**1234567B**	**1234567B**	**1234567B**
13	1/0 DL	12345678	12345678	12345678	12345678
14	1/0 CRC4	12345678	12345678	12345678	12345678
15	1/0 DL	12345678	12345678	12345678	12345678
16	0 FPS	12345678	12345678	12345678	12345678
17	1/0 DL	12345678	12345678	12345678	12345678
18	**1/0 CRC5**	**1234567C**	**1234567C**	**1234567C**	**1234567C**

Table 5-4 *ESF Frame (Continued)*

Frame Number	Framing Bit	Channel 1	Channel 2	Channel 3-23	Channel 24
19	1/0 DL	12345678	12345678	12345678	12345678
20	1 FPS	12345678	12345678	12345678	12345678
21	1/0 DL	12345678	12345678	12345678	12345678
22	1/0 CRC6	12345678	12345678	12345678	12345678
23	1/0 DL	12345678	12345678	12345678	12345678
24	**1 FPS**	**1234567D**	**1234567D**	**1234567D**	**1234567D**

Channel-Associated Signaling

As described earlier, bit robbing allows using the least significant bit in some timeslots to transmit signaling information—the sixth and twelfth frame for SF; and the sixth, twelfth, eighteenth, and twenty-fourth frame for ESF. This method of voice signaling on a T1 is known as *channel-associated signaling* (CAS) or *in-band signaling*. The common forms of signaling on CAS circuits are the same as analog circuits: loop-start, ground-start, and E&M.

In loop-start operation, the FXO uses the B bit, and the FXS uses the A bit. In the on-hook state, the FXS transmits an A bit value of 0, and the FXO transmits a B bit value of 0. To indicate an incoming call, the FXO rings the FXS by toggling the B bit between 0 and 1. The FXS responds by going into the on-hook state and setting the A bit to 1. During a disconnect, the FXO does not change signaling state, and the FXS must detect the end of the call and toggle the A bit back to 0.

Ground-start operation requires the FXO to use both the A and the B bit, whereas in most applications, the FXS still only uses the A bit. In the on-hook state, both the A and B bits of the FXO are set to 1, and the A bit of the FXS is set to 0. During a call from the FXO to the FXS, the FXO sets the A bit to 0 and toggles the B bit to indicate ringing. FXS detects the ringing and sets the A bit to 1 to indicate off-hook. The network goes off-hook by keeping the A bit at 0 and setting the B bit to 1. When the call disconnects, the FXO sets the A bit to 1; upon detection, the FXS sets the A bit to 0 to indicate on-hook.

E&M Signaling

E&M signaling is the preferred method when using T1 CAS circuits. Both network and user transmit all 0s for on-hook and all 1s for off-hook or line seizure. The different methods of E&M signaling described in the “Analog Circuits” section of this chapter are

applicable for digital CAS circuits. Cisco voice gateways support the following E&M signaling types:

- **e&m-wink-start (Feature Group B [FGB])**—The receiving side acknowledges that the originator is off-hook by toggling the A and B bit from 0 to 1 for approximately 200 ms. After the wink, the originator sends the address information.
- **e&m-delay-dial**—The originator goes off-hook and waits 200 ms. Then it verifies that the receiver is on-hook before sending address information.
- **e&m-immediate-start**—The originator goes off-hook and sends the address information. The receiver goes off-hook after receiving the digits.
- **e&m-fgd (Feature Group D [FGD])**—The receiving side acknowledges that the originator is off-hook with a wink. After the wink, the originator sends the address information. The receiver acknowledges the digits with a second wink. FGD allows receipt of additional address information such as ANI.
- **fgd-eana**—FGD Exchange Access North America uses the double-wink mechanism as FGD. FGD-EANA allows sending additional address information to support call services such as ANI or emergency calls.
- **fgd-os**—FGD Operator Services is used for calls from a Bell Operating Company sent toward the carrier switch. It is not typically used in enterprise environments.

Feature Group D

FGD defines the rules that are used for connection between local carriers and interexchange carriers. A trunk that is running the FGD protocol ensures equal access of services for the different carriers. This allows options for end customers and fair access to all services available from all carriers. The protocol passes information, such as the ANI, or calling number, across the FGD trunk.

Both e&m-fgd and fgd-eana support the delivery of DNIS, or called number. e&m-fgd can receive but not send ANI. Fgd-eana can send but not receive ANI. If you require both incoming and outgoing ANI information, you need to have two circuits or circuit groups, as shown in Example 5-4.

Example 5-4 *Sending and Receiving ANI Using FGD*

```
Gateway#config t
Gateway(config)#controller t1 1/0
Gateway(config)#framing esf
Gateway(config)#linecode b8zs
Gateway(config-controller)#ds0-group 1 timeslots 1-12 type e&m-fgd
Gateway(config-controller)#ds0-group 2 timeslots 13-24 type fgd-eana
Gateway(config-controller)#end
Gateway#
```

DS-0 group 1 is used for inbound calls and can receive ANI. DS-0 group 2 is used for outbound calls and can send ANI.

E1

An E1 circuit is similar to a T1 in that it is a TDM circuit that carries DS-0s in a bundled connection. The E1 is widely used in countries in Europe, Asia, and South/Central America. The framing format for E1 circuits is defined in Consultative Committee for International Telegraph and Telephone (CCITT) recommendation G.704 and is supplemented by G.732. It consists of 32 timeslots and a transmission rate of 2.048 Mbps. Of the 32 timeslots, 1 is used for framing, 1 is used for telephony signaling, and 30 are available for user data. Timeslot 0 is used for framing, alarm transport, and international bits. The standard also allows for optional CRC-4 error checking. Timeslot 16 is used to carry signaling information, whether it is CAS, ISDN, SS7, or proprietary signaling.

Cisco supports the same CAS methods on E1 circuits as T1 circuits. Cisco also supports Mercury Exchange Limited (MEL) CAS versions of E&M and FXO/FXS signaling. MEL CAS is primarily used in the United Kingdom.

E1 R2

R2 signaling is defined in ITU Recommendations Q.400 through Q.490 and defines two types of signaling: line signaling and inter-register signaling. R2 signaling is either an international version known as CCITT-R2 or a country-specific variant. Cisco supports both the International Telephony Union (ITU) standard and many country and regional variants.

Line Signaling

Line signaling is used for supervisory signals for call setup and teardown. R2 supports three methods of line signaling:

- **R2-Digital**—R2-Digital is defined in ITU-U Q.421. It is typically used for pulse code modulation (PCM) systems and is described in more detail in this section.
- **R2-Analog**—R2-Analog is defined in ITU-U Q.411 and is typically used in carrier systems. Signaling uses a Tone/A bit.
- **R2-Pulse**—R2-Pulse is defined in ITU-U Supplement 7. It is a variant of R2-Analog in which the Tone/A bit is pulsed rather than continuous and is typically used for satellite links.

R2-Digital signaling uses CAS carried in timeslot 0 of the E1 frame. Of the four bits that are available, only the A and B bits are used by R2 line signaling. In the on-hook state, the A bit is set to 1 and the B bit is set to 0. These bits have different meanings depending on

which side is initiating the call. For the purpose of this discussion, *forward* is defined as bits coming from the calling party, and *backward* is defined as bits coming from the called party. Table 5-5 lists the bit settings for CCITT-R2 signaling.

Table 5-5 *R2 Signaling Bits*

Direction	Signal Type	A Bit	B Bit
Forward	Seizure (off-hook)	1 -> 0	0
Forward	Clear (on-hook)	0 -> 1	0
Backward	Seizure-ack	1	0 -> 1
Backward	Answer	1 -> 0	1
Backward	Clear-back	0 -> 1	1
Backward	Release-guard	0 -> 1	1 -> 0

NOTE 0 -> 1 indicates a bit state transition from 0 to 1. A single digit indicates that the bit state does not change.

Inter-Register Signaling

Register is the term used to describe the switches at each end of the E1 circuit. Therefore, inter-register signaling is used between the switches, or registers, for both address and informational messaging. Informational messaging between registers includes information on priority levels, congestion notification, or charges. Inter-register signaling uses multifrequency tones in the timeslot used for the call so that it is in-band signaling. This signaling is a form of *multifrequency compelled* (MFC) signaling, meaning that a signal from one end is acknowledged by a tone from the opposite end. Most country variants of R2 involve inter-register signaling.

Forward signals are signals from the calling party toward the called party and are classified as Group-I or Group-II signals. Group-I signals convey address information, including DNIS and ANI. Group-II signals identify additional information about the calling party, such as priority level. Backward signals are signals from the called party to the calling party and are classified as Group-A or Group-B. Group-A signals acknowledge Group-I signals and convey additional information, such as congestion. Group-B signals acknowledge Group-II signals and provide called party information. Table 5-6 lists the R2 inter-register signal types.

Table 5-6 *Inter-Register Signals*

Group	Signal	Function
Group-I	I-1 to I-10	Digits 1 to 10
	I-15	Indicates end of address
Group-II	II-1	Subscriber without priority
	II-2 to II-9	Subscriber priority levels
	II-11 to II-15	For national use
Group-A	A-1	Send next digit
	A-3	Address complete
	A-4	Congestion
	A-5	Send calling party category
	A-6	Address complete, charge, setup, speech conditions
Group-B	B-3	Subscriber busy
	B-4	Congestion
	B-5	Unallocated number
	B-6	Subscriber line free charge

Cisco voice gateways support four types of inter-register signaling:

- **R2-Compelled**—Forward tones stay on until the remote end responds. The tones are compelled to stay on until you turn them off.
- **R2-Noncompelled**—Forward tones are sent as pulses. Group-B responses are also sent as pulses. Noncompelled inter-register signaling has no Group-A signals.

NOTE Most installations use the noncompelled type of inter-register signaling.

- **R2-Semi-Compelled**—Forward tones are sent as compelled. Responses are sent as pulses. Semi-compelled is the same as compelled, except that the backward signals are pulsed instead of continuous.
- **DTMF**—In-band DTMF tones are used for address signaling.

ISDN

ISDN is a digitized service offering of telephone carriers to allow users to transmit voice, data, video, and other applications over existing telephone wiring. With ISDN, one channel, called a D-channel, is designated as the signaling channel. It is either 16 kbps or 64 kbps. User data is carried in 64-kbps channels, which are called B–channels, or bearer channels. Because all signaling occurs on the D-channel, ISDN is also called common channel signaling (CCS).

ISDN circuits come in two types: BRI and PRI. A BRI consists of two B-channels and one 16-kbps D-channel, for a total bandwidth of 144 kbps. A T1 PRI consists of 23 B-channels and 1 64-kbps D-channel. An E1 PRI consists of 30 B-channels and 1 64-kbps D-channel.

In its early history, ISDN was the high-speed connection of choice for end users. Quick connection time, the ability to have more bandwidth, and the option to connect several services to one line made ISDN much more attractive than dial services. With the advent of broadband residential services, such as cable and digital subscriber line (DSL), the popularity of ISDN in the home has diminished. The BRI is still used in many data applications, typically as a backup to a WAN connection or for voice connections in Europe, and the PRI is used widely in voice applications.

Covering all aspects of ISDN would take an entire book. This section focuses on the aspects required to connect an ISDN PRI circuit to a Cisco voice gateway.

ISDN Switch Configuration

For PRIs, a Cisco voice gateway acts as an ISDN switch. Various switch types are defined. ISDN switches use a master/slave designation to control signaling. One switch is designated as the network side, or master, and one switch is designated as the user side, or slave. The ISDN Layer 2 protocol Link Access Procedure, D-channel (LAPD) establishes and maintains the D-channel. To establish the D-channel, the network/user configuration must be correct. When connecting to the PSTN, the gateway should be the user side. When connecting to a PBX, either the gateway or the PBX must be configured for the network side. Table 5-7 lists the switch types that Cisco voice gateways support.

Table 5-7 *ISDN Switch Types for PRI*

Switch Type	Description	User Side	Network Side
4ESS	Lucent 4ESS—United States	Y	Y
5ESS	Lucent 5ESS—United States	Y	Y
DMS100	Northern Telecom DMS100—United States	Y	Y
DPNSS	DPNSS—Europe	Y	Y

Table 5-7 *ISDN Switch Types for PRI (Continued)*

Net5	United Kingdom, Europe, Asia and Australia	Y	Y
NI-2	National ISDN—United States	Y	Y
NTT	Japan	Y	Y
QSIG	QSIG	Y	Y
TS014	Australia (Obsolete)	Y	N

The following steps are required to configure a PRI on a Cisco voice gateway:

Step 1 Set the ISDN switch type in global configuration mode using the **isdn switch-type** command:

```
Gateway(config)#isdn switch-type primary-ni
```

Step 2 Configure the controller using the **pri-group** command:

```
Gateway(config)#controller T1 1/0
Gateway(config-controller)pri-group timeslots 1-24
```

Only one PRI group is supported on a T1 or E1. Fractional PRIs are supported. For example, you can configure 12 B-channels. The D-channel is always required. If you do not include the D-channel in the timeslot range, it is added for you.

When you configure a PRI group on a controller, the gateway creates a serial interface. For controller 1/0 shown in the example, this interface is referenced as Serial 1/0:23 for T1s and Serial 1/0:15 for E1s.

Step 3 If necessary, configure the serial interface for network-side support using the **isdn protocol-emulate** command:

```
Gateway(config)#interface serial1/0:23
Gateway(config-if)#isdn protocol-emulate network
```

You can also modify the switch type for each PRI if required using the **isdn switch-type** command under the serial interface.

Step 4 Verify the ISDN status using the **show isdn status** command:

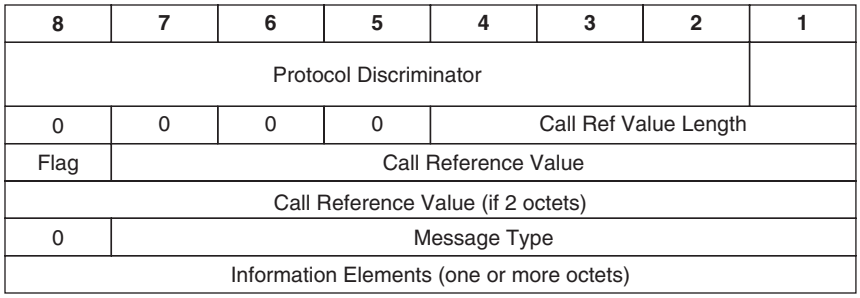
```
Gateway#show isdn status
Global ISDN Switchtype = primary-ni
ISDN Serial1/0:23 interface
        dsl 0, interface ISDN Switchtype = primary-ni
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
```

```
Layer 3 Status:
    0 Active Layer 3 Call(s)
Active dsl 0 CCBs = 0
The Free Channel Mask:  0x80000FFF
Number of L2 Discards = 0, L2 Session ID = 16
Total Allocated ISDN CCBs = 0
```

ISDN Call Signaling

ISDN performs supervisory, address, and informational signaling using Layer 3 messages that are sent on the D-channel. Figure 5-3 shows the ISDN Layer 3 message format.

Figure 5-3 *ISDN Layer 3 Message Format*



- **Protocol Discriminator**—Identifies the protocol type used.
- **Call Ref Value Length**—Indicates the length of the Call Reference Value.
- **Flag**—Set to 0 if the message sender assigned the Call Reference Value. Set to 1 if the message sender did not assign the Call Reference Value.
- **Call Reference Value**—Arbitrary number assigned to identify a call.
- **Message Type**—Defines the function of the message. Message types are described later in this section.
- **Information Elements**—Additional information as required. Information Elements are described in more detail later in this section.

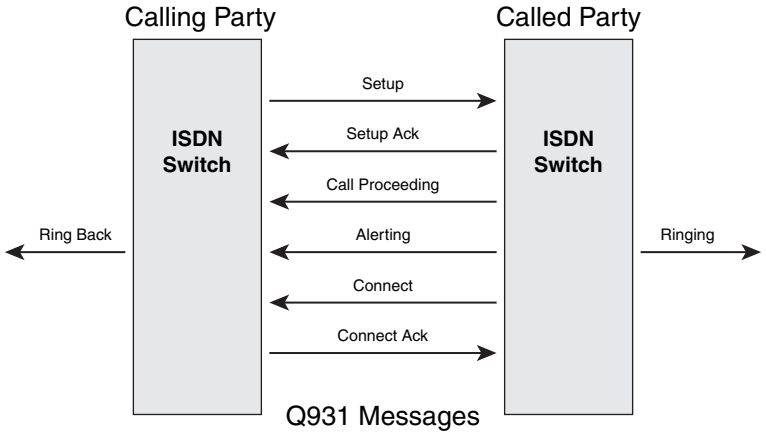
The message type is typically one octet. The first bit is always 0. The next two bits are used to group message types. The last five bits define the message. Table 5-8 lists common message types. The capitalized portion of the message is what appears in a debug of ISDN messages.

Table 5-8 *Common ISDN Message Types*

Call Establishment (000)		**Call Clearing (010)**	
000 00001	ALERTing	010 00101	DISConnect
000 00010	CALL PROCeeding	010 00110	Restart
000 00011	PROGress	010 01110	Restart ACK
000 00101	SETUP	010 01101	RELease
000 01101	SETUP ACK	010 11010	RELease COMplete
000 00111	CONNect	–	–
000 01111	CONN ACK	–	–
Call Information (001)		**Miscellaneous (011)**	
001 00000	USER INFO	011 00000	SEGment
001 00101	SUSPend	011 00010	FACility
001 00001	SUSP REJ	011 01110	NOTIFY
001 01101	SUSP ACK	011 10101	STATUS ENQuiry
001 00110	RESume	011 11001	Congestion
001 00010	RES REJ	011 11011	INFO
001 01110	RES ACK	011 11101	STATUS

Call setup and call clearing message types provide the supervisory signaling. Figure 5-4 illustrates a typical message flow in an ISDN call.

Figure 5-4 *ISDN Message Flow*



Information Elements

The Information Element (IE) field in the message type is used to convey specific information regarding the call. For example, a DISConnect or RELease message includes a CAUSE IE that indicates why the call was terminated.

The common IEs are as follows:

- **Channel ID**—Indicates the channel that is being used. Included in Call Establishment and Call Clearing messages.
- **Cause**—Indicates the reason that a call was terminated or that call setup failed.
- **Display**—Sends information to be displayed, such as calling name or number. Some switch types use Facility IEs for calling name information.
- **Facility**—Invokes supplemental services.
- **Progress**—Used for informational feedback, such as ring or network announcements. For example, "The number you have dialed is not in service."

You can view the ISDN Layer 3 messages by issuing the **debug isdn q931** command. Example 5-5 shows the debug output for a call that is placed from 4085550123 to 12012012002.

Example 5-5 *ISDN Q931 Debug*

```
Gateway#debug isdn q931
debug isdn q931 is              ON.
Gateway#
*May  2 04:07:10.727: ISDN Se1/0:23 Q931: Applying typeplan for sw-type 0xD is 0
x0 0x0, Calling num 4085550123
*May  2 04:07:10.731: ISDN Se1/0:23 Q931: Applying typeplan for sw-type 0xD is 0
x0 0x0, Called num 12012012002
*May  2 04:07:10.731: ISDN Se1/0:23 Q931: TX -> SETUP pd = 8  callref = 0x0022
        Bearer Capability i = 0x9090A2
                Standard = CCITT
                Transer Capability = 3.1kHz Audio
                Transfer Mode = Circuit
                Transfer Rate = 64 kbit/s
        Channel ID i = 0xA98381
                Exclusive, Channel 1
        Progress Ind i = 0x8183 - Origination address is non-ISDN
        Calling Party Number i = 0x80, '408550123'
                Plan:Unknown, Type:Unknown
        Called Party Number i = 0x80, '12012012002'
                Plan:Unknown, Type:Unknown
*May  2 04:07:10.779: ISDN Se1/0:23 Q931: RX <- CALL_PROC pd = 8  callref = 0x80
22
        Channel ID i = 0xA98381
                Exclusive, Channel 1
*May  2 04:07:10.923: ISDN Se1/0:23 Q931: RX <- ALERTING pd = 8  callref = 0x802
2
        Progress Ind i = 0x8088 - In-band info or appropriate now available
*May  2 04:07:19.859: ISDN Se1/0:23 Q931: RX <- CONNECT pd = 8  callref = 0x8022
```

Example 5-5 *ISDN Q931 Debug (Continued)*

```
*May  2 04:07:19.859: ISDN Se1/0:23 Q931: TX -> CONNECT_ACK pd = 8  callref = 0x
0022
*May  2 04:07:25.859: %ISDN-6-CONNECT: Interface Serial1/0:0 is now connected to
 12012012002 unknown
*May  2 04:07:42.855: %ISDN-6-DISCONNECT: Interface Serial1/0:0  disconnected fr
om 12012012002 , call lasted 22 seconds
*May  2 04:07:42.855: ISDN Se1/0:23 Q931: TX -> DISCONNECT pd = 8  callref = 0x0
022
        Cause i = 0x8090 - Normal call clearing
*May  2 04:07:42.871: ISDN Se1/0:23 Q931: RX <- RELEASE pd = 8  callref = 0x8022
*May  2 04:07:42.875: ISDN Se1/0:23 Q931: TX -> RELEASE_COMP pd = 8  callref = 0
x0022
```

Address Types

ISDN includes two identifiers that are used to classify the calling and called numbers. These are the number plan identification (NPI) and the type of number (TON). These identifiers are highlighted in the setup message in Example 5-4. The NPI and TON sometimes determine how a call is routed or how the number is displayed. They can also be used for accounting purposes. Many newer switch types ignore the NPI and TON. The NPI and TON are specified in the third octet of the Calling and Called part IEs. The NPI is specified in bits 1 through 4, and the TON is specified in bits 5 through 7. Table 5-9 lists the defined NPIs, and Table 5-10 lists the defined TONs.

Table 5-9 *NPIs*

NPI Bits	NPI	Related Standard
0 0 0 0	Unknown	—
0 0 0 1	ISDN Telephony	E.164 (described below)
0 0 1 1	Data	X.121
0 1 0 0	Telex	F.69
0 1 0 1	Maritime Mobile	E.210 and E.211
0 1 0 1	Land Mobile	E.212
0 1 1 1	ISDN/Mobile	E214
1 0 0 0	National Standard	—
1 0 0 1	Private	—
1 1 1 1	Reserved for Extension	—

Table 5-10 *TONs*

TON Bits	TON
0 0 0	Unknown
0 0 1	International
0 1 0	National
0 1 1	Network Specific
1 0 0	Subscriber
1 1 0	Abbreviated
1 1 1	Reserved for extension

ITU Recommendation E.164, "Numbering Plan for the ISDN Era," defines specifics on how to build a numbering plan to allow interoperability between the numerous public networks. E.164 specifies a format for an international ISDN number, which is variable length arranged in specific fields, as follows:

- **Country Code (CC)**—The country code is a one, two, or three-digit code representing a specific country or region.
- **National (Significant) Number (N(S)N)**—The N(S)N is the number used to select the destination subscriber. The N(S)N is further defined as containing the following fields:
  - **National Destination Code (NDC)**—The NDC is variable in length and contains Destination Network (DN) or Trunk Codes (TC) to indicate how to route a call. This is commonly called an area code.
  - **Subscriber Number (SN)**—SN is variable in length and is assigned to end subscribers.

NFAS

Non-Facility Associated Signaling (NFAS) is an ISDN feature that allows you to share one D-channel for multiple PRI lines. This allows you to use all of the DS-0s on some PRIs as B-channels and increase the amount of user data available. For example, if you have five PRI lines configured for NFAS with only the first PRI providing a D-channel, you will gain four additional B-channels across the bundle. NFAS also allows for a backup D-channel if the primary channel fails. The bundle of T1s configured in an NFAS group and signaled by a common D-channel is typically referred to as a *trunk group*.

QSIG

Q Signaling (QSIG) is an alternative to Q.931 that is used for interconnection in private integrated services network exchange (PINX), consisting of PBXs, key systems, and CallManager. It is an ISDN variant based on Q.931 that is used worldwide for the interconnection of private telephony devices. QSIG consists of three sublayers: Basic Call, Generic Function, and Supplementary Services. Basic Call is an extension of Q.931 that is used to provide call setup, maintenance, and clearing support. Generic Function enables transparent transmission of QSIG facility messages to allow supplementary services and network features. Supplementary Services provide additional functions.

One of the most important qualities of QSIG is its flexibility as a signaling agent. Feature transparency allows for features to be carried between two different QSIG endpoints even if intermediate switching devices do not support the feature set. This allows feature support across multiple vendors and the development of new proprietary features without concerns of interoperability. Flexibility in the number plan, network topology, networking transport, and application allows QSIG to be widely deployed in many environments. QSIG has also been adopted by all the major PBX manufacturers to ensure consistent development.

Echo Cancellation

Echo is an intrusive condition that has existed on telephone networks since their inception. The human ear is sensitive to fluctuations in voice, and the inherent delay that is involved in telecommunications can cause echo on the line to disrupt the user experience. The primary source of echo in traditional telephone networks is line-side echo caused by 2-wire to 4-wire hybrids. These hybrids convert the 2-wire circuit coming from the subscriber loop to the 4-wire trunk circuit. A mismatch in impedance in these devices causes a portion of the “talk” energy to be reflected back onto the receive side. If the delay in this return voice energy is sufficient, the speaker hears his own voice disruptively echoing on the line. Acoustic echo occurs when the microphone of the user picks up the voice of the user multiple times: once when the user speaks, and then again as the voice energy is reflected back from the environment and transmitted by the microphone. Acoustic echo is caused by a faulty device with poor acoustic isolation between the speaker and microphone. Some headsets and speaker phones cause acoustic echo. In modern telecommunications, the analog to digital conversion and the use of bandwidth-saving compression techniques can increase the problems that are associated with echo.

You can control echo by using several mechanisms: balancing impedance on hybrids, inserting loss into the path of the return signal, using echo cancellers, and minimizing the delay. Efforts are made to balance the impedance in a voice network, but imbalance, and the resulting echo, is inevitable. Because of the need to maintain levels and the inevitable delay inserted due to digital processing, the most practical method of controlling echo in networks is the use of an echo cancellation device. Echo cancellers work by using digital signal processors (DSP) and application-specific integrated circuits (ASIC) to sample the

signal and model the signal characteristics and echo in the path. The canceller then compares the returning voice path with the model and dynamically removes the echo.

You can also effectively use cancellers to mitigate the amount of intrusive background noise by adding attenuation and effectively lowering the dB levels of the background noise. This is done by the nonlinear processor (NLP) of the echo canceller. It samples the background noise to learn the frequencies and then uses this information to drop the dB levels. Another important function of the NLP is to generate comfort noise in periods of the conversation when neither party is speaking. This white noise on the line is necessary, because if the line is entirely clean, the speakers believe that the call has been disconnected.

Review Questions

1. Which port type should you use to connect to a 2-wire analog service connecting to the PSTN for both inbound and outbound calling?
2. What types of signaling are required on a voice circuit?
3. What is the difference between CAS and CCS on an E1 circuit?
4. What is the difference between SF and ESF?
5. What two types of echo are possible on a voice circuit?
6. Which signaling type supports ANI on T1 CAS circuits?
7. Which line signaling method should you use on an E1 r2 satellite link?
8. What component of an ISDN message is used to carry information about the call?