Troubleshooting SIP Interfaces to the IP Network

The Cisco Session Initiation Protocol (SIP) implementation enables Cisco access platforms to signal the setup of voice and multimedia calls over IP networks.

SIP is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints. SIP is an alternative protocol developed by the Internet Engineering Task Force (IETF) for multimedia conferencing over IP. SIP features are compliant with IETF RFC 2543, SIP: Session Initiation Protocol, published in March 1999. You can view RFC 2543 at http://www.ietf.org/rfc/rfc2543.txt.

Like other Voice-over-IP protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

This chapter provides procedural and reference information that you can use to determine and resolve problems with SIP interfaces to the IP network.

Note Call flows can help in troubleshooting SIP problems. SIP call flow information can be found in the *Session Initiation Protocol Gateway Call Flows* document.

This chapter contains the following information:

- Troubleshooting the Cisco SIP IP Phone 7960, page 367
- Troubleshooting the Cisco SIP Gateway, page 371
- Troubleshooting the Cisco SIP Proxy Server, page 386
- SIP Messages and Methods, page 388

Troubleshooting the Cisco SIP IP Phone 7960

This section describes troubleshooting features and tips for the Cisco SIP IP phone 7960.

Troubleshooting Features

The following is a list of features on the Cisco SIP IP phone that you can use for troubleshooting:

- Settings button to Network Configuration soft key—Use to view or modify the network configuration of the phone.

- Settings button to SIP Configuration soft key—Use to view or modify a phone's SIP settings.
- Settings button to Status—Display configuration or initialization errors.
- Call messages on LED screen—Display basic SIP message flows.
- Pressing ***i*** or ***?*** key twice during a call—Displays real-time transferring and receiving call statistics. This option is recommended for troubleshooting voice-quality issues.

In addition to the features listed above, the EIA/TIA-232 (RS-232) port located on the back of the Cisco SIP IP phone 7960 is a console port and can be used to gather debug information.

The EIA/TIA-232 port is password-protected and requires a custom RJ-11-to-RJ-45 cable.

Note For a PC connection, the RJ-45 connection needs a DB-9 female DTE adapter or an RJ-45 crossover cable for an octal async connection. You must enter the password "cisco" must be entered to enable any output to be seen via the EIA/TIA-232 port. The connection baud rate, parity, start bits, and stop bits are 9600, N, 8, and 1.

To use the console port, use a RJ-11-to-RJ-45 custom cable to connect the EIA/TIA-232 port to a PC.

Table 48 lists the RJ-11-to-RJ-45 cable pinouts.

Table 48 RJ-11-to-RJ-45 Pinouts

RJ-11 or RJ-12	RJ-45
2	6
3	4
4	3

To connect the console port, complete the following tasks:

Step 1 Insert the RJ-11 end of the rolled cable into the EIA/TIA-232 port on the back of the phone.

Step 2 Use an RJ-45-to-DB-9 female DTE adapter (labeled TERMINAL) to connect the console port to a PC running terminal emulation software.

Step 3 Insert the RJ-45 end of the rollover cable into the DTE adapter.

Step 4 From the console terminal, start the terminal emulation program.

Step 5 Type "cisco". A prompt is displayed.

Step 6 At the prompt, you can issue the following commands to assist you in troubleshooting and debugging the phone:

- **debug error**—Displays error messages that are occurring in the call flow process
- **debug sip-message**—Enables you to view a text display of a call flow

Troubleshooting Tips

This section provides tips for resolving the following Cisco SIP IP phone problems:

- Cisco SIP IP Phone Is Unprovisioned or Is Unable to Obtain an IP Address, page 369

- Cisco SIP IP Phone Does Not Register With the SIP Proxy or SIP Registrar Server, page 369
- Outbound Calls Cannot Be Placed from a Cisco SIP IP Phone, page 370
- Inbound Calls Cannot Be Received on a Cisco SIP IP Phone, page 370
- Poor Voice Quality on the Cisco SIP IP Phone, page 370
- DTMF Digits Do Not Function Properly, page 371
- Cisco SIP IP Phones Do Not Work When Plugged into a Line-Powered Switch, page 371
- Call Transfer Does Not Work Correctly, page 371
- Some SIP Messages are Retransmitted Too Often, page 371

For more information about Cisco SIP IP phones, see the *Cisco IP Phone Administrator Guides for SIP*.

Cisco SIP IP Phone Is Unprovisioned or Is Unable to Obtain an IP Address

To determine why a phone is unprovisioned or unable to obtain an IP address, perform the following tasks as necessary:

- If using TFTP to download configuration files, verify that the SIPDefault.cnf file and the phone-specific configuration file (SIP*mac*.cnf where *mac* is the MAC address of the phone) exist and are configured correctly.
- Verify that the TFTP server is working properly.
- Verify that the Cisco SIP IP phone network configuration parameters are properly configured and the phone is obtaining the proper IP addressing information (IP address, subnet mask, default gateway, TFTP server, and so forth.)
- Press the **Settings** button, select **Status**, and then **Status Messages** to view messages for missing files or other errors.
- If the DHCP server has an IP subnet mask that is different from the one for the Cisco SIP IP phone, verify that "ip helper-address" address is enabled on the local router.
- Verify that the Cisco SIP IP phone software image (P0S3*xxyy*.bin where *xx* is the version number and *yy* is the subversion number) was downloaded from the Cisco website in binary format.

Cisco SIP IP Phone Does Not Register With the SIP Proxy or SIP Registrar Server

To determine why a phone does not register with a SIP proxy or SIP registrar server, perform the following tasks as necessary:

Note The character "x" displayed to the right of a line icon indicates that registration has failed.

- Verify that phone registration with a proxy server is enabled (via the proxy_register parameter in the configuration files). By default, registration during initialization is disabled.
- Verify that the IP address (proxy1_address parameter) of the primary SIP proxy server to be used by the phones is valid.
- If a Fully Qualified Domain Name (FQDN) is specified in the proxy1_address parameter, verify that the DNS server is configured to resolve the FQDN as a DNS A-record type.
- Verify that the Cisco SIP proxy server has been configured to require authentication. If it has, ensure that an authentication name and password have been defined in the Cisco SIP IP phone-specific configuration file (through the use of the linex_authname and linex_password parameters).

- The Cisco SIP IP phone currently supports the HTTP Digest authentication method. Verify that the authentication method required by the Cisco SIP proxy server (through the use of the AuthScheme directive in the sipd.conf file) is HTTP Digest.
- Verify that a registration request hasn't expired. By default, Cisco SIP IP phones reregister every 3600 seconds, but this value can be modified through the use of the time_reqister_expires parameter.

Outbound Calls Cannot Be Placed from a Cisco SIP IP Phone

If a call cannot be placed from a Cisco SIP IP phone, perform the following tasks as necessary:

- Verify that the Cisco SIP IP phone network configuration IP address parameters have been correctly entered or received from a DHCP server.
- Verify that the Cisco SIP proxy server used by the phone is working properly.
- Verify that the Cisco SIP proxy server is correctly configured for routes or registrations to the remote destination.
- Verify that the remote SIP device is available.
- Verify that a dial plan has been defined in the dialplan.xml file and if so, that the configuration is correct. This file should have been downloaded from CCO to the root directory of your TFTP server.
- Determine which error tones are being received and map those tones to the messages displayed on the phone's LCD (SIP 4xx messages, and so forth.)

Inbound Calls Cannot Be Received on a Cisco SIP IP Phone

If inbound calls cannot be received on a Cisco SIP IP phone, perform the following tasks as necessary:

- Verify that the line (user portion) was defined in the Request-URI or the SIP INVITE request. The Cisco SIP IP phone requires this information to determine the proper line to ring.
- Verify that the Request-URI is sent to port 5060 of the phone's IP address. The phone listens on UDP port 5060.
- Verify that the Cisco SIP IP phone is registered with the local proxy server.

Poor Voice Quality on the Cisco SIP IP Phone

If a call's voice quality is compromised on the Cisco SIP IP phone, perform the following tasks as necessary:

- Check the network path for errors, packet drops, loss, loops, and so forth.
- Verify that the ToS level for the media stream being used has been correctly set (through the tos_media parameter in the configuration file).
- Verify that the Cisco SIP IP phone is plugged into a switch rather than a hub to avoid excessive collisions and packet loss.
- Ensure that there is enough bandwidth on the network for the selected codec (especially for calls over a WAN).
- Press the **i** or **?** button twice on the phone during the call to view realtime transferring and receiving call statistics.
- Determine whether the problem occurs with the handset, headset, or speaker phone, or with all of them.

DTMF Digits Do Not Function Properly

If DTMF digits are not functioning properly, perform the following tasks as necessary:

- If out-of-bound signaling through the AVT tone method has been enabled (through the dtmf_outofband configuration file parameter), verify that the remote device supports AVT tones (as defined in RFC 2833). If AVT tones have been enabled and the remote device does not support AVT tones, check for packet loss in the end-to-end path.
- Find out which codec is being used. Lower bandwidth codecs yield poorer results if AVT tones are not supported because the DTMF digits are carried in audio.
- Verify the length of the tones being created. The tone must have a minimum signal duration of 40 ms with signaling velocity (tone and pause) of no less than 93 ms (as defined in RFC 2833).

Cisco SIP IP Phones Do Not Work When Plugged into a Line-Powered Switch

If the Cisco SIP IP phones do not work when plugged into a line-powered switch, perform the following tasks:

- Verify that the phone is running version 2.0 or higher of the Cisco SIP IP Phone software. (Line-powered support was not available in version 1.0.)
- Verify that the network media type Network Settings parameter is set to auto-negotiation (auto).

Call Transfer Does Not Work Correctly

If call transfer does not work correctly, verify that the remote SIP device that is sending the call is using the SIP BYE/Also: method (as defined in Internet draft sip-cc-01.txt.)

Some SIP Messages are Retransmitted Too Often

The Cisco SIP IP phone has several timers (INVITE request retries, BYE request retries, etc.) that can be configured using the sip_invite_retx and sip_retx configuration file parameters. In most networks, the default values work fine, however, conditions such as network delay, slower-processing proxy servers, and packet loss might require that the timers be adjusted. If some SIP messages appear to be retransmitted too often, adjust these parameters.

Troubleshooting the Cisco SIP Gateway

This section provides tips for resolving the following Cisco SIP gateway problems:

- Unable to Make Outbound Calls from the Cisco SIP Gateway to a SIP Endpoint, page 372
- Unable to Make Inbound Calls to a PSTN Through a Cisco SIP Gateway, page 372
- Calls to a PSTN via the Cisco SIP Gateway Fail with a “400 Bad Request” Response, page 373
- Voice Quality Is Compromised on Calls Through or From the Cisco SIP Gateway, page 376
- Some SIP Messages Are Retransmitted Too Often, page 376
- Call Transfer Does Not Work Correctly, page 376
- Troubleshooting Commands, page 377

Unable to Make Outbound Calls from the Cisco SIP Gateway to a SIP Endpoint

If a call cannot be placed from the Cisco SIP gateway, perform the following tasks as necessary:

- Verify that the voice ports are properly configured and enabled for the PSTN-side signaling protocol.
- Verify that there is a valid VoIP dial peer configured that meets the following requirements:
  - Matches the required destination pattern
  - Is SIP-enabled (through the **session protocol sipv2** command)
  - Has the correct dial peer session target defined (through the **session target** ***sip-server*** command
  - Has the codec correctly defined
- Using the **ping** command, verify that the SIP gateway can communicate through IP with the SIP proxy or remote SIP device.
- If the SIP proxy server is defined through the use of a FQDN, verify that the DNS server is correctly configured to resolve that address using a DNS SRV record.
- Ensure that the time zone format configured on the SIP gateway is GMT.
- Check the **debug ccsip all | calls | error | events | messages | states** command output for protocol errors.

Unable to Make Inbound Calls to a PSTN Through a Cisco SIP Gateway

If inbound calls to a PSTN cannot be made through the Cisco SIP gateway, perform the following tasks as necessary to determine the cause:

- Verify that the voice ports are correctly configured and enabled for the PSTN-side signaling protocol.
- Verify that a valid POTS dial peer is configured and that it matches the required destination pattern.
- Using the **ping** command, verify that the Cisco SIP gateway can communicate with the SIP proxy server or remote SIP device through IP.
- If the inbound call has any hostnames defined as a FQDN, ensure that the proper DNS configuration is enabled on the Cisco SIP gateway (to resolve the hosts).
- View the **debug ccsip all | calls | error | events | messages | states** command output for protocol errors.

Calls to a PSTN via the Cisco SIP Gateway Fail with a "400 Bad Request" Response

If the Cisco SIP gateway does not like part of a SIP message (header or SDP), the call attempt fails with a "400 Bad Request" response.

To determine whether the call failed because of a SIP header error, issue the **debug ccsip** command that displays information on the error message, or verify that the required SIP header elements exist as defined in RFC 2543. SIP header fields are shown in Table 49.

Table 49 SIP Header Fields

Header Field	Definition
Call-ID	The Call-ID general-header field uniquely identifies a specific invitation or all registrations of a specific client. Note that a single multimedia conference can give rise to several calls with different Call-IDs. For example, if a user invites a single individual several times to the same (long-running) conference.
Contact	The Contact general-header field MUST appear in INVITE and REGISTER requests and in 200 responses. It can appear in ACK, and in other 1xx, 2xx, 3xx, 485 responses. In general, it provides a URL where the user can be reached for further communications.
Content-Length	The Content-Length entity-header field indicates the size of the message-body, in decimal number of octets, sent to the recipient.
Content-Type	The Content-Type entity-header field indicates the media type of the message-body sent to the recipient.
Cseq	Users MUST add the CSeq (command sequence) general-header field to every request. A CSeq header field in a request contains the request method and a single decimal sequence number chosen by the requesting client, unique within a single value of Call-ID. The sequence number MUST be expressed as a 32-bit unsigned integer. The initial value of the sequence number is arbitrary, but MUST be less than 2**31. Consecutive requests that differ in request method, headers, or body, but have the same Call-ID MUST contain strictly monotonically increasing and contiguous sequence numbers; sequence numbers do not wrap around. Retransmissions of the same request carry the same sequence number, but an INVITE with a different message body or different header fields (a "re-invitation") acquires a new, higher sequence number. A server MUST echo the CSeq value from the request in its response. If the Method value is missing in the received CSeq header field, the server fills it in appropriately.
Date	Date is a general-header field. Its syntax is: SIP-date = rfc1123-date Note that unlike HTTP/1.1, SIP only supports the most recent RFC 1123 [29] formatting for dates.
Diversion	**Note** Currently gateway uses Diversion header in initial outgoing messages.

Table 49 SIP Header Fields (continued)

Header Field	Definition
Expires	The Expires entity-header field gives the date and time after which the message content expires. This header field is currently defined only for the REGISTER and INVITE methods. For REGISTER, it is a request and response-header field. In a REGISTER request, the client indicates how long it wants the registration to be valid. In the response, the server indicates the earliest expiration time of all registrations. The server MAYchoose a shorter time interval than that requested by the client, but SHOULD NOT choose a longer one.
From	Requests and responses MUST contain a From general-header field, indicating the initiator of the request. The From field MUST contain a tag. The server copies the From header field from the request to the response. The optional "display-name" is meant to be rendered by a human-user interface. A system SHOULD use the display name "Anonymous" if the identity of the client is to remain hidden. The SIP-URL MUST NOT contain the "transport-param", "maddr-param", "ttl-param", or "headers"elements. A server that receives a SIP-URL with these elements removes them before further processing.
Max-Forwards	The Max-Forwards request-header field may be used with any SIP method to limit the number of proxies or gateways that can forward the request to the next downstream server. This can also be useful when the client is attempting to trace a request chain which appears to be failing or looping in mid chain. The Max-Forwards value is a decimal integer indicating the remaining number of times this request message is allowed to be forwarded. Each proxy or gateway recipient of a request containing a Max-Forwards header field MUST check and update its value before forwarding the request. If the received value is zero (0), the recipient MUST NOT forward the request. Instead, for the OPTIONS and REGISTER methods, it MUST respond as the final recipient. For all other methods, the server returns 483 (too many hops). If the received Max-Forwards value is greater than zero, then the forwarded message MUST contain an updated Max-Forwards field with a value decremented by one (1).
Require	The Require request-header field is used by clients to tell useragent servers about options that the client expects the server to support in order to properly process the request. If a server does not understand the option, it MUST respond by returning status code 420 (bad extension) and list those options it does not understand in the Unsupported header.
Server	The Server response-header field contains information about the software used by the user agent server to handle the request.
Timestamp	The timestamp general-header field describes when the client sent the request to the server. The value of the timestamp is of significance only to the client and it MAY use any time scale. The server MUST echo the exact same value and MAY, if it has accurate information about this, add a floating point number indicating the number of seconds that have elapsed since receiving the request. The timestamp is used by the client to compute the round-trip time to the server so that it can adjust the time out value for retransmissions.

Table 49 SIP Header Fields (continued)

Header Field	Definition
To	The To general-header field specifies recipient of the request, with the same SIP URL syntax as the From field. Requests and responses MUST contain a To general-header field, indicating the desired recipient of the request. The optional “display-name”is meant to be rendered by a human-user interface. The UAS or redirect server service processing a request MUST always add a tag to To-header.
User-Agent	The User-Agent general-header field contains information about the client user agent originating the request.
Via	The Via field indicates the path taken by the request so far. This prevents request looping and ensures replies take the same path as the requests, which assists in firewall traversal and other unusual routing situations. When the UAC creates a request, it MUST insert a Via into that request.

Possible SDP-related errors are as follows:

- SDP_ERR_INFO_UNAVAIL
- SDP_ERR_VERSINFO_INVALID
- SDP_ERR_CONNINFO_IN
- SDP_ERR_CONNINFO_IP
- SDP_ERR_CONNINFO_NULL
- SDP_ERR_CONNINFO_INVALID
- SDP_ERR_MEDIAINFO_TYPE
- SDP_ERR_MEDIAINFO_INVALID
- SDP_ERR_MEDIAINFO_NULL
- SDP_ERR_OWNERINFO_NULL
- SDP_ERR_OWNERINFO_SESSID_NULL
- SDP_ERR_OWNERINFO_SESSID_INVALID
- SDP_ERR_OWNERINFO_VERSID_NULL
- SDP_ERR_OWNERINFO_VERSID_INVALID
- SDP_ERR_OWNERINFO_IN
- SDP_ERR_OWNERINFO_IP
- SDP_ERR_TIMEINFO_ST_NULL
- SDP_ERR_TIMEINFO_ET_NULL
- SDP_ERR_TIMEINFO_ST_INVALID
- SDP_ERR_TIMEINFO_ET_INVALID
- SDP_ERR_ATTRINFO_INVALID
- SDP_ERR_ATTRINFO_NULL
- SDP_ERR_AUDIO_MEDIA_UNAVAIL
- SDP_ERR_MEDIAINFO_PORT_INVALID

- SDP_ERR_MEDIAINFO_MALLOC_FAIL
- SDP_ERR_ATTRINFO_MALLOC_FAIL

Possible CheckRequest errors are as follows:

- CHK_REQ_FAIL_MISMATCH_CSEQ
- CHK_REQ_FAIL_INVALID_CSEQ
- CHK_REQ_FAIL_FROM_TO
- CHK_REQ_FAIL_VERSION
- CHK_REQ_FAIL_METHOD_UNKNOWN
- CHK_REQ_FAIL_REQUIRE_UNSUPPORTED
- CHK_REQ_FAIL_CONTACT_MISSING
- CHK_REQ_FAIL_MISMATCH_CALLID
- CHK_REQ_FAIL_MALFORMED_CONTACT
- CHK_REQ_FAIL_MALFORMED_RECORD_ROUTE

Voice Quality Is Compromised on Calls Through or From the Cisco SIP Gateway

If the voice quality on calls through or from the Cisco SIP gateway is compromised, perform the following tasks as necessary to determine the cause:

- Check the network path for errors, packet drops, loss, loops, and so forth.
- Verify that the TOS bits have been correctly set in the VoIP dial peer through the use of the **ip precedence** command.
- To minimize excessive collisions and packet loss, connect the Cisco SIP gateway to a switch rather than a hub.
- Verify that enough bandwidth exists on the network for the configured codec (especially for calls over a WAN).
- View the output of the show interface command for packet drops. View the output of the s**how voice dsp** command for DSP-related issues.
- Determine whether errors exist on the voice ports that could be causing the problems.

Some SIP Messages Are Retransmitted Too Often

The Cisco SIP gateway has SIP timers (INVITE request retries, BYE request retries) configured under the SIP UA through the use of the **timers trying number**, **timers expires time**, and **retry invite number** commands. In most networks, the default values work well, but conditions such as network delay, slower-processing proxy servers, and packet loss might require that the timers be adjusted. If some SIP messages appear to be retransmitted too often, adjust these parameters.

Call Transfer Does Not Work Correctly

If call transfer does not work correctly, perform the following tasks to determine the cause:

- Verify that the application session is defined on the VoIP and POTS dial peers.

- Verify that the remote SIP device that is sending the call through the use of the SIP BYE/Also: method (as defined in Internet draft sip-cc-01.txt).
- Use the **debug voip ccapi inout** command to verify that a dial peer that has application session defined is matched. The application used after the BYE request is sent should be "session" instead of "SESSION."

Troubleshooting Commands

There are several debug commands that are useful for troubleshooting problems with SIP, as follows:

- **debug ccsip all**
- **debug ccsip calls**
- **debug ccsip error**
- **debug ccsip events**
- **debug ccsip info**
- **debug ccsip media**
- **debug ccsip messages**
- **debug ccsip preauth**
- **debug ccsip states**

Details about these commands can be found in the *Cisco IOS Debug Command Reference*.

Note The output from these commands can be filtered. For more information, see the "SIP Debug Output Filtering Support" section on page 83.

The following show and debug commands shown can be used to troubleshoot the Cisco SIP gateway:

- **show sip status**—Displays the SIP user agent listener status.

```
sip-2600a# show sip status

SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP max-forwards : 6
```

- **show sip statistics**—Displays SIP user agent statistics.

```
router# show sip statistics

SIP Response Statistics (Inbound/Outbound)
Informational:
Trying 3/0, Ringing 3/0,
Forwarded 0/0, Queued 0/0,
SessionProgress 0/0
Success:
OkInvite 3/0, OkBye 2/0,
OkCancel 0/0, OkOptions 0/0
Redirection (Inbound only):
MultipleChoice 0, MovedPermanently 0,
MovedTemporarily 0, SeeOther 0,
UseProxy 0, AlternateService 0
Client Error:
BadRequest 0/3, Unauthorized 0/0,
```

```
PaymentRequired 0/0, Forbidden 0/0,
NotFound 0/0, MethodNotAllowed 0/0,
NotAcceptable 0/0, ProxyAuthReqd 0/0,
ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
LengthRequired 0/0, ReqEntityTooLarge 0/0,
ReqURITooLarge 0/0, UnsupportedMediaType 0/0,
BadExtension 0/0, TempNotAvailable 0/0,
CallLegNonExistent 0/0, LoopDetected 0/0,
TooManyHops 0/0, AddrIncomplete 0/0,
Ambiguous 0/0, BusyHere 0/0

Server Error:
InternalError 0/0, NotImplemented 0/0,
BadGateway 0/0, ServiceUnavail 0/0,
GatewayTimeout 0/0, BadSipVer 0/0

Global Failure:
BusyEverywhere 0/0, Decline 0/0,
NotExistAnywhere 0/0, NotAcceptable 0/0

SIP Total Traffic Statistics (Inbound/Outbound)
Invite 3/7, Ack 2/1, Bye 0/2,
Cancel 0/0, Options 0/0
Retry Statistics
Invite 2, Bye 0, Cancel 0, Response 1
```

- **debug ccsip**—Displays the different **debug ccsip** commands.

```
router# 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
messages Enable CCSIP SPI messages debugging trace
states   Enable CCSIP SPI states debugging trace
```

The following is a sample of debug output from one side of a call:

```
Router1# debug ccsip all
All SIP call tracing enabled
Router1#
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_NONE, SUBSTATE_NONE)  to
(STATE_IDLE, SUBSTATE_NONE)
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_CC_CALL_SETUP
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_call_setup
*Mar  6 14:10:42:  act_idle_call_setup:Not using Voice Class Codec

*Mar  6 14:10:42: act_idle_call_setup: preferred_codec set[0] type :g711ulaw bytes:
160
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_NONE)  to
(STATE_IDLE, SUBSTATE_CONNECTING)
*Mar  6 14:10:42: REQUEST CONNECTION TO IP:166.34.245.231 PORT:5060

*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to
(STATE_IDLE, SUBSTATE_CONNECTING)
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_connection_created
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_idle_connection_created: Connid(1) created
to 166.34.245.231:5060, local_port 54113
*Mar  6 14:10:42: sipSPIAddLocalContact
*Mar  6 14:10:42:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_method
```

```
*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to
(STATE_SENT_INVITE, SUBSTATE_NONE)
*Mar  6 14:10:42: Sent:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Cisco-Guid: 2881152943-2184249548-0-483039712
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427042
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0

*Mar  6 14:10:42: Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0


*Mar  6 14:10:42: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.231:5060
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:42:  Roundtrip delay 4 milliseconds for method INVITE

*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_SENT_INVITE, SUBSTATE_NONE)
to (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
*Mar  6 14:10:42: Received:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
```

```
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0

*Mar  6 14:10:42: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.231:5060
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session
description
*Mar  6 14:10:42: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:42:  Roundtrip delay 8 milliseconds for method INVITE

*Mar  6 14:10:42: HandleSIP1xxRinging: SDP MediaTypes negotiation successful!
Negotiated Codec      : g711ulaw , bytes :160
Inband Alerting       : 0

*Mar  6 14:10:42: 0x624CFEF8 : State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_PROCEEDING)  to (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_ALERTING)
*Mar  6 14:10:46: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0

*Mar  6 14:10:46: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.231:5060
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  act_recdproc_new_message
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPICheckResponse : Updating session
description
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:46:  Roundtrip delay 3536 milliseconds for method INVITE

*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  act_recdproc_new_message: SDP MediaTypes
negotiation successful!
Negotiated Codec      : g711ulaw , bytes :160

*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sipSPIReconnectConnection
*Mar  6 14:10:46:  Queued event from SIP SPI : SIPSPI_EV_RECONNECT_CONNECTION
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  recv_200_OK_for_invite
*Mar  6 14:10:46:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:10:46: 0x624CFEF8 : State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_ALERTING)  to (STATE_ACTIVE, SUBSTATE_NONE)
*Mar  6 14:10:46: The Call Setup Information is :

        Call Control Block (CCB) : 0x624CFEF8
```

```
         State of The Call        : STATE_ACTIVE
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
         Source IP Port    (Media): 20208
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20038
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231

*Mar  6 14:10:46: HandleUdpReconnection: Udp socket connected for fd: 1 with
166.34.245.231:5060
*Mar  6 14:10:46: Sent:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 137
CSeq: 101 ACK

v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0

*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  6 14:10:46: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  6 14:10:46: ccsip_caps_ind: set DSP for dtmf-relay =
CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  6 14:10:46: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  6 14:10:50: Received:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:36:44 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612207
CSeq: 101 BYE
Content-Length: 0

*Mar  6 14:10:50: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.231:54835
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  act_active_new_message
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sact_active_new_message_request
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  6 14:10:50:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sipSPIInitiateCallDisconnect : Initiate call
disconnect(16) for outgoing call
*Mar  6 14:10:50: 0x624CFEF8 : State change from (STATE_ACTIVE, SUBSTATE_NONE)  to
(STATE_DISCONNECTING, SUBSTATE_NONE)
*Mar  6 14:10:50: Sent:
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Sat, 06 Mar 1993 19:10:50 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612207
Content-Length: 0
CSeq: 101 BYE

*Mar  6 14:10:50:  Queued event From SIP SPI to CCAPI/DNS :
SIPSPI_EV_CC_CALL_DISCONNECT
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  act_disconnecting_disconnect
*Mar  6 14:10:50: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  6 14:10:50:  Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
*Mar  6 14:10:50: CLOSE CONNECTION TO CONNID:1

*Mar  6 14:10:50: sipSPIIcpifUpdate :CallState: 4 Playout: 1755 DiscTime:48305031
ConnTime 48304651

*Mar  6 14:10:50: 0x624CFEF8 : State change from (STATE_DISCONNECTING, SUBSTATE_NONE)
to (STATE_DEAD, SUBSTATE_NONE)
*Mar  6 14:10:50: The Call Setup Information is :

        Call Control Block (CCB) : 0x624CFEF8
         State of The Call        : STATE_DEAD
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.230
         Source IP Port    (Media): 20208
         Destn  IP Address (Media): 166.34.245.231
         Destn  IP Port    (Media): 20038
         Destn SIP Addr (Control) : 166.34.245.231
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.231

*Mar  6 14:10:50:

        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200

*Mar  6 14:10:50: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote
port: 5060
Router1#
```

The debug output is as follows from the other side of the call:

```
3660-2# debug ccsip all
All SIP call tracing enabled
3660-2#
*Mar  8 17:36:40: Received:
INVITE sip:3660210@166.34.245.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Cisco-Guid: 2881152943-2184249548-0-483039712
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
```

```
Timestamp: 731427042
Contact: <sip:3660110@166.34.245.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0

*Mar  8 17:36:40: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.230:54113
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sipSPISipIncomingCall
*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_NONE, SUBSTATE_NONE)  to
(STATE_IDLE, SUBSTATE_NONE)
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_idle_new_message
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sact_idle_new_message_invite
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:40:  sact_idle_new_message_invite:Not Using Voice Class Codec

*Mar  8 17:36:40: sact_idle_new_message_invite: Preferred codec[0] type: g711ulaw
Bytes :160
*Mar  8 17:36:40: sact_idle_new_message_invite: Media Negotiation successful for an
incoming call

*Mar  8 17:36:40: sact_idle_new_message_invite: Negotiated Codec      : g711ulaw,
bytes :160
Preferred Codec       : g711ulaw, bytes :160

*Mar  8 17:36:40:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:40: Num of Contact Locations 1 3660110 166.34.245.230 5060

*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_IDLE, SUBSTATE_NONE)  to
(STATE_RECD_INVITE, SUBSTATE_RECD_INVITE_CALL_SETUP)
*Mar  8 17:36:40: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0

*Mar  8 17:36:40:  Queued event From SIP SPI to CCAPI/DNS :
SIPSPI_EV_CC_CALL_PROCEEDING
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_recdinvite_proceeding
*Mar  8 17:36:40:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_ALERTING
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  ccsip_caps_ind
*Mar  8 17:36:40: ccsip_caps_ind: codec(negotiated) = 5(Bytes 160)
*Mar  8 17:36:40: ccsip_caps_ind: Load DSP with codec (5) g711ulaw, Bytes=160
*Mar  8 17:36:40: ccsip_caps_ind: set DSP for dtmf-relay =
CC_CAP_DTMF_RELAY_INBAND_VOICE
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  ccsip_caps_ack
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  act_recdinvite_alerting
*Mar  8 17:36:40:  180 Ringing with SDP - not likely

*Mar  8 17:36:40:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
```

```
*Mar  8 17:36:40: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:40: 0x624D8CCC : State change from (STATE_RECD_INVITE,
SUBSTATE_RECD_INVITE_CALL_SETUP)  to (STATE_SENT_ALERTING, SUBSTATE_NONE)
*Mar  8 17:36:40: Sent:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0

*Mar  8 17:36:44:  Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_CONNECT
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  act_sentalert_connect
*Mar  8 17:36:44: sipSPIAddLocalContact
*Mar  8 17:36:44:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:44: 0x624D8CCC : State change from (STATE_SENT_ALERTING, SUBSTATE_NONE)
to (STATE_SENT_SUCCESS, SUBSTATE_NONE)
*Mar  8 17:36:44: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Mon, 08 Mar 1993 22:36:40 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Timestamp: 731427042
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@166.34.245.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 137

v=0
o=CiscoSystemsSIP-GW-UserAgent 969 7889 IN IP4 166.34.245.231
s=SIP Call
t=0 0
c=IN IP4 166.34.245.231
m=audio 20038 RTP/AVP 0

*Mar  8 17:36:44: Received:
ACK sip:3660210@166.34.245.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.230:54113
From: "3660110" <sip:3660110@166.34.245.230>
To: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
Date: Sat, 06 Mar 1993 19:10:42 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 137
CSeq: 101 ACK

v=0
```

```
o=CiscoSystemsSIP-GW-UserAgent 1212 283 IN IP4 166.34.245.230
s=SIP Call
t=0 0
c=IN IP4 166.34.245.230
m=audio 20208 RTP/AVP 0

*Mar  8 17:36:44: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.230:54113
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  act_sentsucc_new_message
*Mar  8 17:36:44: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:44: 0x624D8CCC : State change from (STATE_SENT_SUCCESS, SUBSTATE_NONE)
to (STATE_ACTIVE, SUBSTATE_NONE)
*Mar  8 17:36:44: The Call Setup Information is :

        Call Control Block (CCB) : 0x624D8CCC
         State of The Call        : STATE_ACTIVE
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.231
         Source IP Port    (Media): 20038
         Destn  IP Address (Media): 166.34.245.230
         Destn  IP Port    (Media): 20208
         Destn SIP Addr (Control) : 166.34.245.230
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.230

*Mar  8 17:36:47:  Queued event From SIP SPI to CCAPI/DNS :
SIPSPI_EV_CC_CALL_DISCONNECT
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_active_disconnect
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_CREATE_CONNECTION
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_NONE)  to
(STATE_ACTIVE, SUBSTATE_CONNECTING)
*Mar  8 17:36:47: REQUEST CONNECTION TO IP:166.34.245.230 PORT:5060

*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)
to (STATE_ACTIVE, SUBSTATE_CONNECTING)
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_active_connection_created
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckSocketConnection: Connid(1) created
to 166.34.245.230:5060, local_port 54835
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_SEND_MESSAGE
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sip_stats_method
*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)
to (STATE_DISCONNECTING, SUBSTATE_NONE)
*Mar  8 17:36:47: Sent:
BYE sip:3660110@166.34.245.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
Date: Mon, 08 Mar 1993 22:36:44 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612207
CSeq: 101 BYE
Content-Length: 0

*Mar  8 17:36:47: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  166.34.245.231:54835
From: <sip:3660210@166.34.245.231;user=phone;phone-context=unknown>;tag=27D3FCA8-C7F
To: "3660110" <sip:3660110@166.34.245.230>
```

```
Date: Sat, 06 Mar 1993 19:10:50 GMT
Call-ID: ABBAE7AF-823100CE-0-1CCAA69C@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612207
Content-Length: 0
CSeq: 101 BYE

*Mar  8 17:36:47: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
166.34.245.230:54113
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  act_disconnecting_new_message
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sact_disconnecting_new_message_response
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICheckResponse
*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sip_stats_status_code
*Mar  8 17:36:47:  Roundtrip delay 4 milliseconds for method BYE

*Mar  8 17:36:47: CCSIP-SPI-CONTROL:  sipSPICallCleanup
*Mar  8 17:36:47:  Queued event from SIP SPI : SIPSPI_EV_CLOSE_CONNECTION
*Mar  8 17:36:47: CLOSE CONNECTION TO CONNID:1

*Mar  8 17:36:47: sipSPIIcpifUpdate :CallState: 4 Playout: 1265 DiscTime:66820800
ConnTime 66820420

*Mar  8 17:36:47: 0x624D8CCC : State change from (STATE_DISCONNECTING, SUBSTATE_NONE)
to (STATE_DEAD, SUBSTATE_NONE)
*Mar  8 17:36:47: The Call Setup Information is :

        Call Control Block (CCB) : 0x624D8CCC
         State of The Call        : STATE_DEAD
         TCP Sockets Used         : NO
         Calling Number           : 3660110
         Called Number            : 3660210
         Negotiated Codec         : g711ulaw
         Source IP Address (Media): 166.34.245.231
         Source IP Port    (Media): 20038
         Destn  IP Address (Media): 166.34.245.230
         Destn  IP Port    (Media): 20208
         Destn SIP Addr (Control) : 166.34.245.230
         Destn SIP Port (Control) : 5060
         Destination Name         : 166.34.245.230

*Mar  8 17:36:47:

        Disconnect Cause (CC)    : 16
        Disconnect Cause (SIP)   : 200

*Mar  8 17:36:47: udpsock_close_connect: Socket fd: 1 closed for connid 1 with remote
port: 5060
```

Troubleshooting the Cisco SIP Proxy Server

This section describes troubleshooting features and tips for the Cisco SIP proxy server. It provides tips under the following headings:

- Troubleshooting Tips, page 387
- Cisco SIP Proxy Server Does Not Start, page 387
- Cisco SIP Proxy Server Does Not Allow Devices to Register, page 387
- Cisco SIP Proxy Server Does Not Route Calls Properly, page 388
- Cisco SIP Proxy Server Reports That SIP Messages Are Bad, page 388

- Cisco SIP Proxy Server Farming Does Not Work Correctly, page 388
- Voice Quality Problems, page 388

Troubleshooting Tips

When trying to troubleshoot problems with the Cisco SIP proxy server, remember the following:

- Each module with the Cisco SIP proxy server has debugging capabilities that can be set via a debug flag in the sipd.conf file. When these debug flags are set to On, and the server is running in multi-process mode, the debug output is written to the ./logs/error_log file. When the flags are set to On and single-process mode is enabled, the debug output is written to standard output.
- Changes to the sipd.conf file do not automatically take effect. To have any changes take effect, issue a graceful restart by issuing the following command:

  ./sipdctl graceful

Cisco SIP Proxy Server Does Not Start

If the Cisco SIP proxy server does not start, perform the following tasks as necessary to determine the cause:

- Verify that the /usr/local/sip directory (on Linux) or the opt/local/sip/ directory (on Solaris) has the read and write permissions set that allow you to start the Cisco SIP proxy server.
- Verify that the LD_LIBRARY_PATH environment variable has been enabled as defined in the *Cisco SIP Proxy Server Administrator Guide*, version 2.0.
- If using the Linux RPM version of the Cisco SIP proxy server, verify that the software has been correctly installed.
- Verify that an older version of the Cisco SIP proxy server is not still running. Issue the following command:

  ps -ef | grep -i sip

  If another version is running, disable these processes by issuing the following command:

  ./sipdctl stop

- Verify that the Cisco SIP Proxy Server can resolve its hostname through DNS.

Cisco SIP Proxy Server Does Not Allow Devices to Register

If the Cisco SIP proxy server does not allow devices to register, perform the following tasks as necessary to determine the cause:

- Verify that registration services are enabled in the mod_sip_registry module in the sipd.conf file.
- If authentication is required, ensure that the SIP UA and a password are defined in the MySQL database subscriber table and that the Cisco SIP proxy server can connect to the MySQL database.
- Verify which type of HTTP Digest authentication method that the SIP UAs are using.

Cisco SIP Proxy Server Does Not Route Calls Properly

If the Cisco SIP proxy server does not properly route calls, perform the following tasks as necessary to determine the cause:

- Verify that numbering plan statements are configured correctly in the mod_sip_numexpand module in the sipd.conf file.
- Verify that the translation modules (mod_sip_registry, mod_sip_enum, and mod_sip_gktmp are correctly configured and have the correct entries populated.
- Verify that the correct routes exist in the static routing table of the sipd.conf file.
- Verify that the DNS server is configured for DNS SRV and DNS A records of the devices to be routed.
- View the error_log file for error messages (bad SIP messages, process errors, and so forth.).

Cisco SIP Proxy Server Reports That SIP Messages Are Bad

If the Cisco SIP proxy server reports SIP messages as bad, enable the StateMachine debug flag in the sipd.conf file and view the SIP messages in the error_log file. The error_log file should contain SIP messages that are received in ASCII format. Check the SIP headers of those messages against the headers defined in RFC 2543 or check the SDP information against the information defined in RFC 2327.

Cisco SIP Proxy Server Farming Does Not Work Correctly

If Cisco SIP proxy server farming does not work correctly, perform the following tasks as necessary to determine the cause:

- Verify that all members of the farm have the same sipd.conf file configuration.
- Verify that all members of the farm have an entry for the other farm members defined in the Cisco_Registry_Farm_Members directive in their sipd.conf file.
- Verify that all members of the farm are running the same version of the Cisco SIP proxy server.
- Verify that all members of the farm are synchronized to the same clock source through Network Time Protocol (NTP).

Voice Quality Problems

SIP uses RTP to transmit media between two endpoints. The Cisco SIP proxy server is only involved with the SIP signaling and not the media. Therefore, voice-quality issues should be determined in the endpoint devices, not the Cisco SIP proxy server, because the media does not pass through it.

SIP Messages and Methods

All SIP messages are either requests from a server or client or responses to a request. The messages are formatted according to RFC 822, “Standard for the format of ARPA internet text messages.” For all messages, the general format is:

- A start line
- One or more header fields
- An empty line
- A message body (optional)

Each line must end with a carriage return-line feed (CRLF).

Requests

SIP uses six types (methods) of requests:

- INVITE—Indicates that a user or service is being invited to participate in a call session
- ACK—Confirms that the client has received a final response to an INVITE request
- BYE—Terminates a call and can be sent by the calling or called party
- CANCEL—Cancels any pending searches but does not terminate a call that has already been accepted
- OPTIONS—Queries the capabilities of servers
- REGISTER—Registers the address listed in the To header field with a SIP server

Responses

The following types of responses are used by SIP and generated by the Cisco SIP proxy server:

- SIP 1*xx*—Informational responses
- SIP 2*xx*—Successful responses
- SIP 3*xx*—Redirection responses
- SIP 4*xx*—Client failure responses
- SIP 5*xx*—Server failure responses
- SIP 6xx—Global failure responses

Registration Process

A registration occurs when a client needs to inform a proxy or redirect server of its location. During this process, the client sends a REGISTER request to the proxy or redirect server and includes the address (or addresses) at which it can be reached.

Invitation Process

An invitation occurs when one SIP endpoint (user A) "invites" another SIP endpoint (user B) to join in a call. During this process, user A sends an INVITE message requesting that user B join a particular conference or establish a two-party conversation. If user B wants to join the call, it sends an affirmative response (SIP 2xx). Otherwise, it sends a failure response (SIP 4xx). Upon receiving the response, user A acknowledges the response with an ACK message. If user A no longer wants to establish this conference, it sends a BYE message instead of an ACK message.

Note For examples of SIP call flows, refer to the "Cisco SIP Proxy Server (CSPS) Call Flows" chapter in the *Cisco SIP Proxy Server Administrator Guide, Version 2.0*.