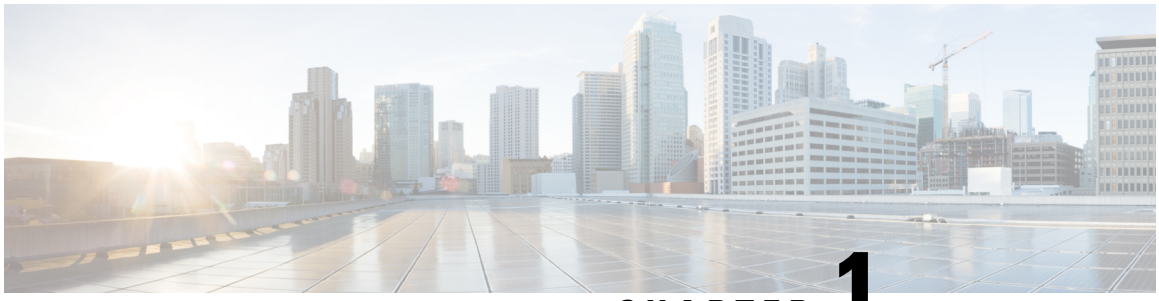
SIP Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Overview of SIP

This chapter provides an overview of the Session Initiation Protocol (SIP).

Information About SIP

Session Initiation Protocol (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.

The Cisco SIP implementation enables supported Cisco platforms to signal the setup of voice and multimedia calls over IP networks.

Like other VoIP 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.

SIP Capabilities

SIP provides the following capabilities:

- Determines the location of the target endpoint--SIP supports address resolution, name mapping, and call redirection.
- Determines the media capabilities of the target endpoint--SIP determines the lowest level of common services between the endpoints through Session Description Protocol (SDP). Conferences are established using only the media capabilities that can be supported by all endpoints.
- Determines the availability of the target endpoint--If a call cannot be completed because the target endpoint is unavailable, SIP determines whether the called party is connected to a call already or did not answer in the allotted number of rings. SIP then returns a message indicating why the target endpoint was unavailable.

- Establishes a session between the originating and target endpoints--If the call can be completed, SIP establishes a session between the endpoints. SIP also supports midcall changes, such as the addition of another endpoint to the conference or the changing of a media characteristic or codec.
- Handles the transfer and termination of calls--SIP supports the transfer of calls from one endpoint to another. During a call transfer, SIP simply establishes a session between the transferee and a new endpoint (specified by the transferring party) and terminates the session between the transferee and the transferring party. At the end of a call, SIP terminates the sessions among all parties.

Note The term "conference" describes an established session (or call) between two or more endpoints. Conferences consist of two or more users and can be established using multicast or multiple unicast sessions.

SIP Components

SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs). A UA can function in one of the following roles:

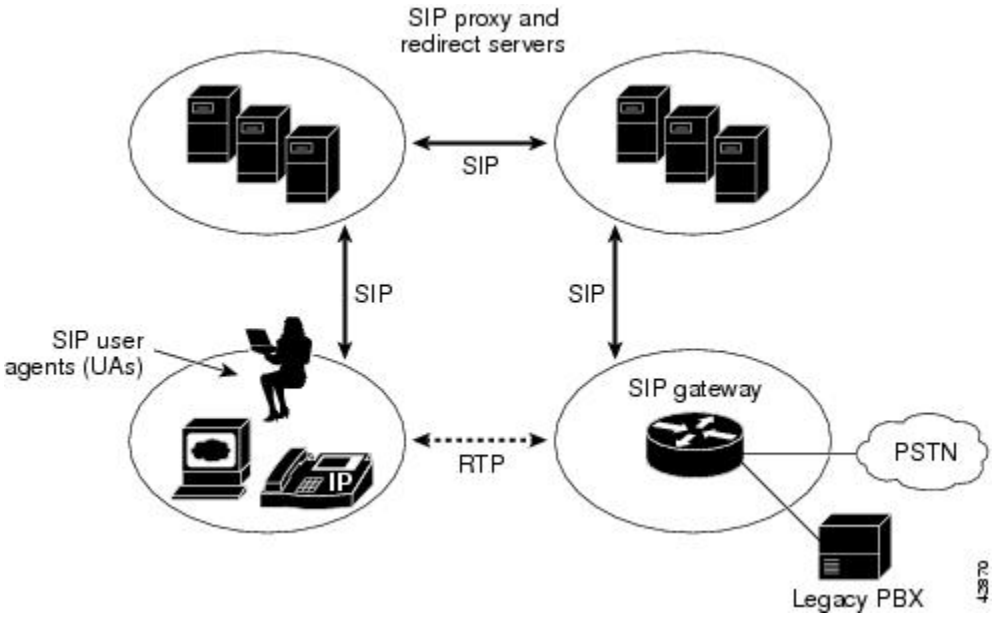
- User-agent client (UAC)--A client application that initiates the SIP request.
- User-agent server (UAS)--A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.

Typically, a SIP endpoint is capable of functioning as both a UAC and a UAS, but functions only as one or the other per transaction. Whether the endpoint functions as a UAC or a UAS depends on the user agent that initiated the request.

From an architectural standpoint, the physical components of a SIP network can be grouped into two categories: clients (endpoints) and servers. The figure below illustrates the architecture of a SIP network.

Note In addition, the SIP servers can interact with other application services, such as Lightweight Directory Access Protocol (LDAP) servers, location servers, a database application, or an extensible markup language (XML) application. These application services provide back-end services, such as directory, authentication, and billing services.

Figure 1: SIP Architecture



SIP Clients

- Phones--Can act as either UAS or UAC.
  - Softphones (PCs that have phone capabilities installed) and Cisco SIP IP phones can initiate SIP requests and respond to requests.
  - ephones--IP phones that are not configured on the gateway.
- Gateways--Provide call control. Gateways provide many services, the most common being a translation function between SIP conferencing endpoints and other terminal types. This function includes translation between transmission formats and between communications procedures. In addition, the gateway translates between audio and video codecs and performs call setup and clearing on both the LAN side and the switched-circuit network side.

SIP Servers

- Proxy server--Receives SIP requests from a client and forwards them on the client's behalf. Basically, proxy servers receive SIP messages and forward them to the next SIP server in the network. Proxy servers can provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server--Provides the client with information about the next hop or hops that a message should take and then the client contacts the next-hop server or UAS directly.
- Registrar server--Processes requests from UACs for registration of their current location. Registrar servers are often co-located with a redirect or proxy server.

How SIP Works

SIP is a simple, ASCII-based protocol that uses requests and responses to establish communication among the various components in the network and to ultimately establish a conference between two or more endpoints.

Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and is in the format of sip:*userID @gateway.com* . The user ID can be either a user name or an E.164 address. The gateway can be either a domain (with or without a hostname) or a specific internet IP address.

Note An E.164 address is a telephone number with a string of decimal digits that uniquely indicates the public network termination point. Thus number contains all information necessary to route the call to this termination point.

Users register with a registrar server using their assigned SIP addresses. The registrar server provides this information to the location server upon request.

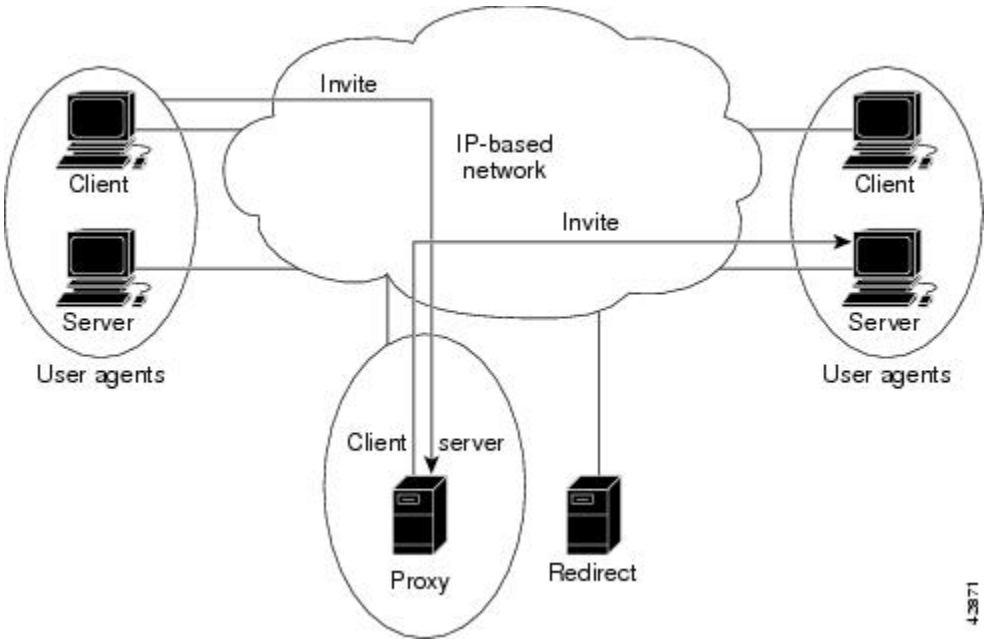
When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller (in the From header field) and the address of the intended called party (in the To header field).

Over time, a SIP end user might move between end systems. The location of the end user can be dynamically registered with the SIP server. The location server can use one or more protocols (including finger, rwhois, and LDAP) to locate the end user. Because the end user can be logged in at more than one station and because the location server can sometimes have inaccurate information, it might return more than one address for the end user. If the request is coming through a SIP proxy server, the proxy server tries each of the returned addresses until it locates the end user. If the request is coming through a SIP redirect server, the redirect server forwards all the addresses to the caller in the Contact header field of the invitation response.

How SIP Works with a Proxy Server

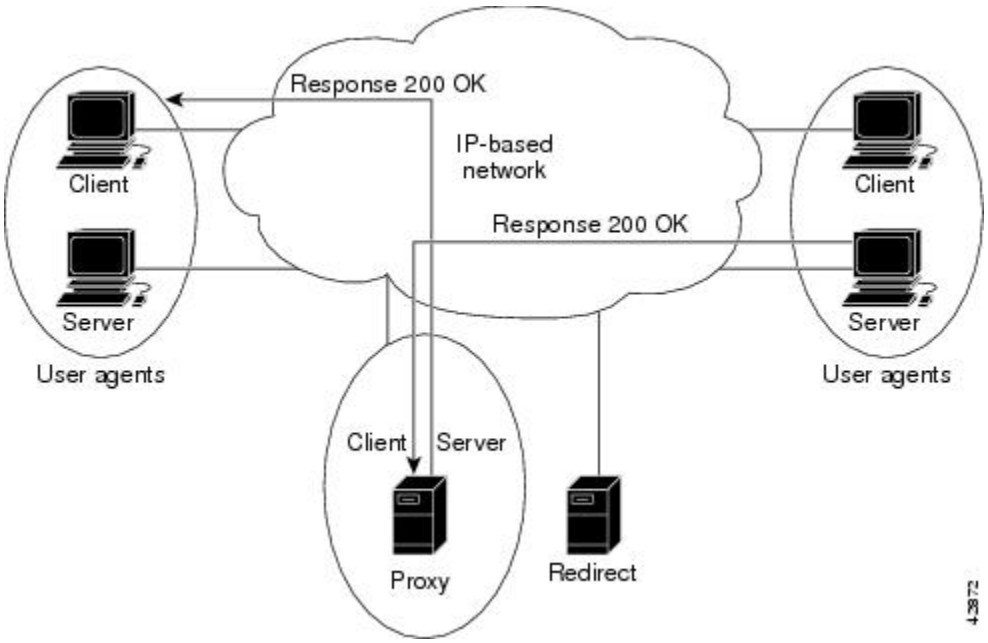
When communicating through a proxy server, the caller UA sends an INVITE request to the proxy server and then the proxy server determines the path and forwards the request to the called party (see the figure below).

Figure 2: SIP INVITE Request Through a Proxy Server



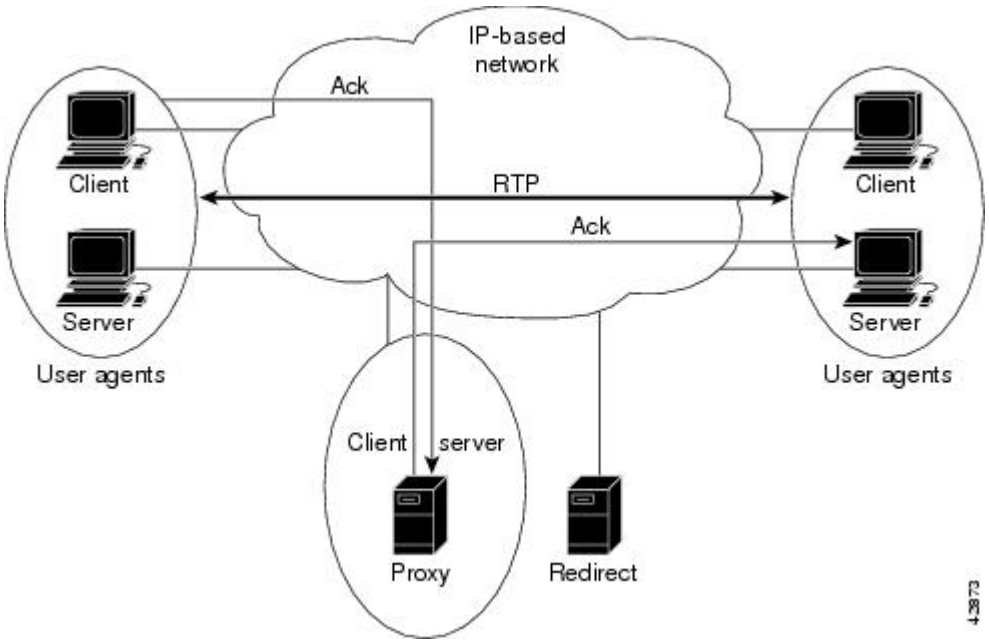
The called UA responds to the proxy server, which then forwards the response to the caller (see the figure below).

Figure 3: SIP Response Through a Proxy Server



When both parties respond with an acknowledgement (SIP ACK message), the proxy server forwards the acknowledgments to their intended party and a session, or conference, is established between them. The Real-time Transfer Protocol (RTP) is then used for communication across the connection now established between the caller and called UA (see the figure below).

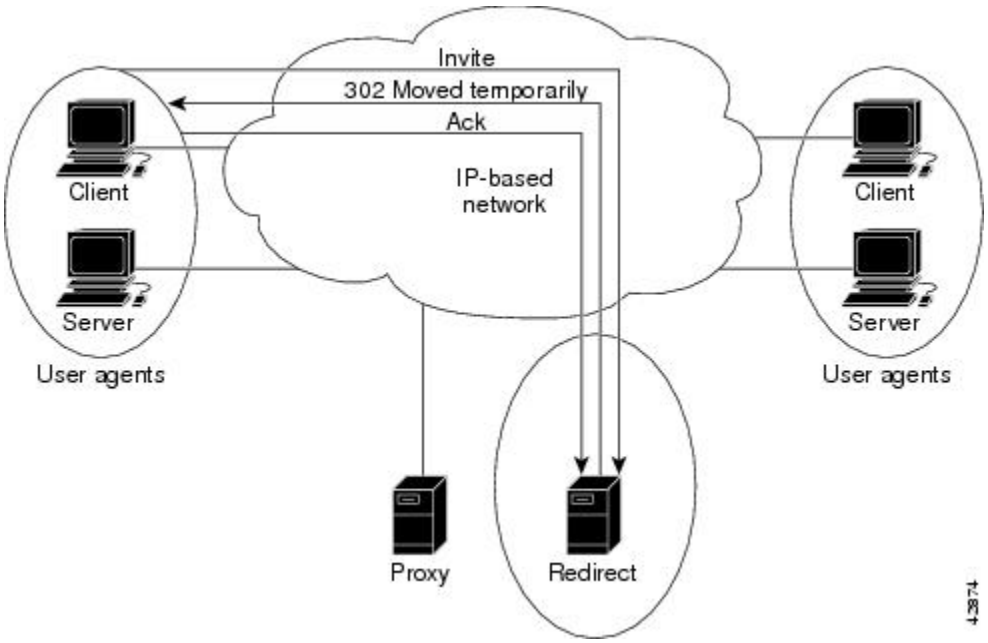
Figure 4: SIP Session Through a Proxy Server



How SIP Works with a Redirect Server

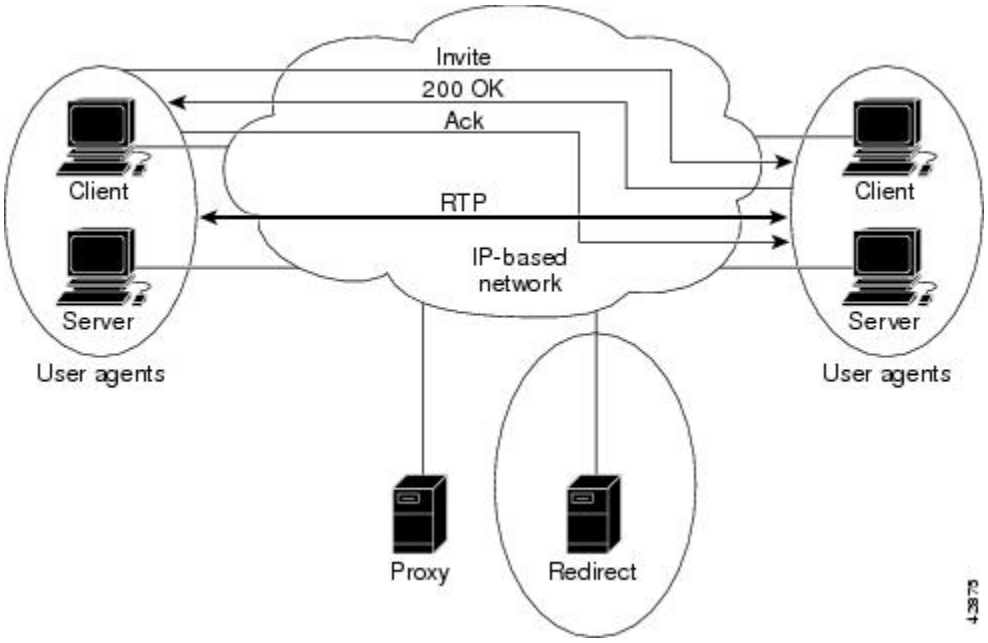
When communicating through a redirect server, the caller UA sends a SIP INVITE request to the redirect server and then the redirect contacts the location server to determine the path to the called party and sends that information back to the caller UA. The caller UA then acknowledges receipt of the information (see the figure below).

Figure 5: SIP INVITE Through a Redirect Server



The caller UA then sends a SIP INVITE request directly to the device indicated in the redirect information, bypassing the redirect server. (The target device at this stage could be either the called UA itself or a proxy server that will forward the request.) Once the request reaches the called UA, the called UA sends a response and, if it is a SIP 200 OK message, the caller UA responds with a SIP ACK message to acknowledge 200 OK response. A session is then established between the two endpoints using RTP for communication between the caller and called UAs (see the figure below).

Figure 6: SIP Session Through a Redirect Server



SIP Call Flows

This topic describes call flows for the following scenarios, which illustrate successful calls:

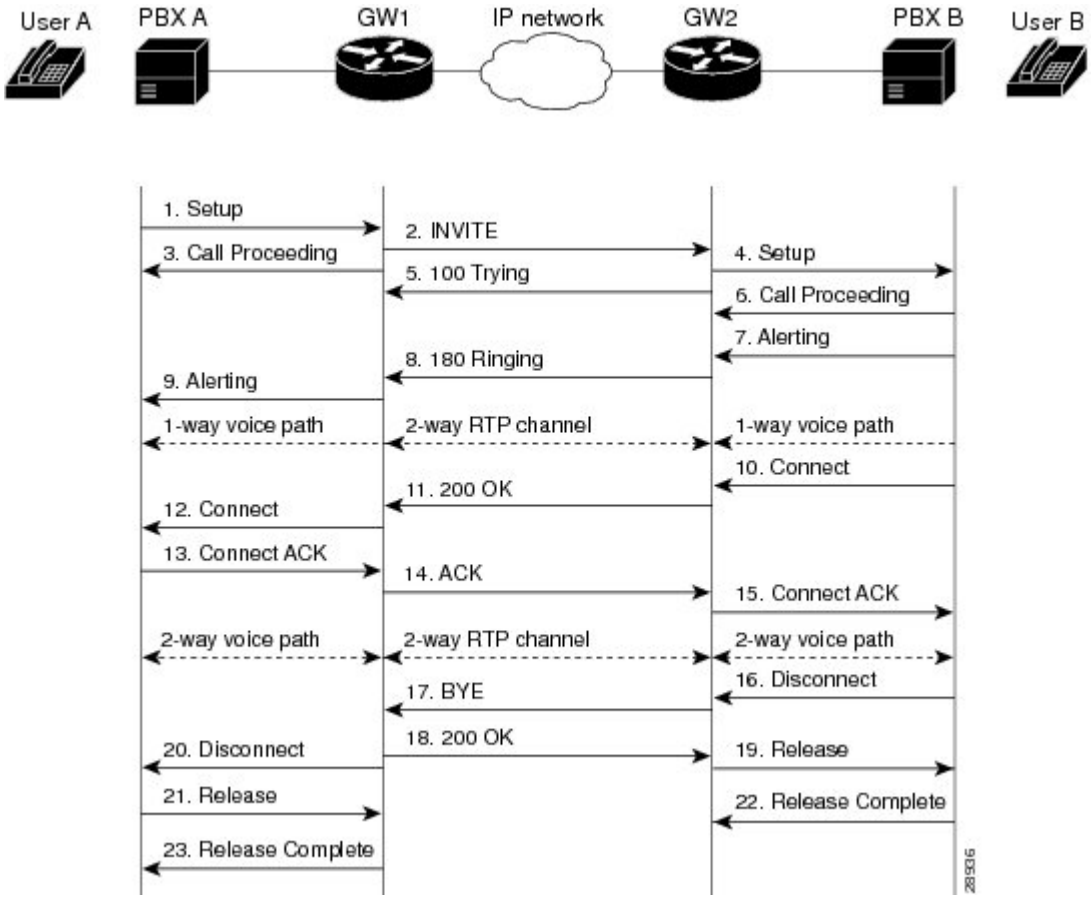
SIP Gateway-to-SIP Gateway--Call Setup and Disconnect

The figure below shows a successful gateway-to-gateway call setup and disconnect. The two end users are User A and User B. User A is located at PBX A, which is connected to SIP gateway 1 via a T1/E1. User B is located at PBX B, which is connected to SIP gateway 2 via a T1/E1. User B's phone number is 555-0100. SIP gateway 1 is connected to SIP gateway 2 over an IP network.

The call flow scenario is as follows:

1. User A calls User B.
2. User B answers the call.
3. User B hangs up.

Figure 7: SIP Gateway-to-SIP Gateway--Call Setup and Disconnect



Note RFC 2543-bis-04 requires that a UAS that receives a BYE request first send a response to any pending requests for that call before disconnecting. After receiving a BYE request, the UAS should respond with a 487 (Request Cancelled) status message.

The following processes occur in the figure.

Process	Description
1. Setup--PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as User A attempts to call User B.
2. INVITE--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an INVITE request to SIP gateway 2. The INVITE request is an invitation to User B to participate in a call session. In the INVITE request the following is the case: • The phone number of User B is inserted in the Request-URI field in the form of a SIP URL. The SIP URL identifies the address of User B and takes a form similar to an email address (user@host where user is the telephone number and host is domain (with or without a hostname) or a numeric network address). For example, the Request-URI field in the INVITE request to User B appears as "INVITE sip:555-0100@example.com; user=phone." The "user=phone" parameter indicates that the Request-URI address is a telephone number rather than a user name. • PBX A is identified as the call session initiator in the From field. • A unique numeric identifier is assigned to the call and is inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability User A is ready to receive is specified. • The port on which SIP gateway 1 is prepared to receive the RTP data is specified.
3. Call Proceeding--SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the setup request.
4. Setup--SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from SIP gateway 1 and initiates call setup with User B via PBX B.
5. 100 Trying--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The 100 Trying response indicates that the INVITE request has been received by SIP gateway 2 but that User B has not yet been located and that some unspecified action, such as a database consultation, is taking place.

Process	Description
6. Call Proceeding--PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the setup request.
7. Alerting--PBX B to SIP gateway 2	PBX B locates User B and sends an Alert message to SIP gateway 2. User B's phone begins ringing.
8. 180 Ringing--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 180 Ringing response to SIP gateway 1. The 180 Ringing response indicates that SIP gateway 2 has located, and is trying to alert, User B.
9. Alerting--SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to User A via PBX A. The Alert message indicates that SIP gateway 1 has received a 180 Ringing response from SIP gateway 2. User A hears the ringback tone that indicates that User B is being alerted. At this point, a one-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
10. Connect--PBX B to SIP gateway 2	User B answers phone. PBX B sends a Connect message to SIP gateway 2. The Connect message notifies SIP gateway 2 that the connection has been made.
11. 200 OK--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 200 OK response to SIP gateway 1. The 200 OK response notifies SIP gateway 1 that the connection has been made. If User B supports the media capability advertised in the INVITE message sent by SIP gateway 1, it advertises the intersection of its own and User A's media capability in the 200 OK response. If User B does not support the media capability advertised by User A, it sends back a 400 Bad Request response with a 304 Warning header field.
12. Connect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The Connect message notifies PBX A that the connection has been made.
13. Connect ACK--PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
14. ACK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that SIP gateway 1 has received the 200 OK response from SIP gateway 2.
15. Connect ACK--SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. The call session is now active over a two-way voice path via Real-time Transport Protocol (RTP). At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
16. Disconnect--PBX B to SIP gateway 2	Once User B hangs up, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.

Process	Description
17. BYE--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a BYE request to SIP gateway 1. The BYE request indicates that User B wants to release the call. Because it is User B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains User B's SIP URL. The cseq value is incremented by one.
18. 200 OK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The 200 OK response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
19. Release--SIP gateway 2 to PBX B	SIP gateway 2 sends a Release message to PBX B.
20. Disconnect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
21. Release--PBX A to SIP gateway 1	PBX A sends a Disconnect message to SIP gateway 1.
22. Release Complete--PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
23. Release Complete--SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the session is terminated.

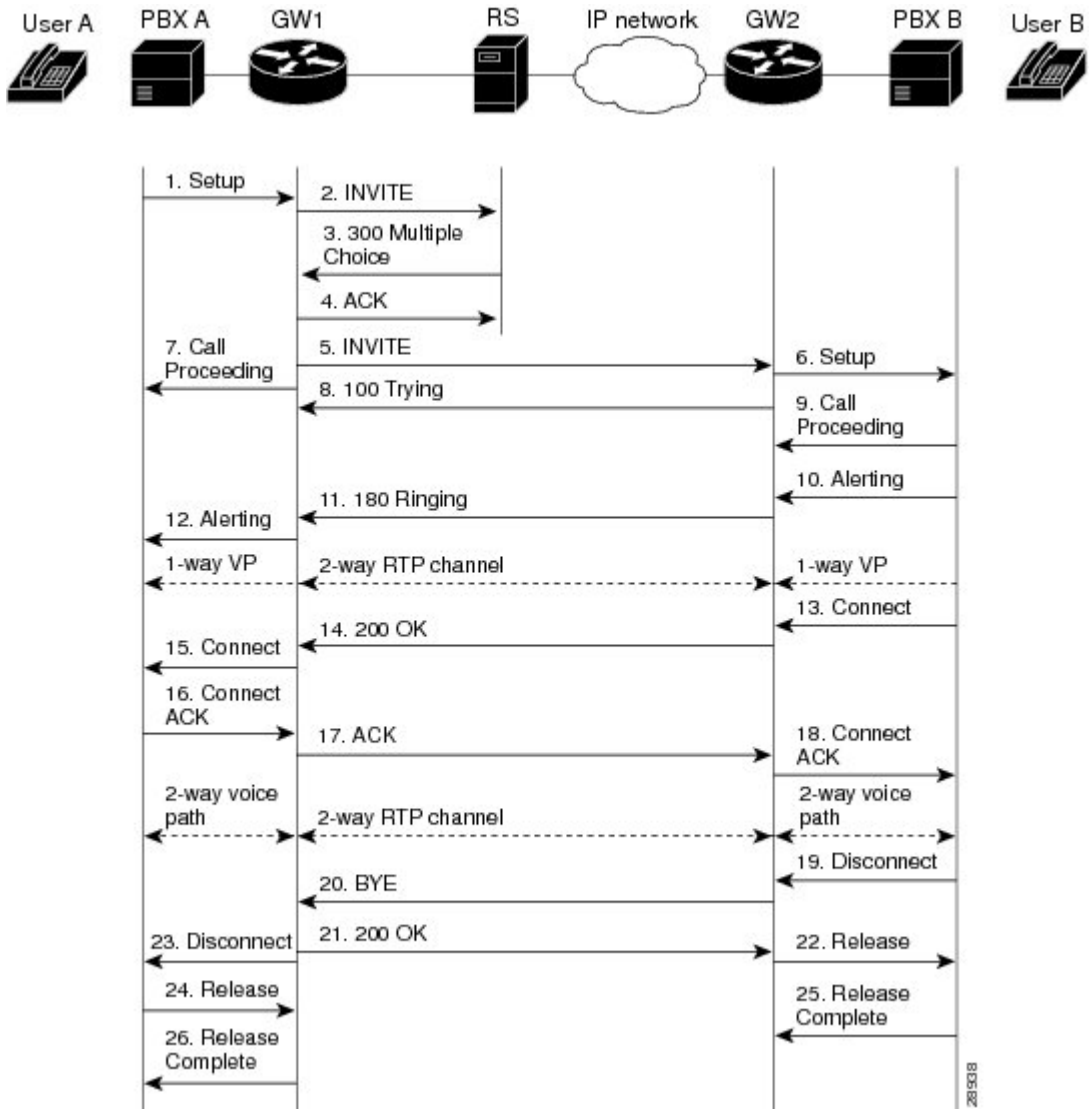
SIP Gateway-to-SIP Gateway--Call via SIP Redirect Server

The figure below shows a successful gateway-to-gateway call setup and disconnect via a SIP redirect server. In this scenario, the two end users are identified as User A and User B. User A is located at PBX A. PBX A is connected to SIP gateway 1 via a T1/E1. SIP gateway 1 is using a SIP redirect server. User B is located at PBX B. PBX B is connected to SIP gateway 2 via a T1/E1. User B's phone number is 555-0100. SIP gateway 1 is connected to SIP gateway 2 over an IP network.

The call flow scenario is as follows:

1. User A calls User B through the SIP gateway 1 using a SIP redirect server.
2. User B answers the call.
3. User B hangs up.

Figure 8: SIP Gateway-to-SIP Gateway--Call via SIP Redirect Server



The following processes occur in the figure.

Process	Description
1. Setup--PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as User A attempts to call User B.

Process	Description
2. INVITE--SIP gateway 1 to SIP redirect server	SIP gateway 1 sends an INVITE request to the SIP redirect server. The INVITE request is an invitation to User B to participate in a call session. In the INVITE request the following is the case: • The phone number of User B is inserted in the Request-URI field in the form of a SIP URL. The SIP URL identifies the address of User B and takes a form similar to an email address (user@host where user is the telephone number and host is a domain (with or without a hostname) or a numeric network address). For example, the Request-URI field in the INVITE request to User B appears as "INVITE sip:555-0100@example.com; user=phone." The "user=phone" parameter distinguishes that the Request-URI address is a telephone number rather than a user name. • PBX A is identified as the call session initiator in the From field. • A unique numeric identifier is assigned to the call and is inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability User A is ready to receive is specified. • The port on which SIP gateway 1 is prepared to receive the RTP data is specified.
3. 300 Multiple Choice--SIP redirect server to SIP gateway 1	The SIP redirect server sends a 300 Multiple Choice response to SIP gateway 1. The 300 Multiple Choice response indicates that the SIP redirect server accepted the INVITE request, contacted a location server with all or part of User B's SIP URL, and the location server provided a list of alternative locations where User B might be located. The SIP redirect server returns these possible addresses to SIP gateway 1 in the 300 Multiple Choice response.
4. ACK--SIP gateway 1 to SIP redirect server	SIP gateway 1 acknowledges the 300 Multiple Choice response with an ACK.
5. INVITE--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a new INVITE request to SIP gateway 2. The new INVITE request includes the first contact listed in the 300 Multiple Choice response as the new address for User B, a higher transaction number in the CSeq field, and the same Call-ID as the first INVITE request.
6. Setup--SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from SIP gateway 1 and initiates call setup with User B through PBX B.
7. Call Proceeding--SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the setup request.

Process	Description
8. 100 Trying--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The 100 Trying response indicates that the INVITE request has been received by SIP gateway 2 but that User B has not yet been located.
9. Call Proceeding--PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the setup request.
10. Alerting--PBX B to SIP gateway 2	PBX B locates User B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.
11. 180 Ringing--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 180 Ringing response to SIP gateway 1. The 180 Ringing response indicates that SIP gateway 2 has located, and is trying to alert, User B.
12. Alerting--SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to PBX A. User A hears ringback tone. At this point, a one-way voice path is established between SIP gateway 1 and PBXA and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
13. Connect--PBX B to SIP gateway 2	User B answers phone. PBX B sends a Connect message to SIP gateway 2. The Connect message notifies SIP gateway 2 that the connection has been made.
14. 200 OK--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 200 OK response to SIP gateway 1. The 200 OK response notifies SIP gateway 1 that the connection has been made. If User B supports the media capability advertised in the INVITE message sent by SIP gateway 1, it advertises the intersection of its own and User A's media capability in the 200 OK response. If User B does not support the media capability advertised by User A, it sends back a 400 Bad Request response with a 304 Warning header field.
15. Connect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The Connect message notifies PBX A that the connection has been made.
16. Connect ACK--PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
17. ACK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that the 200 OK response has been received. The call is now in progress over a two-way voice path via RTP.
18. Connect ACK--SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.

Process	Description
19. Disconnect--PBX B to SIP gateway 2	Once User B hangs up, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
20. BYE--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a BYE request to SIP gateway 1. The BYE request indicates that User B wants to release the call. Because it is User B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains User B's SIP URL.
21. 200 OK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The 200 OK response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
22. Release--SIP gateway 2 to PBX B	SIP gateway 2 sends a Release message to PBX B.
23. Disconnect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
24. Release--PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
25. Release Complete--PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
26. Release Complete--SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the session is terminated.

SIP Gateway-to-SIP Gateway--Call via SIP Proxy Server

The figures below and show a successful gateway-to-gateway call setup and disconnect via a proxy server. The two end users are User A and User B. User A is located at PBX A, which is connected to SIP gateway 1 via a T1/E1. SIP gateway 1 is using a proxy server. SIP gateway 1 is connected to SIP gateway 2 over an IP network. User B is located at PBX B, which is connected to SIP gateway 2 (a SIP gateway) via a T1/E1. User B's phone number is 555-0100.

Note The Record-Route header field is inserted by proxies in a request to force future requests in the dialog to be routed through the proxy.

In the first figure, the record route feature is enabled on the proxy server. In the second figure, record route is disabled on the proxy server.

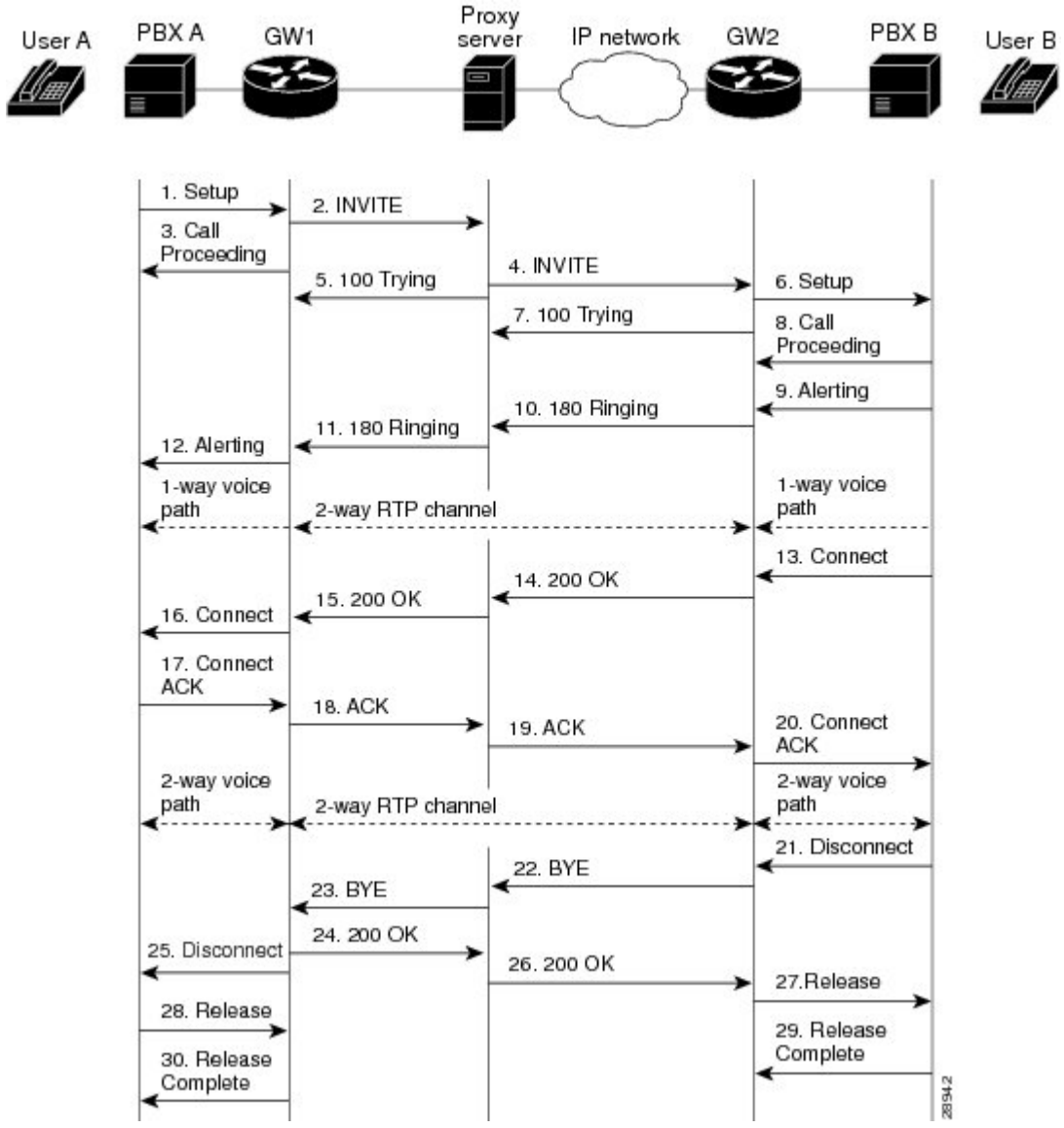
When record route is enabled, the proxy server adds the Record-Route header to the SIP messages to ensure that it is in the path of subsequent SIP requests for the same call leg. The Record-Route field contains a globally reachable Request-URI that identifies the proxy server. When record route is enabled, each proxy server adds its Request-URI to the beginning of the list.

When record route is disabled, SIP messages flow directly through the SIP gateways once a call has been established.

The call flow is as follows:

1. User A calls User B via SIP gateway 1 using a proxy server.
2. User B answers the call.
3. User B hangs up.

Figure 9: SIP Gateway-to-SIP Gateway--Call via SIP Proxy Server with Record Route Enabled



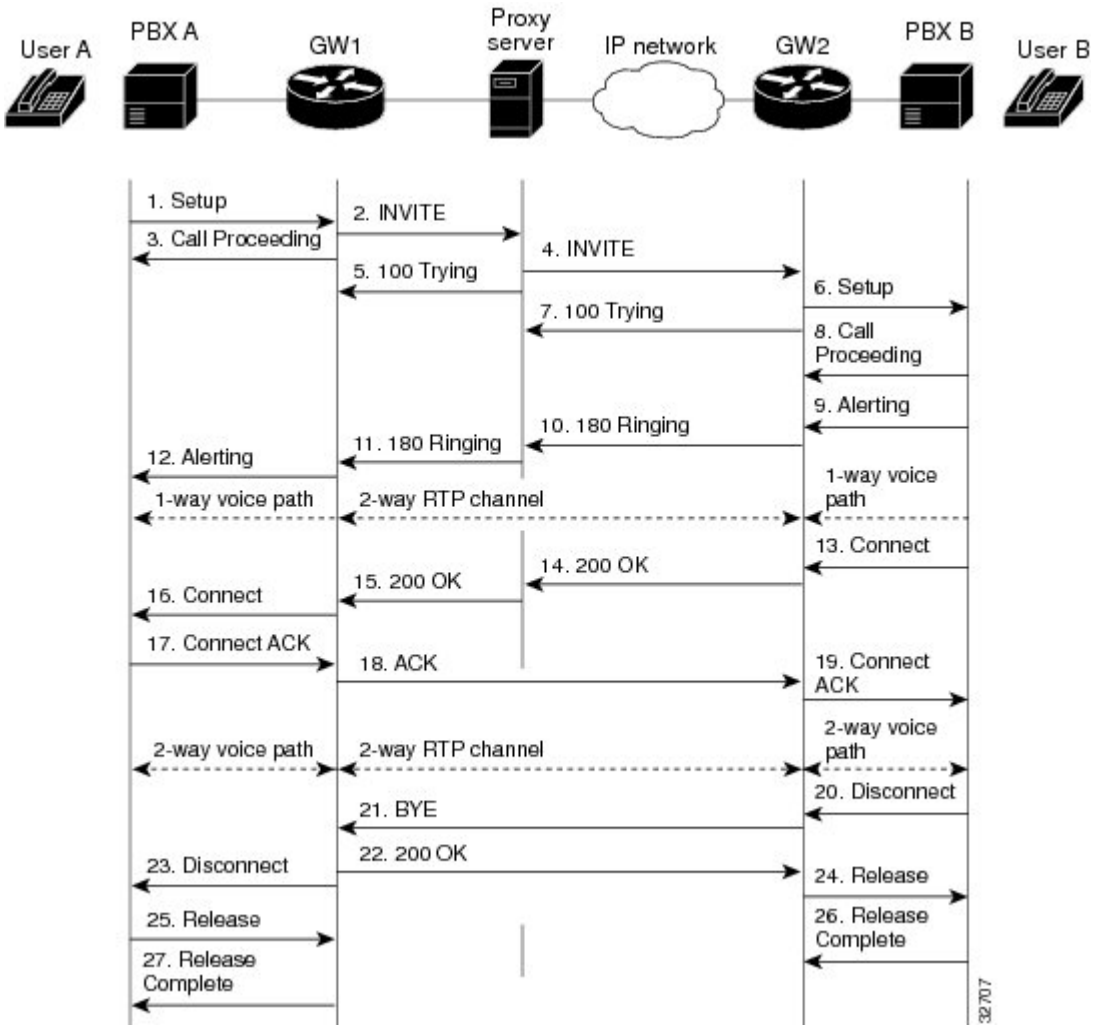
The following processes occur in the figure.

Process	Description
1. Setup--PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as User A attempts to call User B.
2. INVITE--SIP gateway 1 to proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The INVITE request is an invitation to User B to participate in a call session. In the INVITE request: • The phone number of User B is inserted in the Request-URI field in the form of a SIP URL. The SIP URL identifies the address of User B and takes a form similar to an email address (user@host where user is the telephone number and host is a domain (with or without a hostname) or a numeric network address). For example, the Request-URI field in the INVITE request to User B appears as "INVITE sip:555-0100@example.com; user=phone." The "user=phone" parameter distinguishes that the Request-URI address is a telephone number rather than a user name. • PBX A is identified as the call session initiator in the From field. • A unique numeric identifier is assigned to the call and is inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability User A is ready to receive is specified. • The port on which SIP gateway 1 is prepared to receive the RTP data is specified.
3. Call Proceeding--SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the setup request.
4. INVITE--SIP proxy server to SIP gateway 2	The SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and then sends a new INVITE request to SIP gateway 2.
5. 100 Trying--SIP proxy server to SIP gateway 1	The SIP proxy server sends a 100 Trying response to SIP gateway 1.
6. Setup--SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with User B via PBX B.
7. 100 Trying--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server. The SIP proxy server might or might not forward the 100 Trying response to SIP gateway 1.

Process	Description
8. Call Proceeding--PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the setup request.
9. Alerting--PBX B to SIP gateway 2	PBX B locates User B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.
10. 180 Ringing--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 180 Ringing response to the SIP proxy server.
11. 180 Ringing--SIP proxy server to SIP gateway 1	The SIP proxy server forwards the 180 Ringing response to SIP gateway 1.
12. Alerting--SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to User A via PBX A. The Alert message indicates that SIP gateway 1 has received a 180 Ringing response. User A hears the ringback tone that indicates that User B is being alerted. At this point, a one-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
13. Connect--PBX B to SIP gateway 2	User B answers the phone. PBX B sends a Connect message to SIP gateway 2. The connect message notifies SIP gateway 2 that the connection has been made.
14. 200 OK--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 200 OK response (including the Record-Route header received in the INVITE) to the SIP proxy server. The 200 OK response notifies the SIP proxy server that the connection has been made. If User B supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and User A's media capability in the 200 OK response. If User B does not support the media capability advertised by User A, it sends back a 400 Bad Request response with a 304 Warning header field. The SIP proxy server must forward 200 OK responses upstream.
15. 200 OK--SIP proxy server to SIP gateway 1	The SIP proxy server forwards the 200 OK response that it received from SIP gateway 2 to SIP gateway 1. In the 200 OK response, the SIP proxy server forwards the Record-Route header to ensure that it is in the path of subsequent SIP requests for the same call leg. In the Record-Route field, the SIP proxy server adds its Request-URI.
16. Connect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The Connect message notifies PBX A that the connection has been made.
17. Connect ACK--PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
18. ACK--SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an ACK to the SIP proxy server. The ACK confirms that SIP gateway 1 has received the 200 OK response from the SIP proxy server.

Process	Description
19. ACK--SIP proxy server to SIP gateway 2	Depending on the values in the To, From, CSeq, and Call-ID field, the SIP proxy server might process the ACK locally or proxy it. If the fields in the ACK match those in previous requests processed by the SIP proxy server, the server proxies the ACK. If there is no match, the ACK is proxied as if it were an INVITE request. The SIP proxy server forwards SIP gateway 1's ACK response to SIP gateway 2.
20. Connect ACK--SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. The call session is now active. The 2-way voice path is established directly between SIP gateway 1 and SIP gateway 2; not via the SIP proxy server. At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
21. Disconnect--PBX B to SIP gateway 2	After the call is completed, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
22. BYE--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a BYE request to the SIP proxy server. The BYE request indicates that User B wants to release the call. Because it is User B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains User B's SIP URL.
23. BYE--SIP proxy server to SIP gateway 1	The SIP proxy server forwards the BYE request to SIP gateway 1.
24. 200 OK--SIP gateway 1 to SIP proxy server	SIP gateway 1 sends a 200 OK response to the SIP proxy server. The 200 OK response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
25. Disconnect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
26. 200 OK--SIP proxy server to SIP v	The SIP proxy server forwards the 200 OK response to SIP gateway 2.
27. Release--SIP gateway 2 to PBX B	After the call is completed, SIP gateway 2 sends a Release message to PBX B.
28. Release--PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
29. Release Complete--PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
30. Release Complete--SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session is terminated.

Figure 10: SIP Gateway-to-SIP Gateway--Call via a Proxy Server with Record Route Disabled



The following processes occur in the figure.

Process	Description
1. Setup--PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as User A attempts to call User B.

Process	Description
2. INVITE--SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The INVITE request is an invitation to User B to participate in a call session. In the INVITE request the following is the case: • The phone number of User B is inserted in the Request-URI field in the form of a SIP URL. The SIP URL identifies the address of User B and takes a form similar to an email address (user@host where user is the telephone number and host is a domain (with or without a hostname) or a numeric network address). For example, the Request-URI field in the INVITE request to User B appears as "INVITE sip:555-0100@example.com; user=phone." The "user=phone" parameter distinguishes that the Request-URI address is a telephone number rather than a user name. • PBX A is identified as the call session initiator in the From field. • A unique numeric identifier is assigned to the call and is inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability User A is ready to receive is specified. • The port on which SIP gateway 1 is prepared to receive the RTP data is specified.
3. Call Proceeding--SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the setup request.
4. INVITE--SIP proxy server to SIP gateway 2	The SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and then sends a new INVITE request to SIP gateway 2.
5. 100 Trying--SIP proxy server to SIP gateway 1	The SIP proxy server sends a 100 Trying response to SIP gateway 1.
6. Setup--SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with User B via PBX B.
7. 100 Trying--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server. The SIP proxy server might or might not forward the 100 Trying response to SIP gateway 1.
8. Call Proceeding--PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge setup request.
9. Alerting--PBX B to SIP gateway 2	PBX B locates User B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.

Process	Description
10. 180 Ringing--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 180 Ringing response to the SIP proxy server.
11. 180 Ringing--SIP proxy server to SIP gateway 1	The SIP proxy server forwards the 180 Ringing response to SIP gateway 1.
12. Alerting--SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to User A via PBX A. The Alert message indicates that SIP gateway 1 has received a 180 Ringing response. User A hears the ringback tone that indicates that User B is being alerted. At this point, a one-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.
13. Connect--PBX B to SIP gateway 2	User B answers the phone. PBX B sends a Connect message to SIP gateway 2. The connect message notifies SIP gateway 2 that the connection has been made.
14. 200 OK--SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 200 OK response to the SIP proxy server. The 200 OK response notifies the SIP proxy server that the connection has been made. If User B supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and User A's media capability in the 200 OK response. If User B does not support the media capability advertised by User A, it sends back a 400 Bad Request response with a 304 Warning header field. The SIP proxy server must forward 200 OK responses upstream.
15. 200 OK--SIP proxy server to SIP gateway 1	The SIP proxy server forwards the 200 OK response that it received from SIP gateway 2 to SIP gateway 1.
16. Connect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The Connect message notifies PBX A that the connection has been made.
17. Connect ACK--PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
18. ACK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that SIP gateway 1 has received the 200 OK response from the SIP proxy server.
19. Connect ACK--SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. The call session is now active. The 2-way voice path is established directly between SIP gateway 1 and SIP gateway 2; not via the SIP proxy server. At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.

Process	Description
20. Disconnect--PBX B to SIP gateway 2	After the call is completed, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
21. BYE--SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a BYE request to SIP gateway 1. The BYE request indicates that User B wants to release the call. Because it is User B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains User B's SIP URL.
22. 200 OK--SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The 200 OK response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
23. Disconnect--SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
24. Release--SIP gateway 2 to PBX B	After the call is completed, SIP gateway 2 sends a Release message to PBX B.
25. Release--PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
26. Release Complete--PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
27. Release Complete--SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session is terminated.

Additional References

The following sections provide references related to SIP.

Note In addition to the references listed below, each chapter provides additional references related to SIP.

- Some of the products and services mentioned in this guide may have reached end of life, end of sale, or both. Details are available at http://www.cisco.com/en/US/products/prod_end_of_life.html.

Related Documents

Related Topic	Document Title
Basic router configuration	• Cisco 2600 series documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/
	• Cisco 3600 series documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/
	• Cisco 3700 series documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/
	• Cisco AS5300 documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv
Cisco IOS command references	• *Cisco IOS Debug Command Reference, Release 12.3T* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123tcr/123dbr/index.htm
	• *Cisco IOS Voice Command Reference, Release 12.3T* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123tcr/123tvr/index.htm
Cisco IOS configuration fundamentals and examples	• *Cisco IOS Configuration Fundamentals Configuration Guide* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/
	• *Cisco IOS Interface Command Reference* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_r/index.htm
	• *Cisco IOS Interface Configuration Guide* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_c/
	• *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2 at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvvfax_c/index.htm
	• Cisco Systems Technologies website at http://cisco.com/en/US/tech/index.html From the website, select a technology category and subsequent hierarchy of subcategories, then click **T** **Documentation > Configuration Examples**.
Cisco IOS Voice Configuration Library, including library preface and glossary	• Cisco IOS Voice Configuration Library at http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary
Developer support	• Developer Support Agreement at http://www.cisco.com/go/developersupport
IVR script information	• *TCL IVR API 2.0 Programming Guide* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/tclivrv2/index.htm

Related Topic	Document Title
NAT configuration	• *Configuring Network Address Translation: Getting Started* at http://www.cisco.com/warp/publ
SIP documents	• Cisco SIP proxy server documents at http://www.cisco.com/univercd/cc/td/doc/product/voice/s • *Guide to Cisco Systems' VoIP Infrastructure Solution for SIP* at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipsols/biggulp/index.htm • *Session Initiation Protocol Gateway Call Flows* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/rel_docs/sip_flo/index.htm
SS7 for voice gateways	• *Configuring Media Gateways for the SS7 Interconnect for Voice Gateways Solution* at http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel7/soln/das22/gateway/dascfg5.h
Tcl IVR programming	• *Tcl IVR API Version 2.0 Programmer's Guide* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/tclivrv2/index.htm
Troubleshooting	• *Cisco IOS Debug Command Reference, Release 12.3T* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123tcr/123dbr/index.htm
	• *Cisco IOS Voice Troubleshooting and Monitoring Guide* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/voipt_c/in
	• *Internetwork Troubleshooting Guide* at http://www.cisco.com/univercd/cc/td/doc/cisintwk/itg_
	• *Troubleshooting and Debugging VoIP Call Basics* at http://www.cisco.com/warp/public/788/voip/
	• *Voice over IP Troubleshooting and Monitoring* at http://cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/voipt_c/index.ht
	• *VoIP Debug Commands* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/1700/1750/1750voip/debug
VoATM configuration	• *Configuring AAL2 and AAL5 for the High-Performance Advanced Integration Module on the C* athttp://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122x/122xa
VoIP configuration	• *Voice over IP for the Cisco 2600/3600 Series* at http://www.cisco.com/univercd/cc/td/doc/product/access/nubuvoip/voip3600/index.htm
	• *Voice over IP for the Cisco AS5300* at http://www.cisco.com/univercd/cc/td/doc/product/access/nubuvoip/voip5300/index.htm
	• *Voice over IP for the Cisco AS5800* at http://www.cisco.com/univercd/cc/td/doc/product/access/nubuvoip/voip5800/index.htm

Related Topic	Document Title
VSA information	• *RADIUS Vendor-Specific Attributes Voice Implementation Guide* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm
WAN configuration	• *Cisco IOS Wide-Area Networking Command Reference* athttp://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_r/index.htm
	• *Cisco IOS Wide-Area Networking Configuration Guide* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwan_c/wcfatm.htm
Other documents	*Cisco Internetworking Terms and Acronyms* at http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/
	Cisco Resource Policy Management System 2.0 at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/rpms/rpms_2-0/
	VoIP Call Admission Control at http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/voipsol/cac

Standards

Standards[1]	Title
ANSI TI.619/a	*ISDN Multi-Level Precedence and Preemption (MLPP) Service Capability*
draft-ietf-avt-profile-new-12.txt	*RTP Profile for Audio and Video Conferences with Minimal Control*
draft-ietf-avt-rtp-cn-06.txt	*RTP Payload for Comfort Noise* , Internet Draft of the Internet Engineering Task Force (IETF) Audio/Video Transport (AVT) working group
draft-ietf-avt-rtp-mime-06.txt	*MIME Type Registration of RTP Payload Formats*
draft-ietf-mmusic-sdp-comedia-04.txt	*Connection-Oriented Media Transport in SDP*
draft-ietf-sipping-reason-header-for-preemption-00	*Extending the SIP for Preemption Events*
draft-ietf-sip-privacy-02	*SIP Extensions for Caller Identity and Privacy*
draft-ietf-sip-resource-priority-05	*Communications Resources Priority for SIP*
draft-levy-diversion-06.txt	[Sip] verification of diversion header (draft-levy)
GR-268-CORE	*ISDN Basic Rate Interface Call Control Switching and Signalling Generic Requirements*

[1] Not all supported standards are listed.

MIBs

MIBs	MIBs Link
CISCO-SIP-UA-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

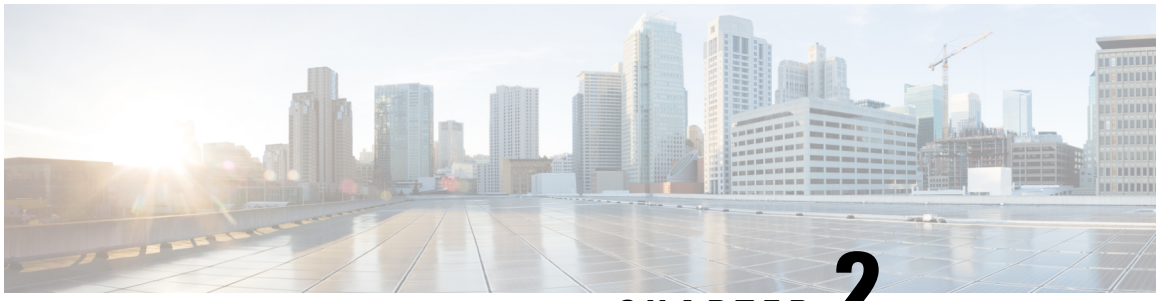
RFC[2]	Title
• RFC 1889 (obsoleted by RFC 3550 in July 2003)	RTP: A Transport Protocol for Real-Time Applications
• RFC 2052 (obsoleted by RFC 2782 in Feb. 2000)	A DNS RR for Specifying location of services (DNS SRV)
• RFC 2543 (and RFC 2543-bis-04) (obsoleted by RFCs 3261, 3262, 3263, 3264, and 3265 in June 2002)	SIP: Session Initiation Protocol
• RFC 2617	HTTP Authentication: Basic and Digest Access Authentication
• RFC 2782 (replaced RFC 2052 in Feb. 2000)	A DNS RR for specifying the location of services (DNS SRV)
• RFC 2806	URLs for Telephone Calls
• RFC 2833 (obsoleted by RFCs 4733 and 4734 in Dec. 2006)	RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
• RFC 2976	SIP INFO Method
• RFC 3261 (replaced RFC 2543 in June 2002 and updated by RFCs 3853 (July 2004) and 4320 (Jan. 2006))	SIP: Session Initiation Protocol
• RFC 3262 (replaced RFC 2543 in June 2002)	Reliability of Provisional Responses in Session Initiation Protocol (SIP)
• RFC 3263 (replaced RFC 2543 in June 2002)	Session Initiation Protocol (SIP): Locating SIP Servers
• RFC 3264 (replaced RFC 2543 in June 2002)	An Offer/Answer Model with Session Description Protocol (SDP)
• RFC 3265 (replaced RFC 2543 in June 2002)	Session Initiation Protocol (SIP)-Specific Event Notification

RFC[2]	Title
• RFC 3311	The Session Initiation Protocol (SIP) UPDATE Method
• RFC 3312 (updated by RFC 4032 in March 2005)	Integration of Resource Management and Session Initiation Protocol (SIP)
• RFC 3326	The Reason Header Field for the Session Initiation Protocol
• RFC 3420	Internet Media Type message/sipfrag
• RFC 3515	The Session Initiation Protocol (SIP) Refer Method
• RFC 3550 (replaced RFC 1889 in July 2003)	RTP: A Transport Protocol for Real-Time Applications
• RFC 3853 (updated RFC 3261 in July 2004)	S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)
• RFC 4032 (updated RFC 3312 in March 2005)	Update to the Session Initiation Protocol (SIP) Preconditions Framework
• RFC 4320 (updated RFC 3261 in July 2004)	Actions Addressing Identified Issues with the Session Initiation Protocol's (SIP) Non-INVITE Transaction
• RFC 4733 (replaced RFC 2833 and was updated by RFC 4734 in Dec. 2006)	RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals
• RFC 4734 (replaced RFC 2833 and updated RFC 4733 in Dec. 2006)	Definition of Events for Modem, Fax, and Text Telephony Signals

[2] Not all supported RFCs are listed.

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 2

Basic SIP Configuration

This chapter provides basic configuration information for the following features:

- SIP Register Support
- SIP Redirect Processing Enhancement
- SIP 300 Multiple Choice Messages
- SIP implementation enhancements:
  - Interaction with Forking Proxies
  - SIP Intra-Gateway Hairpinning

Feature History for SIP Register Support, SIP Redirect Processing Enhancement, and SIP 300 Multiple Choice Messages

Release	Modification
12.2(15)ZJ	This feature was introduced.
12.3(4)T	This feature was integrated into the release.

Feature History for SIP Implementation Enhancements: Interaction with Forking Proxies and SIP Intra-Gateway Hairpinning

Release	Modification
12.2(2)XB	These features were introduced.
12.2(8)T	This feature were integrated into the release.

Finding Support Information for Platforms and Cisco Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

- Prerequisites for Basic SIP Configuration, on page 30
- Restrictions for Basic SIP Configuration, on page 30

Prerequisites for Basic SIP Configuration

SIP Redirect Processing Enhancement Feature

- Ensure that your SIP gateway supports 300 or 302 Redirect messages.

Restrictions for Basic SIP Configuration

- If Hot Standby Router Protocol (HSRP) is configured on the Cisco IOS Gateway, IP-TDM calls are not supported.

Information About Basic SIP Configuration

SIP Register Support

With H.323, Cisco IOS gateways can register E.164 numbers of a POTS dial peer with a gatekeeper, which informs the gatekeeper of a user's contact information. Session Initiation Protocoal (SIP) gateways allow the same functionality, but with the registration taking place with a SIP proxy or registrar. SIP gateways allow registration of E.164 numbers to a SIP proxy or registrar on behalf of analog telephone voice ports (FXS), IP phone virtual voice ports (EFXS), and local SCCP phones.

When registering dial peers with an external registrar, you can also register with a secondary SIP proxy or registrar to provide redundancy. The secondary registration can be used if the primary registrar fails.

SIP gateways allow registration of E.164 numbers to a SIP proxy or registrar server on behalf of analog telephone voice ports (FXS), IP phone virtual voice ports (EFXS), and local SCCP phones. By default, SIP gateways do not generate SIP Register messages. The following tasks set up the gateway to register E.164 telephone numbers with an external SIP registrar.

Note There are no commands that allow registration between the H.323 and SIP protocols.

SIP Redirect Processing Enhancement

SIP Redirect Processing allows flexibility in the handling of incoming redirect or 3*xx* class of responses. Redirect responses can be enabled or disabled through the command-line interface, providing a benefit to service providers who deploy Cisco SIP gateways. Redirect processing is active by default, which means that SIP gateways handle incoming 3*xx* messages in compliance with RFC 2543. RFC 2543 states that redirect

response messages are used by SIP user agents to initiate a new Invite when a user agent learns that a user has moved from a previously known location.

In accordance with RFC 2543-bis-04, the processing of 3*xx* redirection is as follows:

- The uniform resource identifier (URI) of the redirected INVITE is updated to contain the new contact information provided by the 3*xx* redirect message.
- The transmitted CSeq number found in the CSeq header is increased by one. The new INVITE includes the updated CSeq.
- The To, From, and Call ID headers that identify the call leg remain the same. The same Call ID gives consistency when capturing billing history.
- The UAC retries the request at the new address given by the 3*xx* Contact header field.

Redirect handling can be disabled by using the **no redirection** command in SIP user-agent configuration mode. In this case, the user agent treats incoming 3*xx* responses as 4*xx* error class responses. The call is not redirected, and is instead released with the appropriate PSTN cause-code message. The table below shows the mapping of 3*xx* responses to 4*xx* responses.

Table 1: Mapping of 3xx Responses to 4xx Responses

Redirection (3*xx*) Response Message	Mapping to 4*xx* (Client Error) Response
300 Multiple choices	410 Gone
301 Moved Permanently	410 Gone
302 Moved Temporarily	480 Temporarily Unavailable
305 Use Proxy	410 Gone
380 Alternative Service	410 Gone
<any other 3xx response>	410 Gone

SIP Redirect Processing generates call history information with appropriate release cause codes that maybe used for accounting or statistics purposes. When a 3*xx* response is mapped to 4*xx* class of response, the cause code stored in call history is based on the mapped 4*xx* response code.

Call redirection must be enabled on the gateway for SIP call transfer involving redirect servers to be successful.

The Cisco IOS voice gateway can also use call redirection if an incoming VoIP call matches an outbound VoIP dial peer. The gateway sends a 300 or 302 Redirect message to the call originator, allowing the originator to reestablish the call. Two commands allow you to enable the redirect functionality, globally or on a specific inbound dial peer: **redirect ip2ip (dial-peer)**and **redirect ip2ip (voice service)**.

Sending SIP 300 Multiple Choice Messages

Originally, when a call was redirected, the SIP gateway would send a 302 Moved Temporarily message. The first longest match route on a gateway (dial-peer destination pattern) was used in the Contact header of the 302 message. Now, if multiple routes to a destination exist for a redirected number (multiple dial peers are matched), the SIP gateway sends a 300 Multiple Choice message, and the multiple routes in the Contact header are listed.

The **redirect contact order** command gives you the flexibility to choose the order in which routes appear in the Contact header.

How to Perform Basic SIP Configuration

Note For help with a procedure, see the verification and troubleshooting sections listed above.

Configuring SIP VoIP Services on a Cisco Gateway

Shut Down or Enable VoIP Service on Cisco Gateways

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **[no] shutdown [forced]**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.
Step 4	**[no] shutdown [forced]** **Example:** `Router(config-voi-serv)# shutdown forced`	Shuts down or enables VoIP call services.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-voi-serv)# exit`	Exits the current mode.

Shut Down or Enable VoIP Submodes on Cisco Gateways

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **[no] call service stop** [**forced**] [**maintain-registration**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(config-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**[no] call service stop** [**forced**] [**maintain-registration**] **Example:** `Router(conf-serv-sip)# call service stop maintain-registration`	Shuts down or enables VoIP call services for the selected submode.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIP Register Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**]
5. **retry register** *number*
6. **timers register** *milliseconds*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**] **Example:** `Router(config-sip-ua)# registrar ipv4:10.8.17.40 expires 3600 secondary`	Registers E.164 numbers on behalf of analog telephone voice ports (FXS) and IP phone virtual voice ports (EFXS) with an external SIP proxy or SIP registrar server. Keywords and arguments are as follows: • **dns:** *address* --Domain-name server that resolves the name of the dial peer to receive calls. • **ipv4:** *destination-address* --IP address of the dial peer to receive calls.

	Command or Action	Purpose
		• **expires** *seconds* --Default registration time, in seconds. • **tcp** --Sets transport layer protocol to TCP. UDP is the default. • **secondary** --Specifies registration with a secondary SIP proxy or registrar for redundancy purposes. Optional.
Step 5	**retry register** *number* **Example:** `Router(config-sip-ua)# retry register 10`	Use this command to set the total number of SIP Register messages that the gateway should send. The argument is as follows: • *number* --Number of Register message retries. Range: 1 to 10. Default: 10.
Step 6	**timers register** *milliseconds* **Example:** `Router(config-sip-ua)# timers register 500`	Use this command to set how long the SIP user agent waits before sending register requests. The argument is as follows: • *milliseconds* --Waiting time, in ms. Range: 100 to 1000. Default: 500.
Step 7	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Redirect Processing Enhancement

Configure Call-Redirect Processing Enhancement

Redirect processing using the **redirection** command is enabled by default. To disable and then reset redirect processing, perform the steps listed in this section:

IP-to-IP call redirection can be enabled globally or on a dial-peer basis. To configure, perform the steps listed in these sections:

Configuring Call-Redirect Processing Enhancement

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **no redirection**
5. **redirection**
6. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**no redirection** **Example:** `Router(config-sip-ua)# no redirection`	Disables redirect handling--causes the gateway to treat incoming 3*xx* responses as 4*xx* error class responses.
Step 5	**redirection** **Example:** `Router(config-sip-ua)# redirection`	Resets call redirection to work as specified in RFC 2543. The command **default redirection**also resets call redirection to work as specified in RFC 2543.
Step 6	exit **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring Call Redirect to Support Calls Globally

To configure call redirect to support calls globally, perform the following steps.

Note To enable global IP-to-IP call redirection for all VoIP dial peers, use voice-service configuration mode. The default SIP application supports IP-to-IP redirection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **redirect ip2ip**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.
Step 4	**redirect ip2ip** **Example:** `Router(conf-voi-serv)# redirect ip2ip`	Redirect SIP phone calls to SIP phone calls globally on a gateway using the Cisco IOS voice gateway.
Step 5	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits the current mode.

Configuring Call Redirect to Support Calls on a Specific VoIP Dial Peer

Note To specify IP-to-IP call redirection for a specific VoIP dial peer, configure it on an inbound dial peer in dial-peer configuration mode. The default application on SIP SRST supports IP-to-IP redirection.

- When IP-to-IP redirection is configured in dial-peer configuration mode, the configuration on the specific inbound dial peer takes precedence over the global configuration entered under voice service configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **application** *application-name*
5. **redirect ip2ip**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Use this command to enter dial-peer configuration mode. The argument is as follows: • *tag* --Digits that define a particular dial peer. Range: 1to 2,147,483,647 (enter without commas).
Step 4	**application** *application-name* **Example:** `Router(config-dial-peer)# application session`	Enables a specific application on a dial peer. The argument is as follows: • *application-name* --Name of the predefined application you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers. The application must support IP-to-IP redirection
Step 5	**redirect ip2ip** **Example:** `Router(conf-dial-peer)# redirect ip2ip`	Redirects SIP phone calls to SIP phone calls on a specific VoIP dial peer using the Cisco IOS voice gateway.
Step 6	**exit** **Example:** `Router(conf-dial-peer)# exit`	Exits the current mode.

Configuring SIP 300 Multiple Choice Messages

Configuring Sending of SIP 300 Multiple Choice Messages

Note If multiple routes to a destination exist for a redirected number (multiple dial peers are matched), the SIP gateway sends a 300 Multiple Choice message and the multiple routes in the Contact header are listed. This configuration allows users to choose the order in which the routes appear in the Contact header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **redirect contact order** [**best-match** | **longest-match**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(config-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**redirect contact order** [**best-match**	**longest-match**] **Example:** `Router(conf-serv-sip)# redirect contact order best-match`	Sets the order of contacts in the 300 Multiple Choice Message. Keywords are as follows: • **best-match** --Use the current system configuration to set the order of contacts. • **longest-match** --Set the contact order by using the destination pattern longest match first, and then the second longest match, the third longest match, and so on. This is the default.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIP Implementation Enhancements

Minor underlying or minimally configurable features are described in the following sections:

For additional information on SIP implementation enhancements, see "Achieving SIP RFC Compliance."

Interaction with Forking Proxies

Call forking enables the terminating gateway to handle multiple requests and the originating gateway to handle multiple provisional responses for the same call. Call forking is required for the deployment of the *find me/follow me* type of services.

Support for call forking enables the terminating gateway to handle multiple requests and the originating gateway to handle multiple provisional responses for the same call. Interaction with forking proxies applies to gateways acting as a UAC, and takes place when a user is registered to several different locations. When the UAC sends an INVITE message to a proxy, the proxy forks the request and sends it to multiple user agents. The SIP gateway processes multiple 18X responses by treating them as independent transactions under the same call ID. When the relevant dial peers are configured for QoS, the gateway maintains state and initiates RSVP reservations for each of these independent transactions. When it receives an acknowledgment, such as a 200 OK, the gateway accepts the successful acknowledgment and destroys state for all other transactions.

The forking feature sets up RSVP for each transaction *only* if the dial peers are configured for QoS. If not, the calls proceed as best-effort.

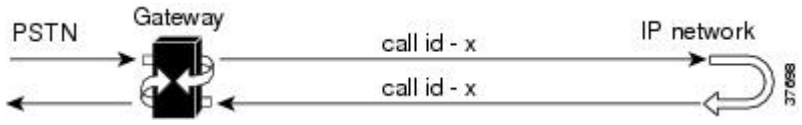
Support for interaction with forking proxies applies only to gateways acting as UACs. It does not apply when the gateway acts as a UAS. In that case, the proxy forks multiple INVITES with the same call ID to the same gateway but with different request URLs.

Also, the forking feature sets up RSVP for each transaction *only* if the dial peers are configured for QoS. If not, the calls proceed as best-effort.

SIP Intra-Gateway Hairpinning

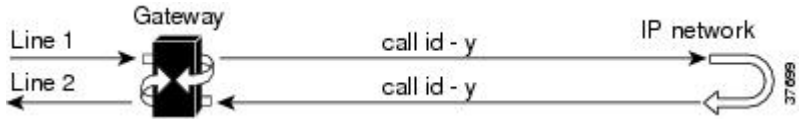
SIP hairpinning is a call routing capability in which an incoming call on a specific gateway is signaled through the IP network and back out the same gateway. This can be a PSTN call routed into the IP network and back out to the PSTN over the same gateway (see the figure below).

Figure 11: PSTN Hairpinning Example



Similarly, SIP hairpinning can be a call signaled from a line (for example, a telephone line) to the IP network and back out to a line on the same access gateway (see the figure below).

Figure 12: Telephone Line Hairpinning Example



With SIP hairpinning, unique gateways for ingress and egress are unnecessary.

SIP supports plain old telephone service (POTS)-to-POTS hairpinning (which means that the call comes in one voice port and is routed out another voice port). It also supports POTS-to-IP call legs and IP-to-POTS call legs. However, it does not support IP-to-IP hairpinning. This means that the SIP gateway cannot take an inbound SIP call and reroute it back to another SIP device using the VoIP dial peers.

Only minimal configuration is required for this feature. To enable hairpinning on the SIP gateway, see the following configuration example for dial peers. Note that:

- The POTS dial peer must have preference 2 defined, and the VoIP dial peer must have preference 1 defined. This ensures that the call is sent out over IP, not Plain Old Telephone Service (POTS).
- The session target is the same gateway because the call is being redirected to it.

```
!
dial-peer voice 53001 pots
 preference 2
 destination-pattern 5300001
 prefix 5300001
!
dial-peer voice 53002 pots
 preference 2
 destination-pattern 5300002
 prefix 5300002
!
dial-peer voice 530011 voip
 preference 1
 destination-pattern 5300001
 session protocol sipv2
 session target ipv4:10.1.1.41
 playout-delay maximum 300
 codec g711alaw
!
dial-peer voice 530022 voip
 preference 1
 destination-pattern 5300002
 session protocol sipv2
 session target ipv4:10.1.1.41
 playout-delay maximum 300
 codec g711alaw
```

Verifying SIP Gateway Status

To verify SIP gateway status and configuration, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show sip service**
2. **show sip-ua register status**
3. **show sip-ua statistics**
4. **show sip-ua status**
5. **show sip-ua timers**

DETAILED STEPS

Step 1 show sip service

Use this command to display the status of SIP call service on a SIP gateway.

The following sample output shows that SIP call service is enabled:

Example:

```
Router# show sip service
SIP Service is up
```

The following sample output shows that SIP call service was shut down with the **shutdown** command:

Example:

```
Router# show sip service
SIP service is shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop** command:

Example:

```
Router# show sip service
SIP service is shut
under 'voice service voip', 'sip' submode
```

The following sample output shows that SIP call service was shut down with the **shutdown forced** command:

Example:

```
Router# show sip service
SIP service is forced shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop forced** command:

Example:

```
Router# show sip service
SIP service is forced shut
under 'voice service voip', 'sip' submode
```

Step 2 show sip-ua register status

Use this command to display the status of E.164 numbers that a SIP gateway has registered with an external primary SIP registrar.

Example:

```
Router# show sip-ua register status
Line peer expires(sec) registered
4001 20001 596          no
4002 20002 596          no
5100 1     596          no
9998 2     596          no
```

Step 3 show sip-ua statistics

Use this command to display response, traffic, and retry SIP statistics, including whether call redirection is disabled.

The following sample shows that four registers were sent:

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
     Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OkSubscribe 0/0, OkNOTIFY 0/0,
      OkInfo 0/0, 202Accepted 0/0
      OkRegister 12/49
     Redirection (Inbound only except for MovedTemp(Inbound/Outbound)) :
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0/0, UseProxy 0,
      AlternateService 0
      Client Error:
      BadRequest 0/0, 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,
      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, RequestCancel 0/0,
      NotAcceptableMedia 0/0, BadEvent 0/0,
      SETooSmall 0/0
     Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
     Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
      Miscellaneous counters:
      RedirectRspMappedToClientErr 0
SIP Total Traffic Statistics (Inbound/Outbound)
      Invite 0/0, Ack 0/0, Bye 0/0,
      Cancel 0/0, Options 0/0,
      Prack 0/0, Comet 0/0,
      Subscribe 0/0, NOTIFY 0/0,
      Refer 0/0, Info 0/0
      Register 49/16
Retry Statistics
      Invite 0, Bye 0, Cancel 0, Response 0,
      Prack 0, Comet 0, Reliable1xx 0, NOTIFY 0
      Register 4
SDP application statistics:
Parses: 0, Builds 0
Invalid token order: 0, Invalid param: 0
Not SDP desc: 0, No resource: 0
Last time SIP Statistics were cleared: <never>
```

The following sample output shows the RedirectResponseMappedToClientError status message. An incremented number indicates that 3*xx* responses are to be treated as 4*xx* responses. When call redirection is enabled (default), the RedirectResponseMappedToClientError status message is not incremented.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OKSubscribe 0/0, OkNotify 0/0,
      202Accepted 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, UseProxy 0,
      AlternateService 0
    Client Error:
      BadRequest 0/0, 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,
      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, RequestCancel 0/0
      NotAcceptableMedia 0/0, BadEvent 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
    Miscellaneous counters:
      RedirectResponseMappedToClientError 1,
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0
Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0, Notify 0
SDP application statistics:
 Parses: 0,  Builds 0
 Invalid token order: 0,  Invalid param: 0
 Not SDP desc: 0,  No resource: 0
```

Step 4 **show sip-ua status**

Use this command to display status for the SIP user agent (UA), including whether call redirection is enabled or disabled.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 1 (rfc 2052)
Redirection (3xx) message handling: ENABLED
```

Step 5 **show sip-ua timers**

Use this command to display the current settings for the SIP user-agent (UA) timers.

The following sample output shows the waiting time before a register request is sent--that is, the value that is set with the **timers register** command:

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 180000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
refer 500, register 500
```

General Troubleshooting Tips

For more information on troubleshooting, see the following references:

- "Cisco IOS Voice Troubleshooting and Monitoring Guide"
- Cisco Technical Support at http://www.cisco.com/en/US/support/index.html
- *Cisco IOS Debug Command Reference*
- *Cisco IOS Voice, Video, and Fax Configuration Guide*
- Troubleshooting and Debugging VoIP Call Basics
- VoIP Debug Commands

Note Commands are listed in alphabetical order.

- Make sure that VoIP is working.
- Make sure that you can make a voice call.
- Verify that SIP-supported codecs are used. Support for codecs varies on different platforms; use the **codec ?** command to determine the codecs available on a specific platform.
- Use the **debug aaa authentication** command to display high-level diagnostics related to AAA logins.

- Use the **debug asnl events**command to verify that the SIP subscription server is up. The output displays a pending message if, for example, the client is unsuccessful in communicating with the server.
- Use the debug call fallback family of commands to display details of VoIP call fallback.
- Use the **debug cch323**family of commands to provide debugging output for various components within an H.323 subsystem.
- Use the **debug ccsip**family of commands for general SIP debugging, including viewing direction-attribute settings and port and network address-translation traces. Use any of the following related commands:
  - **debug ccsip all--**Enables all SIP-related debugging
  - **debug ccsip calls**--Enables tracing of all SIP service-provider interface (SPI) calls
  - **debug ccsip error**--Enables tracing of SIP SPI errors
  - **debug ccsip events**--Enables tracing of all SIP SPI events
  - **debug ccsip info**--Enables tracing of general SIP SPI information, including verification that call redirection is disabled
  - **debug ccsip media**--Enables tracing of SIP media streams
  - **debug ccsip messages**--Enables all SIP SPI message tracing, such as those that are exchanged between the SIP user-agent client (UAC) and the access server
  - **debug ccsip preauth**--Enables diagnostic reporting of authentication, authorization, and accounting (AAA) preauthentication for SIP calls
  - **debug ccsip states**--Enables tracing of all SIP SPI state tracing
  - **debug ccsip transport**--Enables tracing of the SIP transport handler and the TCP or User Datagram Protocol (UDP) process
- Use the **debug isdn q931**command to display information about call setup and teardown of ISDN network connections (layer 3) between the local router (user side) and the network.
- Use the **debug kpml** command to enable debug tracing of KeyPad Markup Language (KPML) parser and builder errors.
- Use the **debug radius** command to enable debug tracing of RADIUS attributes.
- Use the **debug rpms-proc preauth** command to enable debug tracing on the RPMS process for H.323 calls, SIP calls, or both H.323 and SIP calls.
- Use the debug rtr trace command to trace the execution of an SAA operation.
- Use the **debug voip** family of commands, including the following:
  - **debug voip ccapi protoheaders --**Displays messages sent between the originating and terminating gateways. If no headers are being received by the terminating gateway, verify that the **header-passing** command is enabled on the originating gateway.
  - **debug voip ivr script--**Displays any errors that might occur when the Tcl script is run
  - **debug voip rtp session named-event 101 --**Displays information important to DTMF-relay debugging, if you are using codec types g726r16 or g726r24. Be sure to append the argument *101* to thecommand to prevent the console screen from flooding with messages and all calls from failing.

Sample output for some of these commands follows:

Sample Output for the debug ccsip events Command

- The example shows how the Proxy-Authorization header is broken down into a decoded username and password.

```
Router# debug ccsip events
CCSIP SPI: SIP Call Events tracing is enabled
21:03:21: sippmh_parse_proxy_auth: Challenge is 'Basic'.
21:03:21: sippmh_parse_proxy_auth: Base64 user-pass string is 'MTIzNDU2Nzg5MDEyMzQ1Njou'.
21:03:21: sip_process_proxy_auth: Decoded user-pass string is '1234567890123456:.'.
21:03:21: sip_process_proxy_auth: Username is '1234567890123456'.
21:03:21: sip_process_proxy_auth: Pass is '.'.
21:03:21: sipSPIAddBillingInfoToCcb: sipCallId for billing records =
10872472-173611CC-81E9C73D-F836C2B6@172.18.192.19421:03:21: ****Adding to UAS Request table
```

Sample Output for the debug ccsip info Command

This example shows only the portion of the debug output that shows that call redirection is disabled. When call redirection is enabled (default), there are no debug line changes.

```
Router# debug ccsip info
00:20:32: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 172.18.207.10
:5060
00:20:32: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
00:20:32: CCSIP-SPI-CONTROL:  sipSPICheckResponse
00:20:32: sip_stats_status_code
00:20:32: ccsip_get_code_class: !!Call Redirection feature is disabled on the GW
00:20:32: ccsip_map_call_redirect_responses: !!Mapping 302 response to 480
00:20:32:  Roundtrip delay 4 milliseconds for method INVITE
```

Configuration Examples for Basic SIP Configuration

SIP Register Support Example

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
 network 192.168.0.0 255.255.255.0
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
```

```
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
```

```
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP Redirect Processing Enhancement Examples

This section provides configuration examples to match the identified configuration tasks in the previous sections.

Note IP addresses and hostnames in examples are fictitious.

Call Redirection Disabled

This example shows that call redirection is disabled on the gateway.

```
Router# show running-config
Building configuration...
Current configuration : 2791 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
interface FastEthernet2/0
ip address 172.18.200.24 255.255.255.0
duplex auto
no shut
speed 10
ip rsvp bandwidth 7500 7500
!
```

```
voice-port 1/1/1
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 8183821111
port 1/1/1
!
dial-peer voice 3 voip
application session
destination-pattern 7173721111
session protocol sipv2
session target ipv4:172.18.200.36
codec g711ulaw
!
dial-peer voice 4 voip
application session
destination-pattern 6163621111
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
sip-ua
no redirection
   retry invite 1
   retry bye 1
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Call Redirection Enabled

This example shows that call redirection is enabled on the gateway (the default). WHen call redirection is enabled, the output shows no redirection.

```
Router# show running-config
Building configuration...
Current configuration : 2791 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
interface FastEthernet2/0
ip address 172.18.200.24 255.255.255.0
duplex auto
no shut
speed 10
ip rsvp bandwidth 7500 7500
!
voice-port 1/1/1
```

```
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 8183821111
port 1/1/1
!
dial-peer voice 3 voip
application session
destination-pattern 7173721111
session protocol sipv2
session target ipv4:172.18.200.36
codec g711ulaw
!
dial-peer voice 4 voip
application session
destination-pattern 6163621111
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
sip-ua
   retry invite 1
   retry bye 1
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Call Redirection Using IP-to-IP Redirection

This example shows that redirection was set globally on the router.

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
 network 192.168.0.0 255.255.255.0
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
voice call carrier capacity active
!
voice class codec 1
```

```
 codec preference 2 g711ulaw
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
```

```
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP 300 Multiple Choice Messages Example

This section provides a configuration example showing redirect contact order set to best match.

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
 network 192.168.0.0 255.255.255.0
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
```

```
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
```

```
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

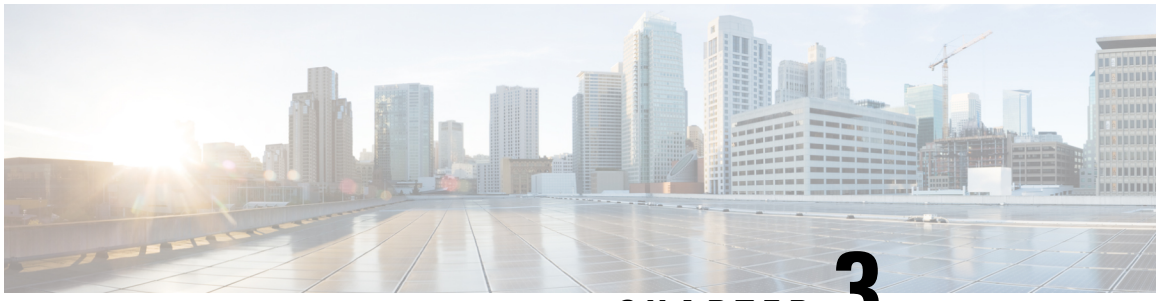
Toll Fraud Prevention

When a Cisco router platform is installed with a voice-capable Cisco IOS software image, appropriate features must be enabled on the platform to prevent potential toll fraud exploitation by unauthorized users. Deploy these features on all Cisco router Unified Communications applications that process voice calls, such as Cisco Unified Communications Manager Express (Cisco Unified CME), Cisco Survivable Remote Site Telephony (SRST), Cisco Unified Border Element (Cisco UBE), Cisco IOS-based router and standalone analog and digital PBX and public-switched telephone network (PSTN) gateways, and Cisco contact-center VoiceXML gateways. These features include, but are not limited to, the following:

- Disable secondary dial tone on voice ports--By default, secondary dial tone is presented on voice ports on Cisco router gateways. Use private line automatic ringdown (PLAR) for foreign exchange office (FXO) ports and direct-inward-dial (DID) for T1/E1 ports to prevent secondary dial tone from being presented to inbound callers.
- Cisco router access control lists (ACLs)--Define ACLs to allow only explicitly valid sources of calls to the router or gateway, and therefore to prevent unauthorized SIP or H.323 calls from unknown parties to be processed and connected by the router or gateway.
- Close unused SIP and H.323 ports--If either the SIP or H.323 protocol is not used in your deployment, close the associated protocol ports. If a Cisco voice gateway has dial peers configured to route calls outbound to the PSTN using either time division multiplexing (TDM) trunks or IP, close the unused H.323 or SIP ports so that calls from unauthorized endpoints cannot connect calls. If the protocols are used and the ports must remain open, use ACLs to limit access to legitimate sources.
- Change SIP port 5060--If SIP is actively used, consider changing the port to something other than well-known port 5060.
- SIP registration--If SIP registration is available on SIP trunks, turn on this feature because it provides an extra level of authentication and validation that only legitimate sources can connect calls. If it is not available, ensure that the appropriate ACLs are in place.
- SIP Digest Authentication--If the SIP Digest Authentication feature is available for either registrations or invites, turn this feature on because it provides an extra level of authentication and validation that only legitimate sources can connect calls.
- Explicit incoming and outgoing dial peers--Use explicit dial peers to control the types and parameters of calls allowed by the router, especially in IP-to-IP connections on Cisco Unified CME, SRST, and Cisco UBE. Incoming dial peers offer additional control on the sources of calls, and outgoing dial peers on the destinations. Incoming dial peers are always used for calls. If a dial peer is not explicitly defined, the implicit dial peer 0 is used to allow all calls.

- Explicit destination patterns--Use dial peers with more granularity than .T for destination patterns to block disallowed off-net call destinations. Use class of restriction (COR) on dial peers with specific destination patterns to allow even more granular control of calls to different destinations on the PSTN.
- Translation rules--Use translation rules to manipulate dialed digits before calls connect to the PSTN to provide better control over who may dial PSTN destinations. Legitimate users dial an access code and an augmented number for PSTN for certain PSTN (for example, international) locations.
- Tcl and VoiceXML scripts--Attach a Tcl/VoiceXML script to dial peers to do database lookups or additional off-router authorization checks to allow or deny call flows based on origination or destination numbers. Tcl/VoiceXML scripts can also be used to add a prefix to inbound DID calls. If the prefix plus DID matches internal extensions, then the call is completed. Otherwise, a prompt can be played to the caller that an invalid number has been dialed.
- Host name validation--Use the “permit hostname” feature to validate initial SIP Invites that contain a fully qualified domain name (FQDN) host name in the Request Uniform Resource Identifier (Request URI) against a configured list of legitimate source hostnames.
- Dynamic Domain Name Service (DNS)--If you are using DNS as the “session target” on dial peers, the actual IP address destination of call connections can vary from one call to the next. Use voice source groups and ACLs to restrict the valid address ranges expected in DNS responses (which are used subsequently for call setup destinations).

For more configuration guidance, see the “ Cisco IOS Unified Communications Manager Express Toll Fraud Prevention ” paper.



CHAPTER 3

Achieving SIP RFC Compliance

This chapter describes how to use or configure Cisco IOS Session Initiation Protocol (SIP) gateways to comply with published SIP standards. It discusses the following features:

- RFC 4040-Based Clear Channel Codec Negotiation for SIP Calls
- SIP: Core SIP Technology Enhancements (RFC 2543 and RFC 2543-bis-04)
- SIP: DNS SRV RFC 2782 Compliance (RFC 2782)
- SIP: RFC 3261 Enhancements (RFC 3261)
- SIP Gateway Compliance to RFC 3261, RFC 3262, and RFC 3264
- SIP Stack Portability

Note This feature is described in the "Configuring SIP Message, Timer, and Response Features" feature module.

Feature History for RFC4040-Based Clear Channel Codec Negotiation for SIP Calls

Release	Modification
15.0(1)XA	This feature is supported on Cisco IOS SIP time division multiplexing (TDM) gateways and Cisco Unified Border Elements (Cisco UBEs). For details about enabling this feature, see the **encap clear-channel standard** and **voice-class sip encap clear-channel** commands in the *Cisco IOS Voice Command Reference*.

Feature History for SIP: Core SIP Technology Enhancements

Release	Modification
12.2(13)T	This feature was introduced to achieve compliance with SIP RFC 2543-bis-04, later published as RFC_3261.

Feature History for SIP - DNS SRV RFC 2782 Compliance

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(8)T	This feature was integrated into the release.

Feature History for SIP: RFC 3261 Enhancements

Release	Modification
12.3(4)T	This feature was introduced.

Feature History for SIP Gateway Compliance to RFC 3261, RFC 3262, and RFC 3264

Release	Modification
12.3(8)T	This feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP RFC Compliance

- Configure a basic VoIP network.
- Enable the Reliable Provisional Response feature.

Note For information on reliable provisional responses, see the "SIP Gateway Support of RSVP and TEL URL" feature module.

Restrictions for SIP RFC Compliance

- As found in RFC 3261, the following are not supported:
  - Sending SIP UPDATE requests; the gateway is able to receive and process only UPDATE requests.
  - SIP with IPv6 host addresses.
  - Secure SIPs. Secure SIPs are secure Uniform Resource Identifiers (URIs). When a caller makes a call using SIPs, the message transport is secure to the called party.
  - Field characters 0x0 to 0x7E in quoted strings within SIP headers encoded in Unicode Transformation Format Version 8 (UTF-8).
- As found in RFC 3264, the following are not supported:
  - Support for bandwidth (b=) SDP attribute equal to 0 is not supported.
  - Initial INVITE with 0.0.0.0 is not supported unless ACK contains a valid IP address.

Note With CSCub35268, the initial INVITE with 0.0.0.0 is supported. When Cisco UBE receives an initial INVITE with 0.0.0.0 IP address, streams are created and Cisco UBE sends out the response for the mid-call DO re-INVITE.

Information About SIP RFC Compliance

SIP RFC 2543 Compliance

The Cisco SIP gateway complies with RFC 2543. However, RFC 3261 has now replaced (obsoleted) RFC 2543. See "Restrictions for SIP RFC Compliance" and "SIP RFC 3261 Compliance" for more information about what is and is not supported in the new RFCs.

SIP RFC 2782 Compliance

SIP on Cisco VoIP gateways uses Domain Name System Server (DNS SRV) query to determine the IP address of the user endpoint. The query string has a prefix in the form of "protocol.transport."as defined by RFC 2052. This prefix is attached to the fully qualified domain name (FQDN) of the next-hop SIP server.

A second prefix style has been added to Cisco VoIP gateways and is now the default. This second style is defined by RFC 2782, which obsoleted RFC 2052 in February 2000. This new style is in compliance with RFC 2782 and appends the protocol label with an underscore "_" as in "_protocol._transport." The addition of the underscore reduces the risk of the same name being used for unrelated purposes.

SIP RFC 3261 Compliance

RFC 3261, which obsoletes RFC 2543, defines the SIP signaling protocol for creating, modifying, and terminating sessions. Cisco's implementation of RFC 3261 supports the following:

- Ability to receive and process SIP UPDATE requests

- Initial Offer and Answer exchanges
- Branch and Sent-by parameters in the Via header
- Merged request detection
- Loose-routing

Benefits of RFC 3261 include the following:

- Continued interoperability of Cisco IOS gateways in current SIP deployments
- Expanded interoperability of Cisco IOS gateways with new SIP products and applications

SIP Header Fields Network Components and Methods

The tables below show RFC 3261 SIP functions--including headers, components, and methods. They also show if the specific functionality is supported by Cisco SIP gateways.

Table 2: SIP Header Fields

Header Field	Supported by Cisco Gateways?
Accept	Yes. Used in OPTIONS response messages.
Accept-Encoding	No
Accept-Language	Yes
Alert-Info	No
Allow	Yes
Also	
Authentication-Info	No
Authorization	
Call-ID	Yes
Call-Info	No
CC-Diversion / Diversion	Yes
Contact	
Content-Disposition	
Content-Encoding	No
Content-Encoding	Yes
Content-Language	No

Header Field	Supported by Cisco Gateways?
Content-Length	Yes
Content-Type	
Cseq	
Date	
Encryption	No
Error-Info	
Event	Yes
Expires	
From	
In-Reply-To	No
Max-Forwards	Yes
MIME-Version	
Min-Expires	
Min-SE	
Organization	No
Priority	
Proxy-Authenticate	
Proxy-Authenticate	Yes
Proxy-Authorization	
Proxy-Require	No
Rack	Yes
Reason	
Record-Route	
Referred-By	
Referred-To	
Replaces	
Requested-By	
Require	

Header Field	Supported by Cisco Gateways?
Response-Key	No
Retry-After	
Retry-After	Yes
Route	
RSeq	
Server	
Session-Expires	
Subject	No
Supported	Yes
Timestamp	
To	
Unsupported	
User-Agent	
Via	
Warning	
WWW-Authenticate	No
WWW-Authenticate	Yes

Table 3: SIP Network Components

SIP Network Components	Supported by Cisco Gateways?
User Agent Client (UAC)	Yes
User Agent Server (UAS)	
Proxy Server	No
Redirect Server	Yes
Registrar Server	

Table 4: SIP Methods

Method	Supported by Cisco Gateways?
ACK	Yes
BYE	
CANCEL	
COMET	Deprecated. Conditions MET. Used in Quality of Service (QoS) implementations to indicate to the opposite endpoint whether or not the conditions have been met--that is, if the proper resources have been reserved.
INVITE	Yes. SIP gateways support midcall Invite requests with the same call-ID but different Session Description Protocols (SDP) session parameters (to change the transport address). Midcall INVITE requests can also change the port number, codec, or refresh the session timer value.
INFO	Yes. SIP gateways can accept and generate INFO messages.
NOTIFY	Yes. Used in implementation of the Refer requests. Notify messages let the initiator of the Refer request know the outcome of the transfer. Notify messages also let a subscriber know of any changes occurring in selected events, such as dual tone multifrequency events (DTMF) or message waiting indication (MWI) events.
OPTIONS	Yes. SIP gateways receive this method only.
PRACK	Yes. Enable or Disable Provisional Reliable Acknowledgements (PRACK).
REFER	Yes. The SIP gateway responds to a Refer request and also generates a Refer request for attended and blind call transfers.
REGISTER	Yes. The SIP gateway can send and receive SIP REGISTER requests.
SUBSCRIBE	Yes. The SIP gateway can generate and accept SUBSCRIBE requests. The gateway processes SUBSCRIBE requests for selected applications such as DTMF telephony events and for MWI.
UPDATE	Yes. The SIP gateway can accept UPDATEs for media changes, target refreshes, and QoS scenarios. The gateway will send UPDATEs only for QoS scenarios.

SIP Responses

The tables below show SIP responses that are supported by Cisco SIP gateways in compliance with RFC 3261.

Cisco SIP gateways do not initiate the use of keepalive messages for calls that they originate or terminate. If the remote gateway uses a keepalive message, the SIP gateway complies.

Table 5: 1xx Responses

1xx Responses	Comments
100 Trying	Action is being taken on behalf of the caller, but that the called party has not yet been located.The SIP gateway generates this response for an incoming Invite request. Upon receiving this response, a gateway stops retransmitting Invite requests and waits for a 180 Ringing or 200 OK response.
180 Ringing	The called party has been located and is being notified of the call. The SIP gateway generates a 180 Ringing response when the called party has been located and is being alerted. Upon receiving this response, the gateway generates local ringback, then it waits for a 200 OK response.
181 Call is being forwarded	The call is being rerouted to another destination. The SIP gateway does not generate these responses. Upon receiving these responses, the gateway processes the responses in the same way it processes a 100 Trying response.
182 Queued	The called party is not currently available, but has elected to queue the call rather than reject it. The SIP gateway does not generate these responses. Upon receiving these responses, the gateway processes the responses in the same way it processes a 100 Trying response.
183 Session progress	Performs inband alerting for the caller. The SIP gateway generates a 183 Session progress response when it receives an ISDN Progress message with an appropriate media indication from the PSTN.

Table 6: 2xx Responses

2xx Responses	Comments
202 Accepted	The SIP gateway will send this response for incoming REFER and SUBSCRIBE requests. It will accept this response for outgoing REFER and SUBSCRIBE requests.
200 OK	The request has been successfully processed. The action taken depends on the request.

Table 7: 3xx Responses

3xx Responses	Comments
The SIP gateway does not generate this response. Upon receiving this response, the gateway contacts the new address in the Contact header field.	
300 Multiple Choice	The address resolved to more than one location. All locations are provided and the user or user agent (UA) is allowed to select which location to use.

3xx Responses	Comments
301 Moved permanently	The user is no longer available at the specified location. An alternate location is included in the header.
302 Moved temporarily	The user is temporarily unavailable at the specified location. An alternate location is included in the header.
305 Use proxy	The caller must use a proxy to contact the called party.
380 Alternative service	The call was unsuccessful, but that alternative services are available.

Table 8: 4xx Responses

4xx Responses	Comments
Upon receiving a 4xx response, the SIP gateway initiates a graceful call disconnect and clears the call.	
423 Interval Too Brief	The SIP gateway generates this response.
400 Bad Request	The request could not be understood because of an illegal format. The SIP gateway generates this response for a badly formed request.
401 Unauthorized	The request requires user authentication. The SIP gateway does not generate this response.
402 Payment required	Payment is required to complete the call. The SIP gateway does not generate this response.
403 Forbidden	The server has received and understood the request but will not provide the service. The SIP gateway does not generate this response.
404 Not Found	The server has definite information that the user does not exist in the specified domain. The SIP gateway generates this response if it is unable to locate the called party.
405 Method Not Allowed	The method specified in the request is not allowed. The response contains a list of allowed methods. The SIP gateway generates this response if an invalid method is specified in the request.
406 Not Acceptable	The requested resource is capable of generating only responses that have content characteristics not acceptable as specified in the accept header of the request. The SIP gateway does not generate this response.
407 Proxy authentication required	Similar to a 401 Unauthorized response. However, the client must first authenticate itself with the proxy. The SIP gateway does not generate this response.
408 Request timeout	The server could not produce a response before the Expires time out. The SIP gateway does not generate this response.

4xx Responses	Comments
410 Gone	A resource is no longer available at the server and no forwarding address is known. The SIP gateway generates this response if the PSTN returns a cause code of unallocated number.
413 Request entity too large	The server refuses to process the request because it is larger than the server is willing or able to process. The SIP gateway does not generate this response.
414 Request-URI too long	The server refuses to process the request because the Request-URI is too long for the server to interpret. The SIP gateway does not generate this response.
415 Unsupported media	The server refuses to process the request because the format of the body is not supported by the destination endpoint. The SIP gateway generates this response when it gets an Info message for an unsupported event-type. Supported event types are 0-9, A-D, # and *.
416 Unsupported Request URI scheme	The SIP gateway generates this response when it gets an unsupported URI scheme such as http: or sips: in a SIP request.
420 Bad extension	The server could not understand the protocol extension indicated in the Require header. The SIP gateway generates this response if it cannot understand the service requested.
421 Extension Required	The SIP gateway does not generate this response.
422 Session Timer Too Small	Generated by the UAS when a request contains a Session-Expires header with a duration that is below the minimum timer for the gateway server. The 422 response must contain a Min-SE header with a minimum timer for that server.
480 Temporarily unavailable	The called party was contacted but is temporarily unavailable. The SIP gateway generates this response if the called party is unavailable. For example, the called party does not answer the phone within a certain amount of time, or the called number does not exist or is no longer in service.
481 Call leg/transaction does not exist	The server is ignoring the request because the request was either a Bye request for which there was no matching leg ID, or a Cancel request for which there was no matching transaction. The SIP gateway generates this response if the call leg ID or transaction cannot be identified.
482 Loop detected	The server received a request that included itself in the path. A SIP gateway generates this response when it detects the same request has arrived more than once in different paths (most likely due to forking).
483 Too many hops	The server received a request that required more hops than allowed by the Max-Forwards header. The SIP gateway does not generate this response.
484 Address incomplete	The server received a request containing an incomplete address. The SIP gateway does not generate this response.

4xx Responses	Comments
485 Ambiguous	The server received a request in which the called party address was ambiguous. It can provide possible alternate addresses. The SIP gateway does not generate this response.
486 Busy here	The called party was contacted but that their system is unable to take additional calls. The SIP gateway generates this response if the called party was contacted but was busy.
487 Request cancelled	The request was terminated by a Bye or Cancel request. The SIP gateway generates this response to an unexpected Bye or Cancel received for a request.
488 Not Acceptable Media	Indicates an error in handling the request at this time. The SIP gateway generates this response if media negotiation fails.
491 Request Pending	The SIP gateway generates this response to reject an UPDATE message proposing a new offer, if it receives the new offer before it receives an answer to an offer it has previously requested.
493 Undecipherable	The SIP gateway does not generate this response.

Table 9: 5xx Responses

5xx Responses	Comments
The SIP gateway generates this response if it encountered an unexpected error that prevented it from processing the request. Upon receiving this response, the gateway initiates a graceful call disconnect and clears the call.	
500 Server internal error	The server or gateway encountered an unexpected error that prevented it from processing the request.
501 Not implemented	The server or gateway does not support the functions required to complete the request.
502 Bad gateway	The server or gateway received an invalid response from a downstream server.
503 Service unavailable	The server or gateway is unable to process the request due to an overload or maintenance problem.
504 Gateway timeout	The server or gateway did not receive a timely response from another server (such as a location server).
505 Version not supported	The server or gateway does not support the version of the SIP protocol used in the request.
513 Message too large	The SIP gateway does not generate this response.

5xx Responses	Comments
580 Precondition failed	A failure in having QoS preconditions met for a call.

Table 10: 6xx Responses

6xx Responses	Comments
The SIP gateway does not generate this response. Upon receiving this response, the gateway initiates a graceful call disconnect and clears the call.	
600 Busy everywhere	The called party was contacted but that the called party is busy and cannot take the call at this time.
603 Decline	The called party was contacted but cannot or does not want to participate in the call.
604 Does not exist anywhere	The server has authoritative information that the called party does not exist in the network.
606 Not acceptable	The called party was contacted, but that some aspect of the session description was unacceptable.

SIP SDP Usage Transport Layer Protocols and DNS Records

The tables below show SIP SDP usage, transport protocols, and DNS records that are supported in RFC 3261. They also show if the specific functionality is supported by Cisco SIP gateways.

Table 11: SIP Session Description Protocol (SDP) Usage Supported in RFC 3261

SIP Network Components	Supported by Cisco Gateways?
a (Media attribute line)	Yes. The primary means for extending SDP and tailoring it to a particular application or media.
c (Connection information)	Yes.
m (Media name and transport address)	
o (Owner/creator and session identifier)	
s (Session name)	
t (Time description)	
v (Protocol version)	

Table 12: SIP Transport Layer Protocols

Protocol	Supported by Cisco Gateways?
Multicast UDP	No
TCP	Yes
TLS	No
Unicast UDP	Yes

Table 13: SIP Domain Name System (DNS) Records

Authentication Encryption Mode	Supported by Cisco Gateways?
RFC 3263 Type A	Yes
RFC 3263 Type NAPTR	No
RFC 3263 Type SRV	Yes

SIP Extensions

The table below shows supported SIP extensions.

Table 14: SIP Extensions

SIP Extension	Comments
RFC 3262: Reliability of Provisional Responses in SIP	Supported.
RFC 3263: Locating SIP Servers	The gateway does not support DNS NAPTR lookups. It supports DNS SRV and A record lookups and has the provision to cycle through the multiple entries.
RFC 3265: SIP Specific Event Notification	The gateway supports the SUBSCRIBE-NOTIFY framework.
RFC 3311: SIP UPDATE Method	The gateway accepts UPDATE for media changes, target refreshes, and QoS Scenarios. It sends UPDATE for only QoS scenarios.
RFC 3312: Integration of Resource Management and SIP - RFC	Midcall QoS changes do not use the 183-PRACK model defined in this RFC.
RFC 3326: Reason Header field for SIP	The gateway uses this to relay the Q.850 cause code to the remote SIP device.

SIP Extension	Comments
RFC 3515: SIP REFER Method	The gateway does not send or accept out-of-dialog REFER requests. Overlapping REFERs are not supported. REFER is supported only in the context of call transfer scenarios (that is, triggered INVITE cases only). The gateway supports relevant portions of RFC 3892 (Referred-By) and RFC 3891 (Replaces header) as needed for call-transfer scenarios.

SIP Security

The tables below show SIP security encryption and responses supported in RFC 3261. They also show if the specific functionality is supported by Cisco SIP gateways.

Table 15: SIP Encryption Modes

Encryption Mode	Supported by Cisco Gateways?
End-to-end Encryption	No. IPSEC can be used for security.
Hop-by-Hop Encryption	
Privacy of SIP Responses	No.
Via Field Encryption	No. IPSEC can be used for security.

Table 16: SIP Authentication Encryption Modes

Authentication Encryption Mode	Supported by Cisco Gateways?
Digest Authentication	Yes
PGP	No
Proxy Authentication	No
Secure SIP or sips	URI scheme is not supported

SIP DTMF Relay

Cisco SIP gateways support DTMF relay in accordance with RFC 2833. The DTMF relay method is based on the transmission of Named Telephony Events (NTE) and DTMF digits over a Real-Time Transport Protocol (RTP) stream.

Cisco SIP gateways also support forwarding DTMF tones by means of cisco-rtp, which is a Cisco proprietary payload type.

The table below shows SIP DTMF relay methods. It also shows if the specific method is supported by Cisco SIP gateways.

Table 17: SIP DTMF Relay Supported in RFC 3261

Method	Supported by Cisco Gateways?
RFC 2833	Yes. The default RTP payload type for rtp-nte is 101. The default method of DTMF relay is inband voice.
Cisco RTP (Cisco proprietary)	Yes, except on Cisco AS5350 and Cisco AS5400.

SIP Fax Relay and T.38

The table below shows fax relay modes that are supported by Cisco SIP gateways in compliance with RFC 3261. It also shows if the specific method is supported by Cisco SIP gateways.

Table 18: Fax Relay Modes Supported in RFC 3261

Method	Supported by Cisco Gateways?
T.38 Fax Relay	Yes
Cisco Fax Relay	Yes, except on Cisco AS5350 and Cisco AS5400

Cisco SIP gateways support T.38 and T.37 fax relay, store, and forward mechanisms. The table below is based on Annex-D of the T.38 ITU recommendation, *Procedures for Real-Time Group 3 Facsimile Communication over IP Networks* , June 1998. The table indicates recommendations from the standard and if Cisco SIP gateways support the requirements.

Table 19: T.38 Fax Requirements

Requirement	Description	Mandatory or Optional	Supported?
SIPt38-01	T.38 over SIP must be implemented as described in ANNEX D of the T.38 ITU recommendation, *Procedures for Real-Time Group 3 Facsimile Communication over IP Networks* , June 1998.	Mandatory	Yes
SIPt38-02	SIP-enabled VoIP gateways detect calling tones (CNG), called station identifier (CED) fax tones, and/or the preamble flag sequence transmitted inside the audio RTP streams.	Mandatory	Yes -- only the CED V.21 preamble and not the CNG tone is used to detect fax.
SIPt38-03	Fax transmission detection is performed by the receiving gateway by recognizing the CED tone.	Mandatory	Yes
SIPt38-04	If the CED tone is not present, the fax transmission is detected by the receiving gateway by recognizing the Preamble flag sequence.	Mandatory	Yes
SIPt38-05	Upon detection of the fax transmission, the receiving gateway initiates the switch over to T.38 fax mode by sending a reINVITE request with SDP.	Mandatory	Yes

Requirement	Description	Mandatory or Optional	Supported?
SIPt38-06	To prevent glare, even if the transmitting gateway detects the fax transmission (CNG tone), the gateway does not initiate the switch over to T.38 fax mode.	Mandatory	Yes
SIPt38-07	If a SIP session starts with audio capabilities and then switches to fax, the session switches back to audio mode at the end of the fax transmission.	Mandatory	Yes
SIPt38-08	Support of SIP T.38 fax calls over TCP.	Desirable	UDP only
SIPt38-09	Facsimile UDP transport Layer (UDPTL) is supported.	Mandatory	Yes
SIPt38-10	The following SDP attributes support T.38 fax sessions: • Registered SDP Protocol format, MIME media type image/t38: • MIME media type name: image • MIME subtype name: t38	Mandatory	Yes
SIPt38-11	The following attributes support T.38 sessions. • T38FaxVersion • T38maxBitRate • T38FaxFillBitRemoval • T38FaxTranscodingMMR • T38FaxTranscodingJBIG • T38FaxRateManagement • T38FaxMaxBuffer • T38FaxMaxDatagram • T38FaxUdpEC	Mandatory	Yes
SIPt38-12	Cisco SIP-enabled gateways supporting T.38 interoperate with gateways from Cisco and other vendors.	Mandatory	Yes
SIPt38-13	Interoperability with gateways that support T.38 over H.323.	Optional	No

Requirement	Description	Mandatory or Optional	Supported?
SIPt38-14	Configuration of SIP enabled gateways include management of SIP T.38 specific configurable choices.	Mandatory	Yes. The following are configurable: • bitrate • TCP/UDP (UDP only) • hs and ls redundancy • ECM
SIPt38-15	Tracking and reporting of SIP T.38 activity on the gateways is desired. This includes generation of Call Detail Records (CDR) for SIP T.38 fax calls.	Mandatory	Yes
SIPt38-16	RFC 3261 security mechanisms apply. Message authentication can be performed on SIP Invite request and Bye requests.	Optional	No

SIP URL Comparison

When Uniform Resource Locators (URLs) are received, they are compared for equality. URL comparison can be done between two From SIP URLs or between two To SIP URLs. The order of the parameters does not need to match precisely. However, for two URLs to be equal, the user, password, host, and port parameters must match.

In Cisco IOS Release 12.3 and later releases, the maddr and transport parameters were removed and no longer used in Cisco SIP gateway implementations. However, in Cisco IOS Release 15.1(1)T and later releases, the maddr parameter is reintroduced so that the sender of a SIP request can specify a different destination for responses to those requests by specifying the maddr value for the URL in the Via header.

If a compared parameter is omitted or not present, it is matched on the basis of its default value. The table below shows a list of SIP URL compared parameters and their default values.

Table 20: SIP URL Compared Parameters and Default Values

SIP URL Compared Parameter	Default
User	--
Password	--
Host	Mandatory
Port	5060
User-param	IP

Assuming that a comparison is taking place, the following is an example of equivalent URLs:

Original URL:

```
sip:36602@172.18.193.120
```

Equivalent URLs:

```
sip:36602@172.18.193.120:
sip:36602@172.18.193.120;tag=499270-A62;pname=pvalue
sip:36602@172.18.193.120;user=ip
sip:36602@172.18.193.120:5060
```

487 Sent for BYE Requests

RFC 3261 requires that a UAS that receives a BYE request first send a response to any pending requests for that call before disconnecting. After receiving a BYE request, the UAS should respond with a 487 (Request Cancelled) status message.

3xx Redirection Responses

See the "Configuring SIP Redirect Processing Enhancement" section in the "Basic SIP Configuration" module in this guide.

DNS SRV Query Procedure

In accordance with RFC 3261, when a Request URI or the session target in the dial peer contains a fully qualified domain name (FQDN), the UAC needs to determine the protocol, port, and IP address of the endpoint before it forwards the request. SIP on Cisco gateways uses Domain Name System Server (DNS SRV) query to determine the protocol, port, and IP address of the user endpoint.

Before Cisco IOS Release 12.2(13)T, the DNS query procedure did not take into account the destination port.

A Time to Live (TTL) value of 3600 seconds is recommended for DNS SRV records. If you have to change the TTL value, the following equation must be true:

Where, A = Number of entries in the DNS SRV record B = Number of INVITE request retries configured using the **retry invite** command C = Waiting time for the SIP user agent configured using the **timers trying** command

CANCEL Request Route Header

A CANCEL message sent by a UAC on an initial INVITE request cannot have a Route header. Route headers cannot appear in a CANCEL message because they take the same path as INVITE requests, and INVITE requests cannot contain Route headers.

Interpret User Parameters

There are instances when the telephone-subscriber or user parameters can contain escaped characters to incorporate space, control characters, quotation marks, hash marks, and other characters. After the receipt of an INVITE message, the telephone-subscriber or user parameter is interpreted before dial-peer matching is done. For example, the escaped telephone number in an incoming INVITE message may appear as:

```
-%32%32%32
```

Although 222 is a valid telephone number, it requires interpretation. If the interpretation is not done, the call attempt fails when the user parameter is matched with the dial-peer destination pattern.

user=phone Parameter

A SIP URL identifies a user's address, which appears similar to an e-mail address. The form of the user's address is user@host where "user" is the user identification and " host" is either a domain name or a numeric network address. For example, the request line of an outgoing INVITE request might appear as:

```
INVITE sip:5550100@example.com
```

The user=phone parameter formerly required in a SIP URL is no longer necessary. However, if an incoming SIP message has a SIP URL with user=phone, user=phone is parsed and used in the subsequent messages of the transaction.

303 and 411 SIP Cause Codes

RFC 3261 obsoletes the SIP cause codes 303 *Redirection: See Other* and 411 *Client Error: Length required* .

Flexibility of Content-Type Header

The Content-Type header, which specifies the media type of the message body, is permitted to have an empty Session Description Protocol (SDP) body.

Optional SDP s= Line

The s= line in SDP is accepted as optional. The s= line describes the reason or subject for SDP information. Cisco SIP gateways can create messages with an s= line in SDP bodies and can accept messages that have no s= line.

Allow Header Addition to INVITEs and 2xx Responses

The use of the Allow header in an initial or re-INVITE request or in any 2*xx* class response to an INVITE is permitted. The Allow header lists the set of methods supported by the user agent that is generating the message. Because it advertises what methods should be invoked on the user agent sending the message, it avoids congesting the message traffic unnecessarily. The Allow header can contain any or all of the following: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, NOTIFY, INFO, SUBSCRIBE.

Simultaneous Cancel and 2xx Class Response

According to RFC 3261, if the UAC desires to end the call before a response is received to an INVITE, the UAC sends a CANCEL. However, if the CANCEL and a 2xx class response to the INVITE "pass on the wire," the UAC also receives a 2xx to the INVITE. When the two messages pass, the UAC terminates the call by sending a BYE request.

UPDATE-Request Processing

RFC 3261, which obsoletes RFC 2543, defines the SIP signaling protocol for creating, modifying and terminating sessions. The SIP Extensions for Caller Identity and Privacy feature provides support for the following SIP gateway implementations that are compliant with the RFC 3261specification:

SIP UPDATE Requests

SIP accomplishes session management through a series of messages that are either requests from a server or client, or responses to a request. SIP uses an INVITE request to initiate and modify sessions between user agents (UAs), and uses the ACK method to acknowledge a final response to an INVITE request. In some cases a session needs to be modified before the INVITE request is answered. This scenario occurs, for example, in a call that sends early media, the information sent to convey call progress during an established session, and for which the INVITE request has not been accepted. In this scenario either the caller or callee should be able to modify the characteristics of a session, for instance, by putting the early media on hold before the call is answered. Prior to the SIP UPDATE method, which allows a client to update session parameters, there was no mechanism to allow a caller or callee to provide updated session information before a final response to the initial INVITE request was generated. The SIP Extensions for Caller Identity and Privacy feature provides support for the UPDATE method and enables the gateway capability to receive and process, but not send, UPDATE requests. The gateway also updates the session timer value after the call is active.

A user agent client (UAC) initiates a session by sending an INVITE request to a user agent server (UAS). The UAS responds to the invitation by sending the following response codes:

- A 1xx provisional response indicating call progress. All 1xx responses are informational and are not final; all non-1xx responses are final.
- A 2xx response indicating successful completion or receipt of a request
- A 3xx, 4xx, 5xx, or 6xx response indicating rejection or failure.

A PRACK response is used to acknowledge receipt of a reliably transported provisional response, including a response with early media indication, while the ACK is used to acknowledge a final response to an INVITE request. A PRACK establishes an early dialog between UAC and UAS, a requirement to receive UPDATE requests with a new offer.

When a 2xx response is sent it establishes a session and also creates a dialog, or call leg. A dialog established by a 1xx response is considered an early dialog, whereas a final response establishes a confirmed dialog. The SIP UPDATE method allows a UAC to update session parameters, such as the set of media streams and their codecs, without affecting the dialog state. Unlike a re-INVITE request, a SIP UPDATE request may be sent to modify a session before the initial INVITE request is answered without impacting the dialog state itself. The UPDATE method is useful for updating session parameters within early dialogs before the initial INVITE request has been answered, for example, when early media is sent.

The SIP UPDATE method makes use of the offer and answer exchange using Session Description Protocol (SDP), as defined in the IETF specification, RFC 3264, *An Offer/Answer Model with the Session Description Protocol (SDP)* . One UA in the session generates an SDP message that constitutes the offer, that is, the set of media streams and codecs the UA wants to use, along with IP addresses and ports where the UA wants to receive the media. The other UA generates an answer, an SDP message responding to the offer.

In the Cisco SIP implementation, a UAS can receive an UPDATE request in both early and confirmed dialogs. The point at which the offer is generated, the UPDATE is received, the presence or absence of reliable provisional response and SDP, are all factors that determine how the gateway handles the UPDATE request. An UPDATE request generates a response indicating one of several possible outcomes:

- Success
- Pending response to outstanding offers
- Failure

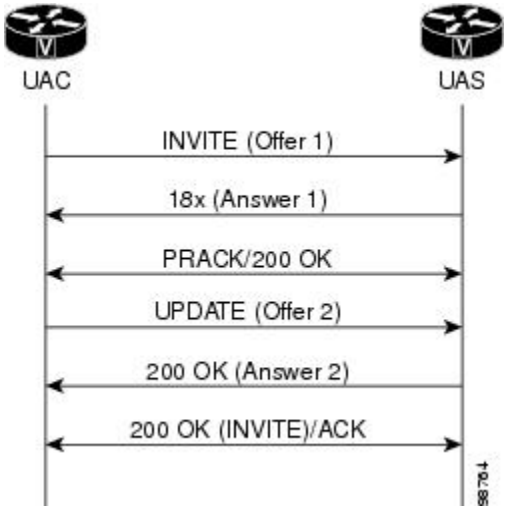
The following sections discuss how UPDATE requests are received and processed in various scenarios and call flows.

UPDATE Request Processing Before the Call Is Active

When the gateway sends a reliable provisional response with SDP, the response includes an Allow header that lists the UPDATE method and informs the caller of the gateway capability to support UPDATE processing.

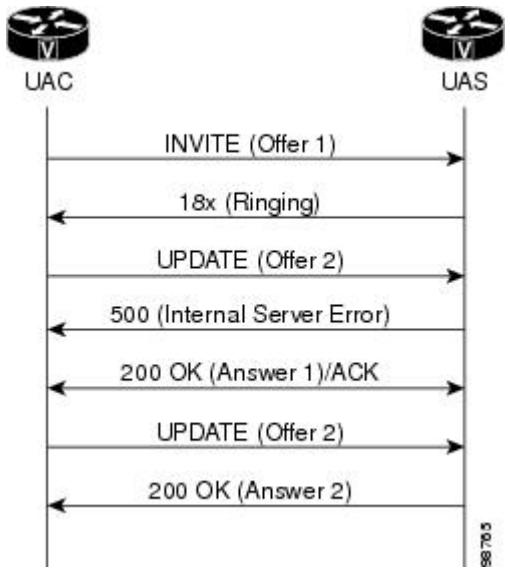
The figure below shows a call where the UAS sent a reliable provisional response (ANSWER 1) to an INVITE request (Offer 1). The18x early media response indicated the gateway capability to support UPDATEs. The UAC sent a provisional acknowledgement (PRACK) and received a 200 OK response to the PRACK request. The UAC requested the UAS modify the existing session media parameters of the early dialog by sending an UPDATE request (Offer 2). The UAS accepted Offer 2 by sending a 200 OK response. If media negotiation had failed, the UAS would have sent a 488 Unacceptable Media response instead. Later the UAS sent a 200 OK final response to the initial INVITE request. The UAS sent an ACK request acknowledging the final response to the INVITE request.

Figure 13: UPDATE for Early Media



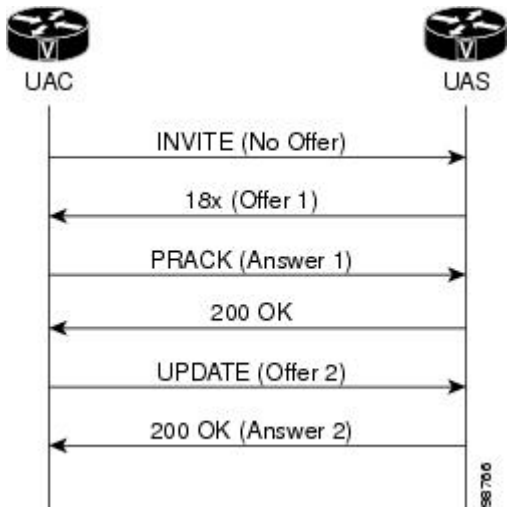
In the figure below, the gateway received an UPDATE (Offer 2) before responding to the INVITE request (Offer 1), causing the gateway to reject the request by sending a 500 Internal Server Error with a Retry-After header field set to a randomly chosen value between zero and ten seconds.

Figure 14: Initial UPDATE Rejected



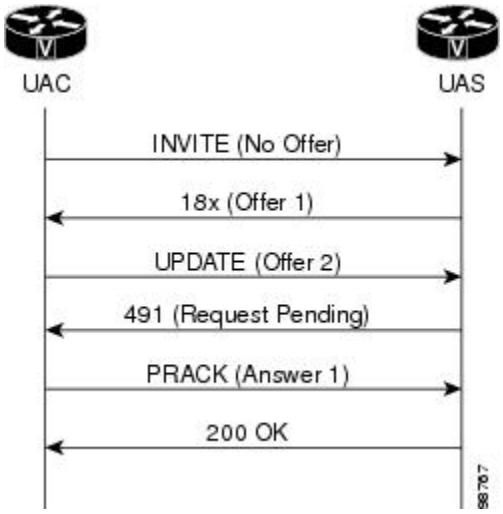
In the figure below, the initial INVITE request did not contain an offer, and the UAS gateway sent SDP with reliable provisional response (Offer 1) which was treated by the UAC as an offer.

Figure 15: UPDATE Request for Delayed Media



In the figure below, the UAS received an UPDATE request with an offer (Offer 2) before receiving a PRACK, that is, before the early dialog is established, causing the UAS (gateway) to generate a 491 Request Pending response.

Figure 16: UPDATE Request Failure for Delayed Media

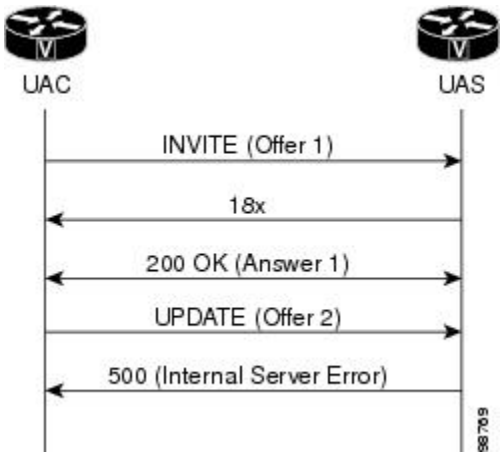


Error Responses to UPDATE Request Processing Before the Call Is Active

In other scenarios, additional rules apply to processing an UPDATE request with an offer when the gateway has sent a 200 OK response to an INVITE request but has not yet received an ACK. The following scenarios generate an error response and are shown in the figure below:

- If the initial INVITE request contains an offer but does not require provisional responses be sent reliably, then the SDP in the 200 OK is treated like an answer. If the UAS then receives an UPDATE request before an ACK response to the 200 OK, the UAS sends a 500 Server Internal error response with a Retry-After header.
- If the initial INVITE does not contain an offer and does not require provisional responses be sent reliably, then the SDP in the 200 OK is treated like an offer. If the UAS then receives an UPDATE request before receiving an ACK to the 200 OK, the UAS sends a 491 Request Pending response.

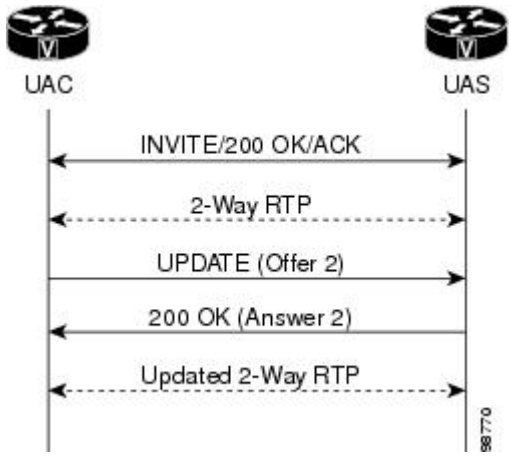
Figure 17: Error Cases for UPDATE Requests



UPDATE Request Processing in the Active State

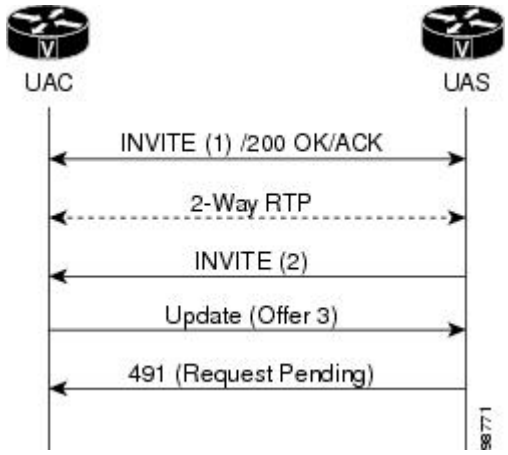
RFC 3261 recommends using a re-INVITE request, the SIP message that changes session parameters of an existing or pending call, to update session parameters after a call is active. UPDATEs received after a call is active are processed like a re-INVITE except that the 200 OK to update is not resent (see the figure below).

Figure 18: UPDATE Request in the Active State



The figure below shows a UAC that sent a mid-call INVITE request which has not yet been answered. In this state, when the gateway receives an UPDATE request with a new offer, it sends a 491 Request Pending error.

Figure 19: Error Response to an UPDATE Request in the Active State



Via Header Parameters and Merged Request Detection

To meet specifications of RFC 3261, the SIP Extensions for Caller Identity and Privacy feature provides support for the branch parameter in the Via header of a request, the information used to identify the transaction created by that request. The branch parameter value begins with the value “z9hG4bK” indicating that the request was generated by a UAC that is RFC 3261 compliant. The SIP Extensions for Caller Identity and Privacy feature also adds support for generating the received parameter with the received address.

The SIP Extensions for Caller Identity and Privacy feature uses the branch and sent-by parameters to detect a merged request, that is, a request that has arrived at the UAS more than once by following different paths. If the request has no tag in the To header field, the UAS checks the request against ongoing transactions. If

the From tag, Call-ID, and CSeq headers exactly match those headers associated with an ongoing transaction, but the topmost Via header, including the branch parameter, does not match, the UAS treats the request as merged. The UAS responds to a merged request with a 482 Loop Detected error.

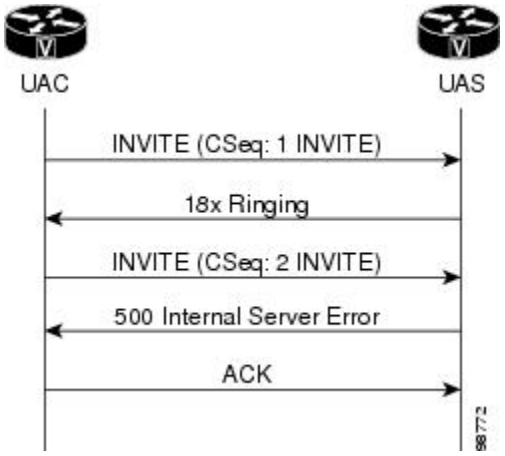
Loose-Routing and the Record-Route Header

The SIP Extensions for Caller Identity and Privacy feature supports loose-routing, a mechanism that helps keep the request target and next route destination separate. The lr parameter, used in the uniform resource indicator (URI) that a proxy places in the Record-Route header, indicates proxy compatibility with RFC 3261. If the lr parameter is missing from a request, the UA assumes the next-hop proxy implements strict-routing in compliance with RFC 2543, and reformats the message to preserve information in the Request-URI.

Multiple INVITE Requests Before a Final Response

This feature implements support for processing multiple INVITE requests received by the UAS before it sends a final response to the initial INVITE request (see the figure below). If the UAS gateway receives a second INVITE request before it sends the final response to the first INVITE request with a lower CSeq sequence number on the same dialog, the UAS returns a 500 Server Internal Error response to the second INVITE request. The error response also includes a Retry-After header field with a random value between 0 and 10 seconds.

Figure 20: Re-INVITE Request Rejected With a 5xx Response

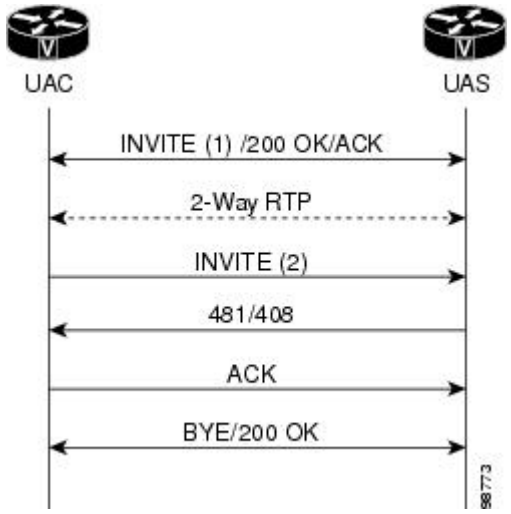


Mid-call Re-INVITE Request Failure

The SIP Extensions for Caller Identity and Privacy feature implements the mid-call re-INVITE request failure treatment shown in the figure below. The UAC terminates a dialog when a non-2xx final response to a mid-call INVITE request is one of the following:

- A 481 Call/Transaction Does Not Exist failure response
- A 408 Request Timeout failure response

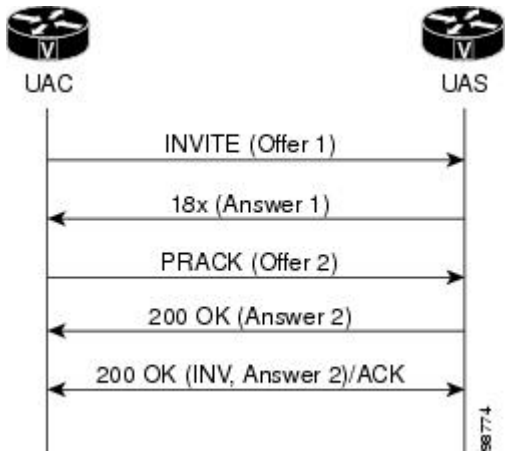
Figure 21: Dialog Termination After a 481 or 408 Response to Re-INVITE Request



PRACK Request with a New Offer

The SIP Extensions for Caller Identity and Privacy feature supports a PRACK request with a new offer (see the figure below). If the UAC receives a reliable provisional response with an answer (Answer 1), it may generate an additional offer in the PRACK (Offer 2). If the UAS receives a PRACK with an updated offer, it generates a 200 OK with an answer (Answer 2) if negotiation is successful. Otherwise the UAS generates a 488 Unacceptable Media response.

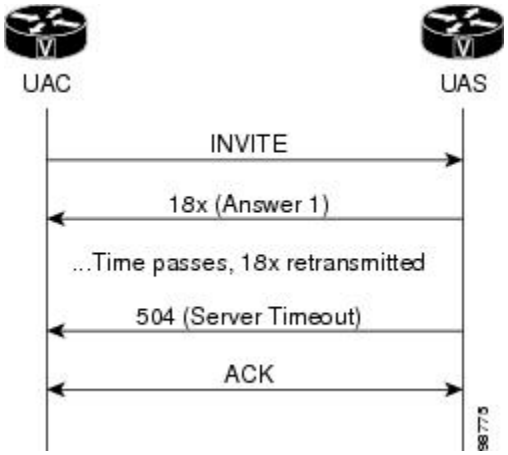
Figure 22: Offer in PRACK Accepted



Reliable Provisional Response Failure

The SIP Extensions for Caller Identity and Privacy feature provides the treatment shown in the figure below when the UAS does not receive a corresponding PRACK after resending a 18x reliable provisional response for the maximum number of retries allowed or for 32 seconds. The UAS generates a 5xx response to clear the call.

Figure 23: Reliable Provisional Response Failure



Sample Messages

This section contains sample SIP messages collected at the terminating SIP gateway.

SIP UPDATE Request Call Flow Example

The following example shows an exchange of SIP requests and responses, including an UPDATE request before the call is active:

```
1w0d:SIP Msg:ccsipDisplayMsg:Received:
INVITE sip:222@192.0.2.12:5060 SIP/2.0
Record-Route:<sip:222@192.0.2.4:5060;maddr=192.0.2.4>
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=5,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK1D38
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>
Date:Mon, 08 Apr 2002 16:58:08 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Supported:timer
```

The next line shows the UAC requires the provisional response be reliably transported.

```
Require:100rel
Min-SE: 1800
Cisco-Guid:2729535908-1246237142-2148443152-4064420637
User-Agent:Cisco-SIPGateway/IOS-12.x
```

The Allow header shows that the UPDATE method is supported.

```
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq:101 INVITE
Max-Forwards:70
Remote-Party-ID:<sip:111@192.0.2.14>;party=calling;screen=no;privacy=off
Timestamp:1018285088
Contact:<sip:111@192.0.2.14:5060>
Expires:180
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:262
```

The following SDP constitutes the initial offer, including media streams and codecs, along with IP addresses and ports to receive media.

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 6579 1987 IN IP4 192.0.2.14
s=SIP Call
c=IN IP4 192.0.2.14
t=0 0
m=audio 17782 RTP/AVP 8 0 18 19
c=IN IP4 192.0.2.14
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=5,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK1D38
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Sat, 07 Oct 2000 02:56:34 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Timestamp:1018285088
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow-Events:telephone-event
Content-Length:0
```

In the following lines, the gateway responds by sending early media in answer to the initial offer.

```
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 183 Session Progress
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=5,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK1D38
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Sat, 07 Oct 2000 02:56:34 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Timestamp:1018285088
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Require:100rel
RSeq:5785
Allow:UPDATE
Allow-Events:telephone-event
Contact:<sip:222@192.0.2.12:5060>
Record-Route:<sip:222@192.0.2.4:5060;maddr=192.0.2.4>
Content-Disposition:session;handling=required
Content-Type:application/sdp
Content-Length:191
v=0
o=CiscoSystemsSIP-GW-UserAgent 5565 7580 IN IP4 192.0.2.12
s=SIP Call
c=IN IP4 192.0.2.12
t=0 0
m=audio 18020 RTP/AVP 8 19
c=IN IP4 192.0.2.12
a=rtpmap:8 PCMA/8000
a=rtpmap:19 CN/8000
```

The following lines show the UAS receiving a PRACK for the 183 response.

```
1w0d:SIP Msg:ccsipDisplayMsg:Received:
PRACK sip:222@192.0.2.12:5060 SIP/2.0
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=6,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK40A
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Mon, 08 Apr 2002 16:58:08 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
CSeq:102 PRACK
RAck:5785 101 INVITE
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=6,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK40A
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Sat, 07 Oct 2000 02:56:34 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Server:Cisco-SIPGateway/IOS-12.x
CSeq:102 PRACK
Content-Length:0
```

The next lines show the UAS receiving an updated offer with different media streams and codecs.

```
1w0d:SIP Msg:ccsipDisplayMsg:Received:
UPDATE sip:222@192.0.2.12:5060 SIP/2.0
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=z9hG4bK10
Via:SIP/2.0/UDP 192.0.2.14:5060
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
CSeq:103 UPDATE
Contact:sip:111@192.0.2.14:5060
Content-Length:262
v=0
o=CiscoSystemsSIP-GW-UserAgent 6579 1987 IN IP4 192.0.2.14
s=SIP Call
c=IN IP4 192.0.2.14
t=0 0
m=audio 17782 RTP/AVP 8 0 18 19
c=IN IP4 192.0.2.14
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
```

The new offer in the UPDATE request is acceptable to the server, so it responds with the corresponding answer in the 200 OK message.

```
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=z9hG4bK10,SIP/2.0/UDP 192.0.2.14:5060
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Sat, 07 Oct 2000 02:56:34 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Server:Cisco-SIPGateway/IOS-12.x
CSeq:103 UPDATE
Content-Type:application/sdp
Content-Length:191
```

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 5565 7580 IN IP4 192.0.2.12
s=SIP Call
c=IN IP4 192.0.2.12
t=0 0
m=audio 18020 RTP/AVP 8 19
c=IN IP4 192.0.2.12
a=rtpmap:8 PCMA/8000
a=rtpmap:19 CN/8000
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=5,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK1D38
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Sat, 07 Oct 2000 02:56:34 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Timestamp:1018285088
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
Allow-Events:telephone-event
Contact:<sip:222@192.0.2.12:5060>
Record-Route:<sip:222@192.0.2.4:5060;maddr=192.0.2.4>
Content-Type:application/sdp
Content-Length:191
v=0
o=CiscoSystemsSIP-GW-UserAgent 5565 7580 IN IP4 192.0.2.12
s=SIP Call
c=IN IP4 192.0.2.12
t=0 0
m=audio 18020 RTP/AVP 8 19
c=IN IP4 192.0.2.12
a=rtpmap:8 PCMA/8000
a=rtpmap:19 CN/8000
1w0d:SIP Msg:ccsipDisplayMsg:Received:
ACK sip:222@192.0.2.12:5060 SIP/2.0
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=7,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK230
From:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
To:<sip:222@192.0.2.4>;tag=24D435A8-C29
Date:Mon, 08 Apr 2002 16:58:08 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Max-Forwards:70
CSeq:101 ACK
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
BYE sip:222@192.0.2.4:50605060;maddr=192.0.2.4 SIP/2.0
Via:SIP/2.0/UDP  192.0.2.12:5060;branch=z9hG4bKCA
From:<sip:222@192.0.2.4>;tag=24D435A8-C29
To:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
Date:Sat, 07 Oct 2000 02:56:35 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
User-Agent:Cisco-SIPGateway/IOS-12.x
Max-Forwards:70
Route:<sip:111@192.0.2.14:5060>
Timestamp:970887414
CSeq:101 BYE
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  192.0.2.12:5060;branch=z9hG4bKCA
From:<sip:222@192.0.2.4>;tag=24D435A8-C29
```

```
To:<sip:111@192.0.2.14>;tag=3DD33DE4-10DF
Date:Mon, 08 Apr 2002 16:58:29 GMT
Call-ID:A2B205CC-4A4811D6-8010A410-F242231D@192.0.2.14
Server:Cisco-SIPGateway/IOS-12.x
Timestamp:970887414
Content-Length:0
CSeq:101 BYE
```

Loose-Routing Call Flow Example

The following sample message shows a loose-routing request:

```
1w0d:SIP Msg:ccsipDisplayMsg:Received:
INVITE sip:222@192.0.2.12:5060 SIP/2.0
```

The SIP messages in the following call flow have the Request-URI set to the SIP URI of the destination UA instead of the SIP URI of the next-hop destination, that is, the SIP proxy server.

```
Record-Route:<sip:222@192.0.2.4:5060;lr;maddr=192.0.2.4>
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=9,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK2394
From:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
To:<sip:222@192.0.2.4>
Date:Mon, 08 Apr 2002 16:58:34 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Supported:timer
Min-SE: 1800
Cisco-Guid:2991015782-1246237142-2148770832-4064420637
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq:101 INVITE
Max-Forwards:70
Remote-Party-ID:<sip:111@192.0.2.14>;party=calling;screen=no;privacy=off
Timestamp:1018285114
Contact:<sip:111@192.0.2.14:5060>
Expires:180
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:262
v=0
o=CiscoSystemsSIP-GW-UserAgent 1981 1761 IN IP4 192.0.2.14
s=SIP Call
c=IN IP4 192.0.2.14
t=0 0
m=audio 18354 RTP/AVP 8 0 18 19
c=IN IP4 192.0.2.14
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=9,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK2394
From:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
To:<sip:222@192.0.2.4>;tag=24D49BE8-2346
Date:Sat, 07 Oct 2000 02:57:00 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Timestamp:1018285114
Server:Cisco-SIPGateway/IOS-12.x
```

```
CSeq:101 INVITE
Allow-Events:telephone-event
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=9,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK2394
From:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
To:<sip:222@192.0.2.4>;tag=24D49BE8-2346
Date:Sat, 07 Oct 2000 02:57:00 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Timestamp:1018285114
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow:UPDATE
Allow-Events:telephone-event
Contact:<sip:222@192.0.2.12:5060>
Record-Route:<sip:222@192.0.2.4:5060;lr;maddr=192.0.2.4>
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
SIP/2.0 200 OK
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=9,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK2394
From:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
To:<sip:222@192.0.2.4>;tag=24D49BE8-2346
Date:Sat, 07 Oct 2000 02:57:00 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Timestamp:1018285114
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
Allow-Events:telephone-event
Contact:<sip:222@192.0.2.12:5060>
Record-Route:<sip:222@192.0.2.4:5060;lr;maddr=192.0.2.4>
Content-Type:application/sdp
Content-Length:191
v=0
o=CiscoSystemsSIP-GW-UserAgent 5181 4737 IN IP4 192.0.2.12
s=SIP Call
c=IN IP4 192.0.2.12
t=0 0
m=audio 16720 RTP/AVP 8 19
c=IN IP4 192.0.2.12
a=rtpmap:8 PCMA/8000
a=rtpmap:19 CN/8000
1w0d:SIP Msg:ccsipDisplayMsg:Received:
ACK sip:222@192.0.2.12:5060 SIP/2.0
Via:SIP/2.0/UDP 192.0.2.4:5060;branch=10,SIP/2.0/UDP
192.0.2.14:5060;branch=z9hG4bK103D
From:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
To:<sip:222@192.0.2.4>;tag=24D49BE8-2346
Date:Mon, 08 Apr 2002 16:58:34 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Max-Forwards:70
CSeq:101 ACK
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Sent:
BYE sip:111@192.0.2.14:5060 SIP/2.0
Via:SIP/2.0/UDP  192.0.2.12:5060;branch=z9hG4bK18B6
From:<sip:222@192.0.2.4>;tag=24D49BE8-2346
To:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
Date:Sat, 07 Oct 2000 02:57:01 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
```

```
User-Agent:Cisco-SIPGateway/IOS-12.x
Max-Forwards:70
Route:<sip:222@192.0.2.4:5060;lr;maddr=192.0.2.4>
Timestamp:970887440
CSeq:101 BYE
Content-Length:0
1w0d:SIP Msg:ccsipDisplayMsg:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  192.0.2.12:5060;branch=z9hG4bK18B6
From:<sip:222@192.0.2.4>;tag=24D49BE8-2346
To:<sip:111@192.0.2.14>;tag=3DD3A404-12A3
Date:Mon, 08 Apr 2002 16:58:54 GMT
Call-ID:B2474766-4A4811D6-8015A410-F242231D@192.0.2.14
Server:Cisco-SIPGateway/IOS-12.x
Timestamp:970887440
Content-Length:0
CSeq:101 BYE
```

SIP RFC 3261 RFC 3262 and RFC 3264 Compliance

The Internet Engineering Task Force (IETF) continually updates SIP standards. This feature describes the specific updates or optimizations that were made on Cisco SIP gateways to remain in compliance with the IETF. The following standards have been updated:

- RFC 3261: Core Standard for SIP (obsoleting RFC 2543)
- RFC 3262: Standard for Reliability of Provisional Responses in SIP
- RFC 3264: Standard for Offer/Answer Model with Session Description Protocol (SDP)

To provide quality service to our SIP customers, Cisco optimizes its SIP gateways to comply with the latest SIP-related RFCs. In addition, backward compatibility is maintained, providing customers interoperability with gateways that do not yet support the current RFCs.

Note In compliance with RFC 4612, the inclusion of T38 parameters in SDP indicates that the parameters are supported and exclusion of these parameters indicate that the parameters are excluded. The SDP parameters are:

```
a=T38FaxTranscodingMMR
a=T38FaxFillBitRemoval
a=T38FixTranscodingJBIG
```

SIP Messaging Enhancements

The following changes or additions were made to SIP messaging:

- This feature is in compliance with RFC 3261. If a user agent server (UAS) generates a 2xx request and is waiting for an acknowledgement (ACK), and the call disconnects at the server side, the UAS does not send a BYE message immediately. The UAS sends a BYE message when the retry timer times out or when the ACK response is received. The BYE message terminates the call to prevent hung networks.
- In compliance with RFC 3261, the user agent (UA) cannot send a BYE message until it receives an ACK response from the originating gateway. This enhancement prevents a race condition, which is when a BYE response arrives at the terminating gateway before the 200 OK response. This enhancement applies to normal disconnects and not to disconnects due to timeouts or errors.

- In compliance with RFC 3262, the user agent client (UAC) now waits for a 1xx provisional response (PRACK) from the terminating gateway before sending a Cancel request to an Invite request. Waiting for a 1xx response prevents resources from being held up, which can happen if the Cancel request arrives at the terminating gateway before the Invite message.
- In compliance with RFC 3261, a Cisco SIP gateway returns a 491 Request Pending response when it receives an Invite requesting session modification on a dialog while an Invite request is still in progress. The gateway that sent the re-Invite and that receives the 491 response starts a timer with a randomly chosen value. When the timer expires, the gateway attempts the Invite request again if it still desires the session modification to take place.

If the UAC generated the request, the timer has a randomly chosen value between 2.1 and 4 seconds, in units of 10 ms. If the UAC did not generate the request, the timer has a randomly chosen value between 0 and 2 seconds, in units of 10 ms.

SIP TCP and UDP Connection Enhancements

Prior to RFC 3261, TCP support was optional for SIP user agents. RFC 3261 now requires support for both UDP and TCP. While Cisco SIP gateways already supported TCP, there have been several optimizations that are described below:

Failed Transmissions of 2xx Responses

The transmission of 2xx responses is in compliance with RFC 3261. If the transport is TCP and a gateway does not receive an acknowledgement to a 2xx response it sent to an INVITE message, the gateway retries the 2xx response over TCP. The retry ensures that a gateway receives a 200 OK message, eliminating the possibility that the 2xx response is lost when hops over the network use an unreliable transport such as UDP.

Reuse of TCP and UDP Connections

Prior to RFC 3261, a remote gateway could not initiate two requests over the same TCP connection. In addition, the gateway created a new connection for each new transaction, and after the completion of a transaction, the gateway closed the connection. Closing the connection, even if a subsequent request was destined for the same location as the previous transaction, resulted in potentially lower performance due to the large number of unnecessary open/close connections. With Cisco IOS Release 12.3(8)T, the gateway opens one TCP connection per remote IP address and port. The gateway opens a new connection only if a connection to the particular destination IP address and port is not already present. The gateway closes the connection when all requests that use that connection have terminated and no activity is detected for a given time period.

The **timers connection** command allows you to time out a TCP or UDP connection because of inactivity.

Transaction-Based Transport Switching and Usage

With Cisco IOS Release 12.3(8)T, if a new transaction request is larger than the threshold switchable value, it is sent over TCP. The threshold switchable value is a value that is 200 bytes or more than the interface or path's MTU. If the message size is smaller than the threshold switchable value, the original configured transport is used. The original transport means the transport configured under the dial peer for the initial Invite request or the transport specified in the incoming response's Contact or Record-Route headers in subsequent requests. In other words, the transport usage is now transaction-based instead of call-based.

Detection of Remote End Connection Closures

Remote gateway closures that go undetected can result in hung TCP connections. If a closed connection remains undetected, the corresponding connection entry is never removed from the connection table. Continuous

occurrences of undetected closures can lead to the connection table being filled with invalid entries and valid SIP requests being rejected, requiring a router reboot. With Cisco IOS Release 12.3(8)T, the SIP gateway uses internal mechanisms to detect remote closures and to clean up the connection table. No user input is required to initiate the cleanup.

Creation of New Connections for Sending Responses in Case the Original Connection Dropped

With Cisco IOS Release 12.3(8)T, if a gateway tears down the connection of an incoming request before a response is sent, the receiving gateway creates a new connection to send out a response. The new connection is based on the port specified in the sent-by parameter of the Via header. Prior to Cisco IOS Release 12.3(8)T, a dropped connection resulted in failure of the call.

Dynamic Transport Switching (UDP to TCP) for Large SIP Requests

RFC 3261 states that large SIP requests, requests within 200 bytes of the maximum transmission unit (MTU), should be transmitted over TCP. Transport over TCP avoids UDP fragmentation, and the switch to TCP can occur even if the gateway is configured to use UDP. If the TCP transmission fails (for example if the terminating gateway does not support TCP), the message is then retried over UDP.

The capability to configure the MTU size on an Ethernet or Fast Ethernet interface already exists on the Cisco SIP gateways. If the MTU is not configured, the default MTU value is 1500 bytes. Assuming an MTU of 1500 bytes, requests larger than 1300 bytes are considered the threshold value for dynamic transport switching.

Two commands allow the user to enable or disable support for dynamic switching. Use the commands to avoid interoperability issues with gateways that do not support TCP and to maintain backward compatibility. The **transport switch** command can be configured at the global level, and the **voice-class sip transport switch** command can be configured at the dial peer level. The global configuration is considered only when there is no matching VoIP dial peer.

This feature is disabled by default.

Call-Hold Enhancement

RFC 3264 recommends that call-hold be initiated using the direction attribute (a=sendonly) in SDP. Cisco POT-SIP gateways follow the new guideline, and these gateways can now initiate call-hold using either one of the two ways. The **offer call-hold** command allows the user to globally specify the format to initiate call-hold. That is, the gateway should use a=sendonly or conn addr=0.0.0.0; it cannot set usage to both. The default configuration is a=sendonly, because this is the RFC recommended method. Specifying a call-hold format is not available at the dial peer level.

Note Cisco POTS-SIP gateways support receiving call-hold requests in either of the two formats, but use of the direction attribute is recommended.

Expanded Range of the max-forwards Command

In compliance with RFC 3261, the **max-forwards** command was enhanced with a greater configurable range (1 to 70) and a higher default value (70).

How to Configure SIP RFC Compliance

Note For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring Compliance to RFC 2543

No configuration tasks are required to enable RFC 2543. It is enabled by default.

Configuring Compliance to RFC 2782

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **srv version** {**1** | **2**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**srv version** {**1**	**2**} **Example:** `Router(config-sip-ua)# srv version 2`	Generates DNS SRV queries with either RFC 2052 or RFC 2782 format. Keywords are as follows: • 1--Domain-name prefix of format protocol.transport. (RFC 2052 style)

	Command or Action	Purpose
		• 2--Domain-name prefix of format _protocol._transport. (RFC 2782 style) • Default: 2.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring Compliance to RFC 3261

No configuration tasks are required to enable RFC 3261. It is enabled by default.

Configuring Compliance to RFC 3261 RFC 3262 and RFC 3264

Configure SIP Messaging

No configuration is necessary.

Configure TCP and UDP Connection Enhancements

To set the time before the SIP UA ages out a TCP or UDP connection because of inactivity, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers connection aging** *timer-value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**timers connection aging** *timer-value* **Example:** `Router(config-sip-ua)# timers connection aging 5`	Sets the time before the SIP UA ages out a TCP or UDP connection because of inactivity. The argument is as follows: • *timer-value* -- Time, in minutes, to wait. Range: 5 to 30. Default: 5.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure Dynamic Transport Switching (UDP to TCP) for Large SIP Requests

RFC 3261 states that large SIP requests, within 200 bytes of the maximum transmission unit (MTU), should be transmitted over TCP. Transport over TCP avoids UDP fragmentation, and the switch to TCP can occur even if the gateway is configured to use UDP.

The configurations below describe setting the gateway to switch from UDP to TCP. The default MTU configuration of 1500 bytes on the interface is assumed. After configuration, the threshold value is 1300 bytes--that is, for all SIP requests over 1300 bytes, TCP is the transport mechanism.

You can configure dynamic transport switching on a dial-peer or global basis.

Configuring Dynamic Transport Switching for Large SIP Requests on a Dial-Peer Basis

To configure switching between UDP and TCP transport mechanisms for a specific dial peer, perform the following steps.

Note Dynamic transport switching from UDP to TCP is disabled by default.

- When the dynamic transport switching mechanism is enabled in dial-peer voice configuration mode, it takes precedence over the global configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip transport switch udp tcp**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 25 voip`	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 4	**voice-class sip transport switch udp tcp** **Example:** `Router(config-dial-peer)# voice-class sip transport switch udp tcp`	Enables switching between UDP and TCP transport mechanisms for large SIP messages for a specific dial peer. Keywords are as follows: • **udp** --Switching transport from UDP on the basis of the size of the SIP request being greater than the MTU size. • **tcp** --Switching transport to TCP.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Dynamic Transport Switching for Large SIP Requests on a Global Basis

To configure switching between UDP and TCP transport mechanisms on all the connections of a Cisco SIP gateway, perform the following steps.

Note Dynamic transport switching from UDP to TCP is disabled by default.

- When the dynamic transport switching mechanism is enabled in dial-peer voice configuration mode, it takes precedence over the global configuration. Consider the global configuration described below only when there is no matching VoIP dial peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **voice service voip**
4. **sip**
5. **transport switch udp tcp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode.
Step 4	**sip** **Example:** `Router(config-voi-srv)# sip`	Enters SIP configuration mode.
Step 5	**transport switch udp tcp** **Example:** `Router(conf-serv-sip)# transport switch udp tcp`	Enables switching between UDP and TCP transport mechanisms globally for large SIP messages. Keywords are as follows: • **udp** -- Switching transport from UDP based on the size of the SIP request being greater than the MTU size. • **tcp** --Switching transport to TCP.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

What to do next

Use the following commands to aid in verifying and troubleshooting the SIP transport and connection configurations:

- **debug ccsip transport**

- **show sip-ua connections**

To learn more about these commands as well as other verification and troubleshooting commands, see "Verifying SIP RFC Compliance" and "Troubleshooting Tips".

Configure Call-Hold

To specify how the POT-SIP gateway should initiate call-hold requests, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **offer call-hold** {**conn-addr** | **direction-attr**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**offer call-hold** {**conn-addr** | **direction-attr**} **Example:** `Router(config-sip-ua)# offer call-hold direction-attr`	Specifies how the POT-SIP gateway should initiate call-hold requests. Keywords are as follows: • **conn-addr** --RFC 2543/RFC 3261 method of using the connection address for initiating call-hold requests. Uses 0.0.0.0. • **direction-attr** --RFC 3264 method of using the direction attribute for initiating call-hold requests. Uses the direction attribute in SDP.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure Max Forwards

To set the maximum number of proxy or redirect servers that can forward the SIP request, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **max-forwards** *number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**max-forwards** *number* **Example:** `Router(config-sip-ua)# max-forwards 65`	Sets the maximum number of hops--that is, proxy or redirect servers that can forward the SIP request. The argument is as follows: • *number* -- Number of forwards. Range: 1 to 70. Default: 70.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Verifying SIP RFC Compliance

To verify SIP RFC compliance, perform the following steps as appropriate (commands are listed in alphabetical order).

Note A typical verification sequence involves use of one of the **show sip-ua connections**commands to view call statistics, followed by judicious use of the **clear sip-ua tcp connection** or **clear sip-ua udp connection** command to clear those statistics.

SUMMARY STEPS

1. **show sip-ua connections** {**tcp** [**tls**] | **udp**} {**brief** | **detail**}
2. **show sip-ua statistics**

DETAILED STEPS

Step 1 show sip-ua connections {tcp [tls] | udp} {brief | detail}

Use this command, after a call is made, to learn connection details.

The following sample output shows multiple calls to multiple destinations. This example shows UDP details, but the command output looks identical for TCP calls.

Example:

```
Router# show sip-ua connections udp detail
Total active connections : 2
No. of send failures : 0
No. of remote closures : 0
No. of conn. failures : 0
No. of inactive conn. ageouts : 0
---------Printing Detailed Connection Report---------
Note:
** Tuples with no matching socket entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port>'
to overcome this error condition
++ Tuples with mismatched address/port entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port> id <connid>'
to overcome this error condition
Remote-Agent:172.18.194.183, Connections-Count:1
Remote-Port Conn-Id Conn-State WriteQ-Size
=========== ======= =========== ===========
5060 1 Established 0
Remote-Agent:172.19.154.18, Connections-Count:1
Remote-Port Conn-Id Conn-State WriteQ-Size
=========== ======= =========== ===========
5060 2 Established 0
```

The following sample output shows sequential display and clearing of call statistics for connection to a particular target (in this case, 172.18.194.183, port 5060).

Caution Take care when you use the **clear** commands. Inappropriate usage without understanding the issue or the implications can lead to erroneous call behavior, inappropriate usage of connections, and call failures.

1. Output for the **show sip-ua connections**command displays call statistics:

Example:

```
Router# show sip-ua connections tcp detail
```

```
Total active connections : 1
No. of send failures : 0
No. of remote closures : 0
No. of conn. failures : 0
No. of inactive conn. ageouts : 0
Max. tcp send msg queue size of 1, recorded for 172.18.194.183:5060
---------Printing Detailed Connection Report---------
Note:
** Tuples with no matching socket entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port>'
to overcome this error condition
++ Tuples with mismatched address/port entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port> id <connid>'
to overcome this error condition
Remote-Agent:172.18.194.183, Connections-Count:1
Remote-Port Conn-Id Conn-State WriteQ-Size
=========== ======= =========== ===========
5060 1 Established 0
```

1. Output for the **clear sip-ua tcp connection** command shows that statistics are being cleared:

Example:

```
Router# clear sip-ua tcp connection id 1 target ipv4:172.18.194.183:5060
Purging the entry from sip tcp process
Purging the entry from reusable global connection table
```

1. Output for the **show sip-ua connections** command verifies that all connections are cleared as expected:

Example:

```
Router# show sip-ua connections tcp detail
Total active connections : 0
No. of send failures : 0
No. of remote closures : 0
No. of conn. failures : 0
No. of inactive conn. ageouts : 0
Max. tcp send msg queue size of 1, recorded for 172.18.194.183:5060
---------Printing Detailed Connection Report---------
Note:
** Tuples with no matching socket entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port>'
to overcome this error condition
++ Tuples with mismatched address/port entry
- Do 'clear sip <tcp/udp> conn t ipv4:<addr>:<port> id <connid>'
to overcome this error condition
Remote-Agent:172.18.194.183, Connections-Count:0
```

Step 2 **show sip-ua statistics**

Use this command to display SIP statistics, including UPDATE requests.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational
     Trying 1/4, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 1/4
    Success:
      OkInvite 1/2, OkBye 1/2,
```

```
      OkCancel 0/2, OkOptions 0/0,
      OkPrack 1/4, OkPreconditionMet 0/0,
      OkSubscribe 0/0, OkNotify 0/0,
      OkInfo 0/0, 202Accepted 0/0,
      OkUpdate 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, UseProxy 0,
      AlternateService 0
    Client Error:
      BadRequest 0/0, 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,
      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, RequestCancel 0/2,
      NotAcceptableMedia 0/0, BadEvent 0/0,
     SETooSmall 0/0, RequestPending 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 2/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
    Miscellaneous counters:
      RedirectRspMappedToClientErr 0
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 4/4, Ack 4/3, Bye 2/1,
    Cancel 2/0, Options 0/0,
    Prack 4/1, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0, Info 0/0,
    Update 0/0
Retry Statistics
    Invite 1, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0, Notify 0
SDP application statistics:
 Parses: 6, Builds 10
 Invalid token order: 0, Invalid param: 0
 Not SDP desc: 0, No resource: 0
Last time SIP Statistics were cleared: <never>
```

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the “General Troubleshooting Tips” section in the “Basic SIP Configuration” module in this guide.

- Use the **debug ccsip all** command to enable SIP-related debugging.

- Use the **debug ccsip transport**command to debug transport and connection related operations while sending out an Invite Message.

Sample output of some of these commands is shown below:

Sample Output for the debug ccsip transport Command

The operations captured here show the following:

- That the connection is established and the Invite was sent.
- That UDP is the transport of the initial Invite message.
- Remote target details; that is where the request is to be sent.
- That the size of the message exceeded the threshold size of the MTU. Therefore transport switching (from UDP to TCP) is enabled.
- That the connection algorithm is started; that is, the counter starts to age out the TCP or UDP connection if inactivity occurs.

```
Router# debug ccsip transport
.
.
.
1w1d: //18/8E16980D800A/SIP/Transport/sipSPISendInvite: Sending Invite to the transport
layer
1w1d: //18/8E16980D800A/SIP/Transport/sipSPIGetSwitchTransportFlag: Return the Global
configuration, Switch Transport is TRUE
1w1d: //18/8E16980D800A/SIP/Transport/sipSPITransportSendMessage: msg=0x64082D50,
addr=172.18.194.183, port=5060, sentBy_port=0, is_req=1, transport=1, switch=1,
callBack=0x614FAB58
1w1d: //18/8E16980D800A/SIP/Transport/sipSPITransportSendMessage: Proceedable for sending
msg immediately
1w1d: //18/8E16980D800A/SIP/Transport/sipTransportLogicSendMsg: switch transport is 1
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportGetInterfaceMtuSize: MTU size for remote
 address 172.18.194.183 is 500
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportVerifyMsgForMTUThreshold: Interface MTU
Size 500, Msg Size 1096
1w1d: //18/8E16980D800A/SIP/Transport/sipTransportLogicSendMsg: Switching msg=0x64082D50
transport UDP->TCP
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportSetAgeingTimer: Aging timer initiated for
 holder=0x64084058,addr=172.18.194.183
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipCreateConnHolder: Created new holder=0x64084058,
addr=172.18.194.183
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportPostRequestConnection: Posting TCP conn
create request for addr=172.18.194.183, port=5060, context=0x64128D5C
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportSetConnWaitTimer: Wait timer set for
connection=0x64129BF4,addr=172.18.194.183, port=5060
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipCreateConnInstance: Created new initiated
conn=0x64129BF4, connid=-1, addr=172.18.194.183, port=5060, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipConnectionManagerProcessConnCreated:
gConnTab=0x64128D5C, addr=172.18.194.183, port=5060, connid=1, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipInstanceHandleConnectionCreated: Moving
connection=0x64129BF4, connid=1state to pending
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportProcessNWConnectionCreated:
context=0x64128D5C
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipConnectionManagerProcessConnCreated:
gConnTab=0x64128D5C, addr=172.18.194.183, port=5060, connid=1, transport=tcp
1w1d: //-1/xxxxxxxxxxxx/SIP/Transport/sipTransportPostSendMessage: Posting send for
msg=0x64082D50, addr=172.18.194.183, port=5060, connId=1 for TCP
```

.
.
.

Configuration Examples for SIP RFC Compliance

Note IP addresses and hostnames in examples are fictitious.

SIP Gateway Compliance to RFC 3261 RFC 3262 and RFC 3264 Example

This section provides a configuration example to match the identified configuration tasks in the previous sections.

```
1w1d: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...
Current configuration : 3326 bytes
!
!Last configuration change at 18:09:20 EDT Fri Apr 23 2004
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
boot-start-marker
boot system tftp mantis/c3640-is-mz.disc_w_pi 172.18.207.10
boot-end-marker
!
clock timezone EST -5
clock summer-time EDT recurring
voice-card 3
!
aaa new-model
!
aaa accounting connection h323 start-stop group radius
aaa nas port extended
aaa session-id common
ip subnet-zero
!
ip cef
ip host example.com 172.18.194.183
ip host CALLGEN-SECURITY-V2 10.36.54.81 10.1.0.0
ip name-server 172.18.192.48
!
isdn switch-type primary-ni
!
trunk group 1
!
voice service voip
 sip
 rel1xx require "100rel"
 transport switch udp tcp
!
voice class uri 800 sip
pattern test@example.com
!
```

```
controller T1 3/0
 framing sf
 linecode ami
 pri-group timeslots 1-24
!
controller T1 3/1
 framing sf
 linecode ami
 pri-group timeslots 1-24
 gw-accounting aaa
!
interface Ethernet0/0
 description CentreComm Hub port 9 in PP070
 ip address 172.18.194.170 255.255.255.0
 no ip proxy-arp
 ip mtu 500
 half-duplex
 no cdp enable
 ip rsvp bandwidth 100 100
!
interface Serial3/0:23
 no ip address
 no logging event link-status
 isdn switch-type primary-ni
 isdn incoming-voice voice
 no cdp enable
!
interface Serial3/1:23
 no ip address
 no logging event link-status
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice voice
 no cdp enable
!
no ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.194.1
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.194.1
ip route 172.16.0.0 255.0.0.0 Ethernet0/0
!
dialer-list 1 protocol ip permit
no cdp run
!
radius-server host 10.13.84.133 auth-port 1645 acct-port 1646
radius-server timeout 2
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
call application voice testapp79 tftp://172.18.207.10/mantis/my_app.tcl
call application voice testapp888 tftp://172.18.207.10/mantis/AL_FEAT_SIP_URL_O_RV_79.tcl
call application voice testapp888 mcid-dtmf 9876
call application voice testapp888 test 5444
!
voice-port 1/1/0
!
voice-port 1/1/1
!
voice-port 3/0:23
!
```

```
voice-port 3/1:23
!
dial-peer cor custom
!
dial-peer voice 9876 voip
 destination-pattern 9876
 voice-class sip transport switch udp tcp
 session protocol sipv2
 session target ipv4:172.18.194.183
 session transport udp
!
dial-peer voice 222 pots
 incoming called-number .
 direct-inward-dial
!
sip-ua
 max-forwards 65
 retry invite 4
 retry bye 4
 retry cancel 4
 retry comet 4
 retry notify 4
 timers connection aging 15
 offer call-hold conn-addr
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password password1
ntp clock-period 17179695
ntp server 172.18.194.178
ntp server 10.81.254.131
!
end
```

Additional References

The following sections provide references related to the Achieving SIP RFC Compliance.

Related Documents

Related Topic	Document Title
Cisco IOS SIP Configuration Guide , "Overview of SIP" module	http://www.cisco.com/en/US/docs/ios/voice/sip/configuration/guide/sip_cg-overview.html
Cisco IOS Tcl IVR and VoiceXML Application Guide	Cisco IOS Release 12.3(14)T and later: http://www.cisco.com/en/US/docs/ios/voice/ivr/configuration/guide/tcl_c.html
	Cisco IOS releases prior to 12.3(14)T: http://www.cisco.com/en/US/docs/ios/voice/ivr/pre12.3_14_t/configuration/guide/ivrapp.pdf

Related Topic	Document Title
Cisco IOS Voice Command Reference	http://www.cisco.com/en/US/docs/ios/voice/command/reference/vr_book.html
Cisco Unified Communications Manager Express Command Reference	http://www.cisco.com/en/US/docs/voice_ip_comm/cucme/command/reference/cme_cr.html
Cisco Unified Communications Manager Express support documentation	http://www.cisco.com/en/US/products/sw/voicesw/ps4625/tsd_products_support_series_home.html
Cisco Unified SIP SRST System Administrator Guide	http://www.cisco.com/en/US/docs/voice_ip_comm/cusrst/admin/sccp_sip_srst/configuration/guide/srst_
Cisco VoiceXML Programmer's Guide	http://www.cisco.com/en/US/docs/ios/voice/vxml/developer/guide/vxmlprg.html
SIP Gateway Support of RSVP and TEL URL	http://www.cisco.com/en/US/docs/ios/12_2t/12_2t11/feature/guide/vvfresrv.html
Tcl IVR API Version 2.0 Programming Guide	http://www.cisco.com/en/US/docs/ios/voice/tcl/developer/guide/tclivrv2.html

Standards

Standard	Title
International Organization for Standardization (ISO) specification, ISO 639	Codes for Representation of Names of Languages

MIBs

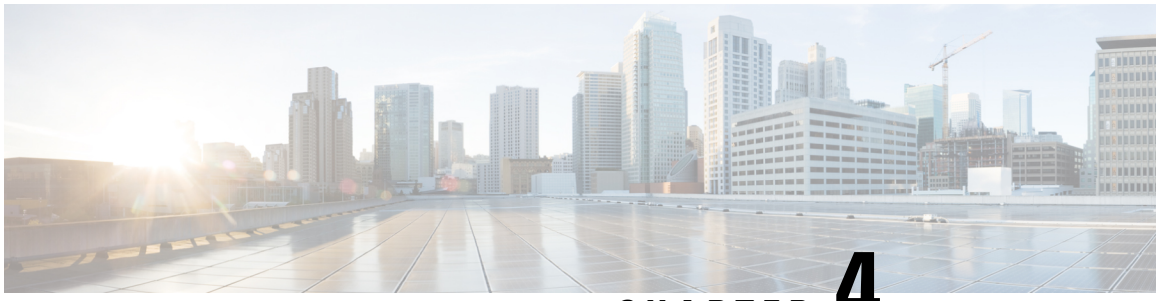
MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2833	RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
RFC 3261	SIP: Session Initiation Protocol
RFC 3262	Reliability of Provisional Responses in the Session Initiation Protocol (SIP)
RFC 3264	An Offer/Answer Model with the Session Description Protocol (SDP)
RFC 3265	Session Initiation Protocol (SIP)-Specific Event Notification
RFC 3312	Integration of Resource Management and Session Initiation Protocol (SIP)
RFC 3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)
RFC 3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
RFC 3326	The Reason Header Field for the Session Initiation Protocol (SIP)
RFC 4028	Session Timers in the Session Initiation Protocol (SIP)
RFC 4244	An Extension to the Session Initiation Protocol (SIP) for Request History Information

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 4

Configuring SIP Call-Transfer Features

This chapter describes how to configure SIP call-transfer features. It describes the following features:

- SIP - Call Transfer Using Refer Method
- SIP - Call Transfer Enhancements Using Refer Method
- SIP Transfer Using the Refer Method and Call Forwarding
- SIP Stack Portability

Note The SIP Stack Portability feature is described in the "Configuring SIP, Timer, and Response Features" chapter.

Feature History for SIP - Call Transfer Using Refer Method

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB2	The feature was implemented on an additional platform.
12.2(11)T	The feature was integrated into this release and support was added for additional platforms.

Feature History for SIP - Call Transfer Enhancements Using Refer Method

Release	Modification
12.2(13)T	This feature was introduced.

Feature History for SIP Transfer Using the Refer Method and Call Forwarding

Release	Modification
12.2(13)T	This feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Call Transfer

All SIP Call-Transfer Features

- Establish a working IP network and configure VoIP.

Note For information about configuring VoIP, see "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms".

- Ensure that the gateway has voice functionality configured for SIP.
- Ensure that your Cisco router has minimum memory requirements.
- With all SIP call-transfer methods, configure dial peers for correct functioning of the Refer method.

Note For dial-peer configuration steps, see the "Configure SIP Call Transfer on a POTS Dial Peer".

- As necessary, configure the router to use Greenwich Mean Time (GMT). SIP requires that all times be sent in GMT. The INVITE is sent in GMT. However, the default for routers is to use Coordinated Universal Time (UTC). To configure the router to use GMT, issue the **clock timezone** command in global configuration mode and specify GMT.

SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Feature

- Load Cisco IOS Release 12.2(15)T or a later release.
- Configure hookflash signaling.

- Write a Tool Command Language (Tcl) Interactive Voice Response (IVR) 2.0 script that implements Cisco IOS call-transfer and call-forward functionality.

Restrictions for SIP Call Transfer

All SIP Call-Transfer Features

- The SIP gateway does not support codecs other than those listed in the table titled “SIP Codec Support by Platform and Cisco IOS Release” in the “Enhanced Codec Support for SIP Using Dynamic Payloads” section of the “Configuring SIP QoS Features” document.
- SIP requires that all times be sent in GMT.
- Although SIP Cisco IOS gateways currently support SIP URLs and TEL URLs, the Refer-To header and the Also header must be in SIP URL format to be valid. The TEL URL is only supported in the Refer-To header for blind transfer. The TEL URL format cannot be used because it does not provide a host portion, and without one, the triggered Invite request cannot be routed.
- Only three overloaded headers in the Refer-to header are accepted: Accept-Contact, Proxy-Authorization, and Replaces. All other headers present in the Refer-To are ignored.
- The Refer-To and Contact headers are required in the Refer request. The absence of either header results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4*xx* class response.
- The Referred-By header is required in a Refer request. The absence of this header results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4*xx* class response.
- With all SIP call-transfer methods, dial peers must be configured for correct functioning of the Refer method.

Note For dial-peer configuration steps, see "Configure SIP Call Transfer on a POTS Dial Peer".

- With call transfer using the Bye method, the Requested-By header identifies the party initiating the transfer. The Requested-By header is included in the INVITE request that is sent to the transferred-to party only if a Requested-By header was also included in the Bye request.
- With call transfer using the Also method, the Also header identifies the transferred-to party. To invoke a transfer, the user portion of the Also header must be defined explicitly or with wildcards as a destination pattern on a VoIP dial peer. The transferred call is routed using the session target parameter on the dial peer instead of the host portion of the Also header. Therefore, the Also header can contain user@host, but the host portion is ignored for call routing purposes.
- The grammar for the Also and Requested-By headers is not fully supported. Only the name-addr is supported. This implies that the crypto-param, which might be present in the Bye request, is not populated in the ensuing Invite to the transferred-to party.
- Cisco SIP gateways do not support the “user=np-queried” parameter in a Request URI.

- If a Cisco SIP gateway receives an ISDN Progress message, it generates a 183 Session progress message. If the gateway receives an ISDN ALERT, it generates a 180 Ringing message.
- The SIP gateway requires each INVITE to include a Session Description Protocol (SDP) header.
- The contents of the SDP header cannot change between the 180 Ringing message and the 200 OK message.
- VoIP dial peers allow a user to configure the bytes parameter associated with a codec. Cisco SIP gateways present or respond to the a=ptime parameter in the SDP body of a SIP message. However, only one a=ptime attribute is allowed per m-line block.
- If early transfer is attempted, and the call between the originator and final-recipient involves QoS or RSVP, the triggered Invite from the recipient with the Replaces header is not processed and the transfer fails. The session between the originator and the final-recipient remains unchanged.

SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Feature

- SIP call transfer and call forwarding using Tcl IVR 2.0 and VoiceXML applications feature is supported only through Tcl IVR 2.0 and VoiceXML applications; the feature is not supported for Tcl IVR 1.0 applications or the DEFAULT session application.
- Only Cisco 1700 series, Cisco 2600 series, and Cisco 3600 series routers support the initiating of call transfer and call redirection.
- Cisco SIP customer premise equipment (CPE) such as 79xx and Analog Telephone Adaptors (ATAs) do not currently support TEL URLs.
- RLT on CAS or analog (FXS) ports are necessary to initiate SIP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS, FXO, T1, E1, and CAS phones are not supported.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

Information About SIP Call Transfer

SIP Call-Transfer Basics

Basic Terminology of SIP Call Transfer

The Refer method provides call-transfer capabilities to supplement the Bye and Also methods already implemented on Cisco IOS SIP gateways.

Call transfer allows a wide variety of decentralized multiparty call operations. These decentralized call operations form the basis for third-party call control, and thus are important features for VoIP and SIP. Call

transfer is also critical for conference calling, where calls can transition smoothly between multiple point-to-point links and IP level multicasting.

Refer Method

The SIP Refer method provides call-transfer capabilities to supplement the Bye and Also methods already implemented on Cisco IOS SIP gateways. The Refer method has three main roles:

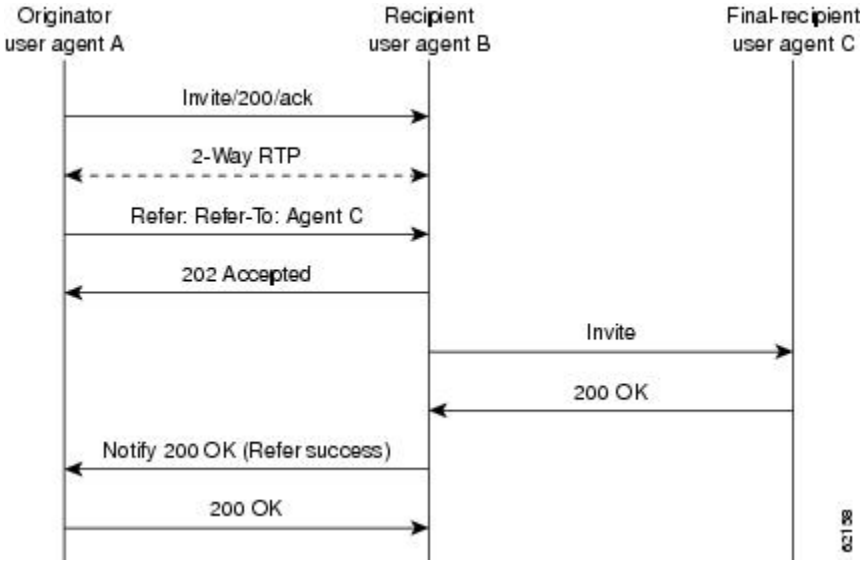
- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

Note A gateway can be a recipient or final-recipient; but not an originator.

The Refer method always begins within the context of an existing call and starts with the *originator* . The originator sends a Refer request to the *recipient* (user agent receiving the Refer request) to initiate a triggered Invite request. The triggered Invite request uses the SIP URL contained in the Refer-To header as the destination of the Invite request. The recipient then contacts the resource in the Refer-To header (*final-recipient*), and returns a SIP 202 (Accepted) response to the originator. The recipient also must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify Method, SIP's event notification mechanism. A Notify message with a message body of SIP 200 OK indicates a successful transfer, while a body of SIP 503 Service Unavailable indicates an unsuccessful transfer. If the call was successful, a call between the recipient and the final-recipient results.

The figure below shows the call flow of a successful Refer transaction initiated within the context of an existing call.

Figure 24: Successful Refer Transaction



Refer-To Header

The recipient receives from the originator a Refer request that always contains a single Refer-to header. The Refer-to header includes a SIP URL that indicates the party to invite and must be in SIP URL format.

Note The TEL URL format cannot be used in a Refer-to header, because it does not provide a host portion, and without one, the triggered Invite request cannot be routed.

The Refer-To header may contain three additional overloaded headers to form the triggered Invite request. If any of these three headers are present, they are included in the triggered Invite request. The three headers are:

- Accept-Contact--Optional in a Refer request. A SIP IOS gateway that receives an Invite request with an Accept-Contact does not act upon this header. This header is defined in draft-ietf-sip-callerprefs-03.txt and may be used by user agents that support caller preferences.
- Proxy-Authorization--A nonstandard header that SIP gateways do not act on. It is echoed in the triggered Invite request because proxies occasionally require it for billing purposes.
- Replaces--The Replaces header is used by SIP gateways to indicate whether the originator of the Refer request is requesting a blind or attended transfer. It is required if the originator is performing an attended transfer, and not required for a blind transfer.

All other headers present in the Refer-To are ignored, and are not sent in the triggered invite.

Note The Refer-To and Contact headers are required in the Refer request. The absence of these headers results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4*xx* class response.

Referred-By Header

The Referred-By header is required in a Refer request. It identifies the originator and may also contain a signature (included for security purposes). SIP gateways echo the contents of the Referred-By header in the triggered Invite request, but on receiving an Invite request with this header, gateways do not act on it.

Note The Referred-By header is required in a Refer request. The absence of this header results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4*xx* class response.

Notify Method

Once the outcome of the Refer transaction is known, the recipient of the Refer request must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify method, SIP's event notification mechanism. The notification contains a message body with a SIP response status line and the response class in the status line indicates the success or failure of the Refer transaction.

The Notify message must:

- Reflect the same To, From, and Call-ID headers that were received in the Refer request.
- Contain an Event header refer.
- Contain a message body with a SIP response line. For example: SIP/2.0 200 OK to report a successful Refer transaction, or SIP/2.0 503 Service Unavailable to report a failure. To report that the recipient disconnected before the transfer finished, it must use SIP/2.0 487 Request Canceled.

Two Cisco IOS commands pertain to the Notify method.

- The **timers notify** command sets the amount of time that the recipient should wait before retransmitting a Notify message to the originator.
- The **retry notify** command configures the number of times a Notify message is retransmitted to the originator.

Note For information on these commands, see the *Cisco IOS Voice Command Reference*.

Types of SIP Call Transfer Using the Refer Method

This section discusses how the Refer method facilitates call transfer.

There are two types of call transfer: blind and attended. The primary difference between the two is that the Replaces header is used in attended call transfers. The Replaces header is interpreted by the final-recipient and contains a Call-ID header, indicating that the initial call leg is to be replaced with the incoming Invite request.

As outlined in the Refer method, there are three main roles:

- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

A gateway can be a recipient or final-recipient but not an originator.

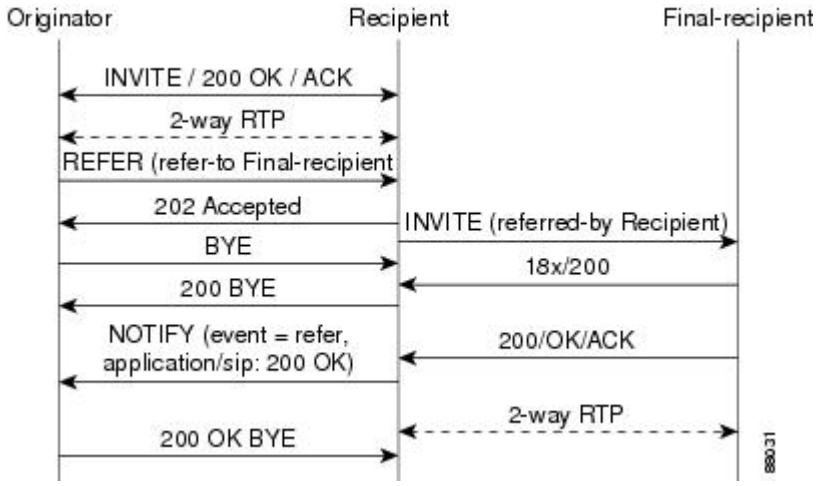
Blind Call-Transfer Process

A blind, or unattended, transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. This is different from a consultative, or attended, transfer in which one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller to the third party. Blind transfers are often preferred by automated devices that do not have the capability to make consultation calls.

The basic process of blind transfers works as described in the figure below. In blind transfer, the originator (user agent that initiates the transfer or Refer request) sets up a call with the recipient (user agent that receives the Refer request). After the originator issues a Refer request to the recipient, the recipient, triggered by the Refer request, sends an Invite request to the final-recipient (user agent introduced into a call with the recipient). The recipient returns a SIP 202 (Accepted) response to the originator, and notifies the originator of the outcome of the Refer transaction--if the final-recipient was successfully (SIP 200 OK) or unsuccessfully (SIP 503 Service Unavailable) contacted.

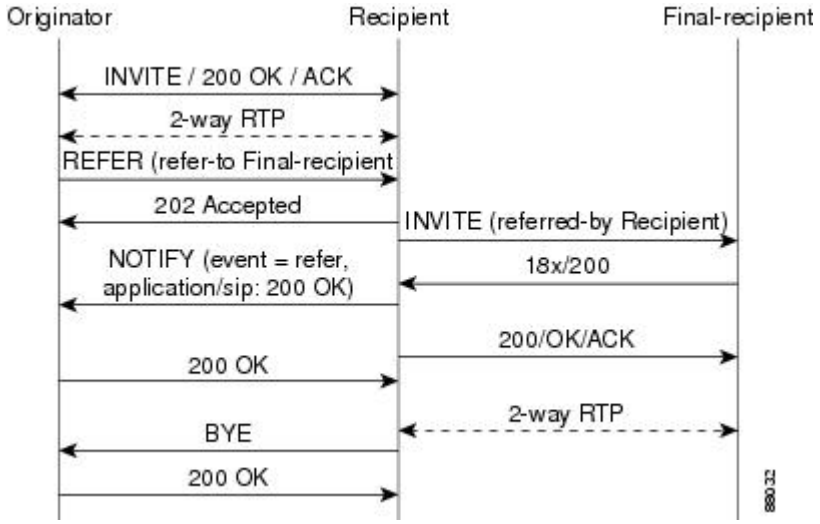
If successful, a call is established between the recipient and the final-recipient. The original signaling relationship between the originator and recipient is terminated when a Bye request is sent by one of the parties. On a successful transfer, if the originator does not send a Bye request after receiving an acknowledgement for the Notify message, the recipient initiates a Bye request. The figure below shows a successful blind or unattended call transfer in which the originator initiates a Bye request to terminate signaling with the recipient.

Figure 25: Successful Blind or Unattended Transfer--Originator Initiating a Bye Request



The figure below shows a successful blind or unattended call transfer in which the recipient initiates a Bye request to terminate signaling with the originator. A Notify message is always sent by the recipient to the originator after the final outcome of the call is known.

Figure 26: Successful Blind or Unattended Transfer --Recipient Initiating a Bye Request

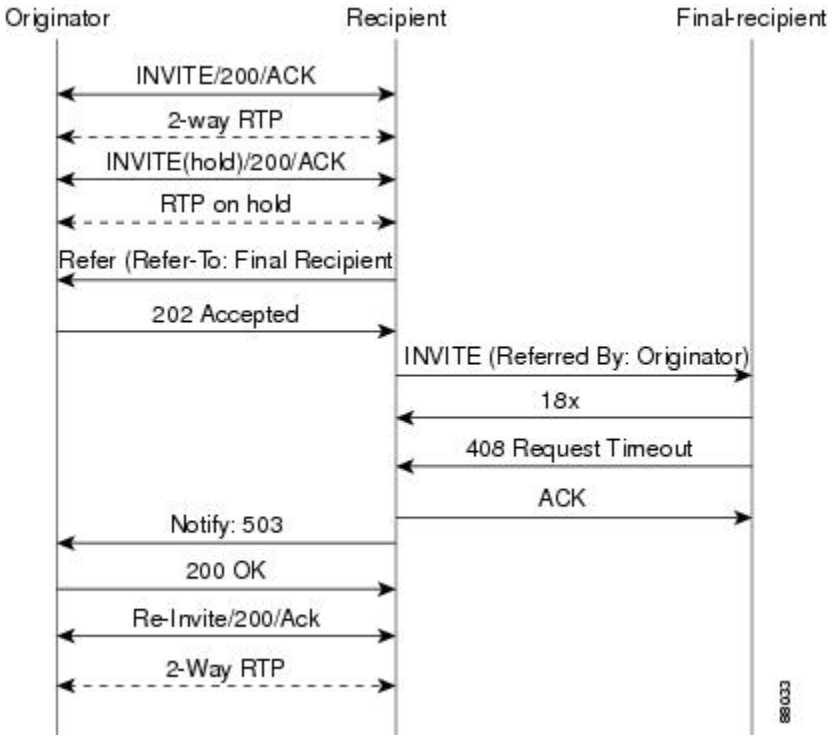


If a failure occurs with the triggered Invite to the final-recipient, the call between the originator and the recipient is not disconnected. The originator sends a re-Invite which takes the call off hold and returns to the original call with the recipient. With prior blind transfer functionality, if the recipient receives an 18x informational response from the final-recipient and then the call fails, the originator can not recover the call with the recipient.

A failure can be caused by an error condition or timeout.

The figure below shows that the call leg between the originator and the recipient remains active. Thus, if the Invite to the final-recipient fails (408 Request Timeout), the recipient notifies the originator of the failure with a Notify message. The originator sends a re-Invite and returns to the original call with the recipient.

Figure 27: Failed Blind Transfer--Originator Returns to Original Call with Recipient



Attended Transfer

In attended transfers, the Replaces header is inserted by the initiator of the Refer request as an overloaded header in the Refer-To and is copied into the triggered Invite request sent to the final-recipient. The header has no affect on the recipient, but is interpreted by the final-recipient as a way to distinguish between blind transfer and attended transfer. The attended transfer process is described in the table below.

Table 21: Attended Transfer Process

Process	Description or Detail
1. Originator sets up a call with the recipient.	After the call is set up, originator places recipient on hold.
1. Originator establishes a call to the final-recipient.	--
1. Originator sends recipient a Refer request with an overloaded Replaces header in the Refer-To header.	--
1. Upon receipt of the Refer request, recipient sends a triggered Invite request to the final-recipient.	The Invite request received by final-recipient includes the Replaces header, identifying the call leg between the originator and final-recipient.

Process	Description or Detail
1. Recipient returns a SIP 202 (Accepted) response to the originator.	The SIP 202 (Accepted) acknowledges that the Invite has been sent.
1. Final-recipient establishes a direct signaling relationship with recipient.	Receipt of the Replaces header is what indicates that the initial call leg is to be shut down and replaced by the incoming Invite request.
1. Recipient notifies originator of the outcome of the Refer transaction.	Recipient notifies the originator if the final-recipient was successfully or unsuccessfully contacted.
1. Recipient terminates the session with originator by sending a Bye request.	--

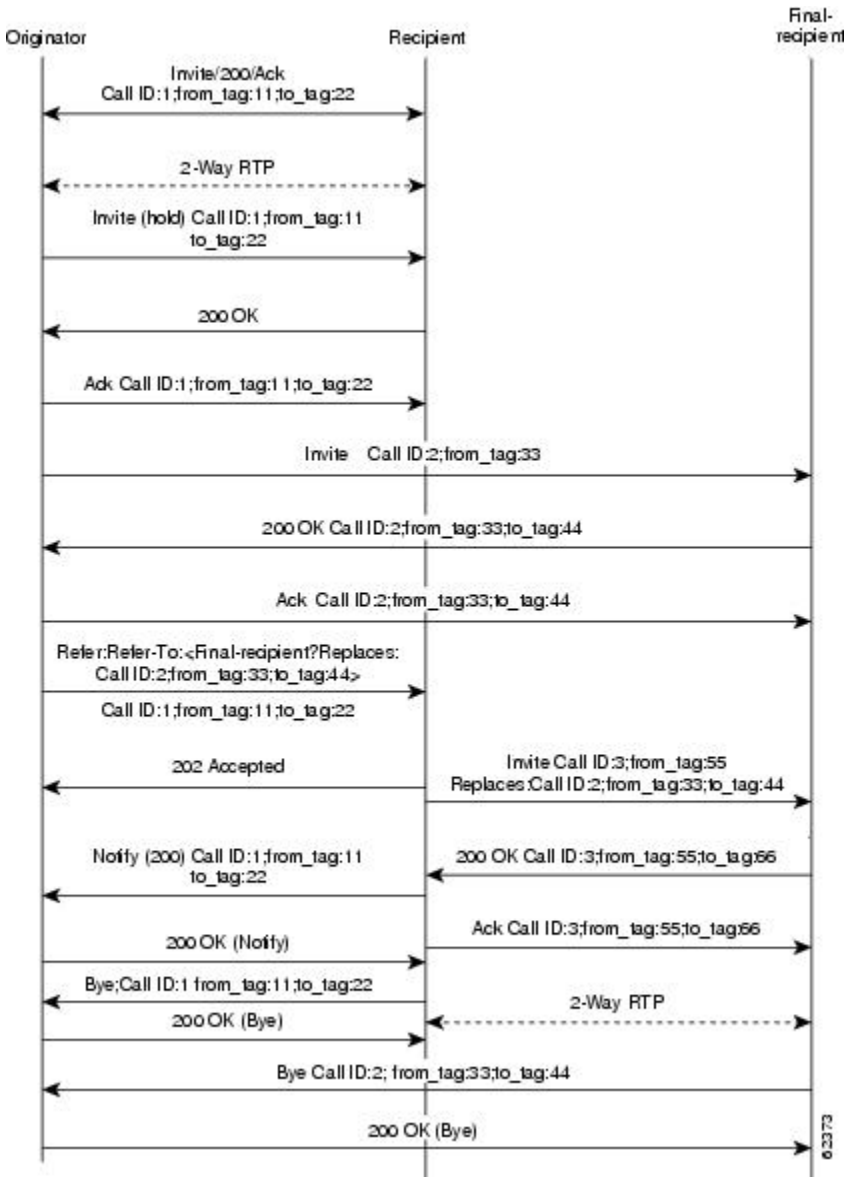
Replaces Header

The Replaces header is required in attended transfers. It indicates to the final-recipient that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming Invite request. The final-recipient sends a Bye request to the originator to terminate its session.

If the information provided by the Replaces header does not match an existing call leg, or if the information provided by the Replaces header matches a call leg but the call leg is not active (a Connect, 200 OK to the Invite request has not been sent by the final-recipient), the triggered Invite does not replace the initial call leg and the triggered Invite request is processed normally.

Any failure resulting from the triggered Invite request from the recipient to final-recipient does not destroy the call between the originator and the final-recipient. In these scenarios, all calls that are active (originator to recipient and originator to final-recipient) remain active after the failed attended transfer attempt. The figure below shows a call flow for a successful attended transfer.

Figure 28: Successful Attended Transfer



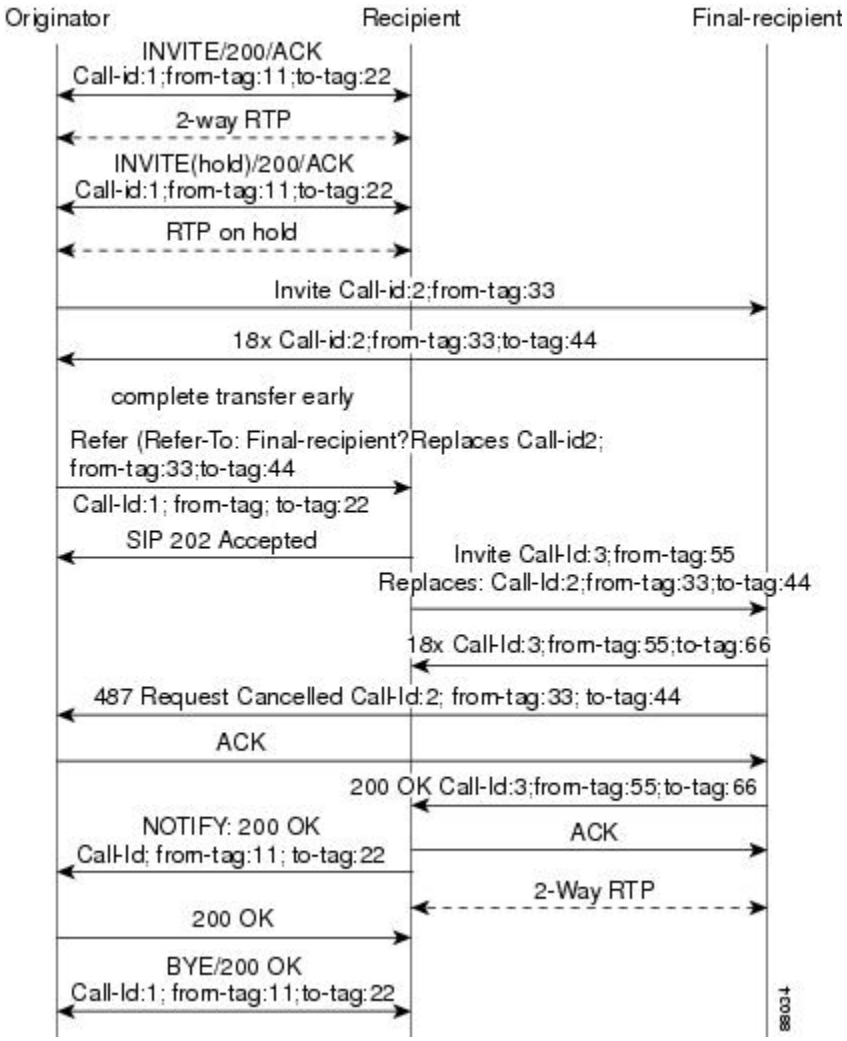
Attended Transfer with Early Completion

Attended transfers allow the originator to have a call established between both the recipient and the final-recipient. With attended transfer with early completion, the call between the originator and the final-recipient does not have to be active, or in the talking state, before the originator can transfer it to the recipient. The originator establishes a call with the recipient and only needs to be in the process of setting up a call with the final-recipient. The final-recipient may be ringing, but has not answered the call from the originator when it receives a re-Invite to replace the call with the originator and the recipient. The figure below shows the process of attended transfer with early completion, and the detailed actions involved are described in the table below.

Table 22: Attended Transfer with Early Completion Process

Process	Description or Detail
1. Originator sets up a call with recipient.	After the call is set up, originator places recipient on hold.
1. Originator contacts final-recipient.	--
1. When originator gets an indication that final-recipient is ringing, it sends recipient a Refer request with an overloaded Replaces header in the Refer-to header.	The Replaces header is required in attended transfers and distinguishes between blind transfer and attended transfers.
1. Recipient returns a SIP 202 (Accepted) response to originator.	The SIP 202 (Accepted) acknowledges that the Invite has been sent.
1. Upon receipt of the Refer request, recipient sends a triggered Invite request to final-recipient.	The Invite request received by final-recipient includes the Replaces header, which indicates that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming Invite request.
1. Final-recipient establishes a direct signaling relationship with recipient.	Final-recipient tries to match the Call-ID header and the To or From tags in the Replaces header of the incoming Invite with an active call leg in its call control block. If a matching active call leg is found, final-recipient replies with exactly the same status as the found call leg. However it then terminates the found call leg with a 487 Request Cancelled response.
Note If early transfer is attempted and the call involves quality of service (QoS) or Resource Reservation Protocol (RSVP), the triggered Invite from the recipient with the Replaces header is not processed and the transfer fails. The session between originator and final-recipient remains unchanged.	
1. Recipient notifies originator of the outcome of the Refer transaction--that is, whether final-recipient was successfully or unsuccessfully contacted.	--
1. Recipient or originator terminates the session by sending a Bye request.	--

Figure 29: Attended Transfer with Early Completion



VSA for Call Transfer

You can use a vendor-specific attribute (VSA) for SIP call transfer.

Referred-By Header

For consistency with existing billing models, the Referred-By and Requested-By headers are populated in call history tables as a VSA. Cisco VSAs are used for VoIP call authorization. The new VSA tag **supp-svc-xfer-by**helps to associate the call-legs for Call Detail Records (CDR) generation. The call-legs could be originator to recipient or recipient to final-recipient.

The new VSA tag **supp-svc-xfer-by** contains the user@host portion of the SIP URL of the Referred-By header for transfers performed with the Refer method. For transfers performed with the Bye/Also method, the tag contains the user@host portion of the SIP URL of the Requested-By header. For each call on the gateway, there are two RADIUS records that are generated: start and stop. The **supp-svc-xfer-by**VSA is only generated for stop records and is only generated on the recipient gateway--the gateway receiving the Refer or Bye/Also message.

The VSA is generated when a gateway that acts as a recipient receives a Refer or Bye/Also message with the Referred-By or Requested-By headers. There are usually two pairs of start and stop records. There is a start and stop record between the recipient and the originator and also between the recipient to final-recipient. In the latter case, the VSA is generated between the recipient to final-recipient only.

Business Group Field

A new business group VSA field has also been added that assists service providers with billing. The field allows service providers to add a proprietary header to call records. The VSA tag for business group ID is **cust-biz-grp-id** and is only generated for stop records. It is generated when the gateway receives an initial Invite with a vendor dial-plan header to be used in call records. In cases when the gateway acts as a recipient, the VSA is populated in the stop records between the recipient and originator and the recipient final-recipient.

Note For more information about VSAs and CDRs, see the *CDR Accounting for Cisco IOS Voice Gateways* guide.

SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications

SIP Call Transfer and Call Forwarding with a Tcl IVR Script

When using a Tcl IVR 2.0 application, you can implement SIP support of blind or attended call-transfer and call-forwarding requests from a Cisco IOS gateway. A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. An attended transfer is one that is consultative--one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller. Blind transfers are often preferred by automated devices that do not have the capability to make consultative calls.

Before implementing blind transfer and call forwarding, you must write a custom Tcl IVR 2.0 script that implements call transfer and call forwarding. The script is responsible for receiving the hookflash event, providing dial tone, matching against the dial plan, initiating call transfer, and reestablishing the original call if the transfer attempt fails.

Note For information on writing a Tcl IVR script, see the *Tcl IVR API Version 2.0 Programmer's Guide*.

When the Tcl IVR script runs on the Cisco gateway, it can respond to requests to initiate blind call transfer (transfer without consultation) on a SIP call leg. SIP call forwarding on ephones (IP phones that are not configured on the gateway) is also supported.

Note SIP Call Transfer and Call Forwarding is compliant with Voice Extensible Markup Language (VXML). VXML scripts can also be used to implement call transfer and call forwarding.

Release Link Trunking on SIP Gateways

RLT functionality has been added to Cisco IOS SIP gateways. With RLT functionality, SIP call transfer can now be triggered by CAS trunk signaling, which the custom Tcl IVR application can monitor. After a SIP call transfer has transpired and the CAS interface is no longer required, the CAS interface can be released.

The RLT functionality can be used to initiate blind transfers on SIP gateways. Blind call transfer uses the Refer method. A full description of blind transfer and the refer Method can be found in "Call Transfer Capabilities Using the Refer Method" documentation.

RLT and SIP Call Transfers

Call transfer can be triggered by CAS trunk signaling and then captured by the custom Tcl IVR script on a gateway. The process begins with the originator (the SIP user agent that initiates the transfer or Refer request) responding with a dial tone once the originator receives the signal or hookflash from the PSTN call leg. The originator then prepares to receive dual-tone multifrequency (DTMF) digits that identify the final- recipient (the user agent introduced into a call with the recipient).

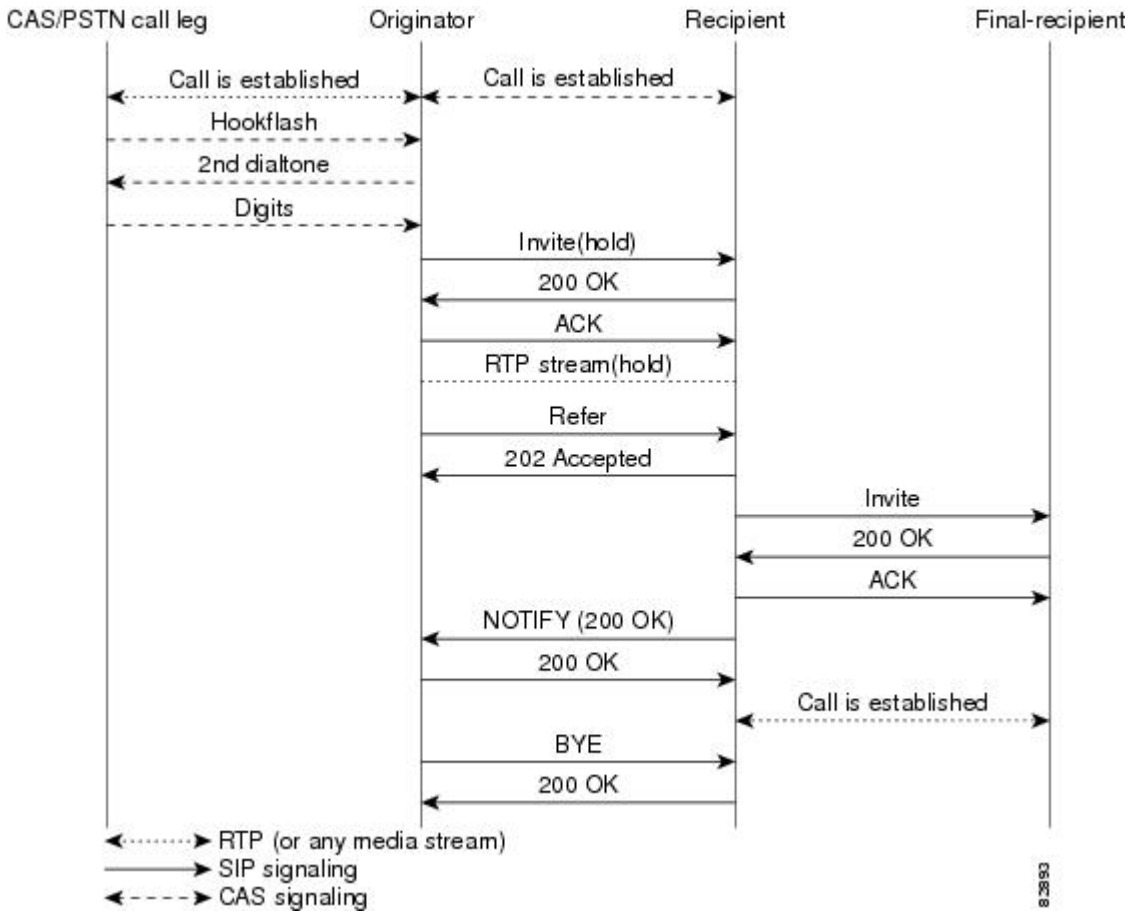
Once the first DTMF digit is received, the dial tone is discontinued. DTMF-digit collection is not completed until a 4-second interdigit timeout occurs, or an on-hook is received on that specific CAS time slot. Call transfer starts when DTMF-digit collection is successful. If digit collection fails, for example if not enough DTMF digits or invalid digits are collected, the initial call is reestablished.

Once the DTMF digits are successfully collected, the custom Tcl IVR script can initiate call transfer. SIP messaging begins when the transfer is initiated with the Refer method. The originator sends an Invite to the recipient (the user agent that receives the Refer request and is transferred to the final-recipient) to hold the call and request that the recipient not return Real-Time Transport Protocol (RTP) packets to the originator. The originator then sends a SIP Refer request to the recipient to start the transfer process. When the recipient receives the request, the recipient returns a 202 *Accepted* acknowledgement to the originator. The Tcl IVR script run by the originator can then release the CAS trunk and close the primary call (see the figure below).

If the recipient does not support the Refer method, a 501 *Not implemented* message is returned. However, for backward compatibility purposes, the call transfer is automatically continued with the Bye/Also method. The originator sends a Bye/Also request to the recipient and releases the CAS trunk with the PSTN call leg. The primary call between the originator and the recipient is closed when a 200 OK response is received.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down.

Figure 30: Call Transfer Using the Refer Method



SIP and TEL URLs in Call Transfers

When the SIP call-transfer originator collects DTMF digits from the CAS trunk, it attempts to find a dial peer. If a dial peer is found, the session target in the dial peer is used to formulate a Session Initiation Protocol Uniform Resource Locator (SIP URL). This URL can be used with both the Refer method and the Bye/Also method. A SIP URL is in the following form:

```
sip:JohnSmith@example.com
```

If a valid dial peer is not found, a Telephone Uniform Resource Locator (TEL URL) is formulated in the Refer-To header. A TEL URL is in the following form:

```
tel:+11231234567
```

The choice of which URL to use is critical when correctly routing SIP calls. For example, the originating gateway can send out a Bye with an Also header, but the Also header can carry only a SIP URL. The Also header cannot carry a TEL URL. That is, if the gateway decides to send a Bye/Also but cannot find a matched dial peer, the gateway reports an error on the transfer gateway and sends a Bye without the Also header.

If the recipient of a SIP call transfer is a SIP phone, the phone must have the capability to interpret either the Refer method or the Bye/Also method for the call transfer to work. If the recipient is a Cisco IOS gateway, there needs to be a matching dial peer for the Refer-To *user* . *User,* looking at the previous example, can be

either *JohnSmith* or *11231234567* . The dial peer also needs to have an application session defined, where session can be the name of a Tcl IVR application. If there's no match, a 4*xx* error is sent back and no transfer occurs. If there's a POTS dial peer match, a call is made to that POTS phone. Before the 12.2(15)T release, if there's a VoIP match, the Refer-To URL is used to initiate a SIP call. In release 12.2(15)T and later releases, the application session target in the dial peer is used for the SIP call.

Note For information on the application session target, see the "Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer".

SIP Gateway Initiation of Call Transfers

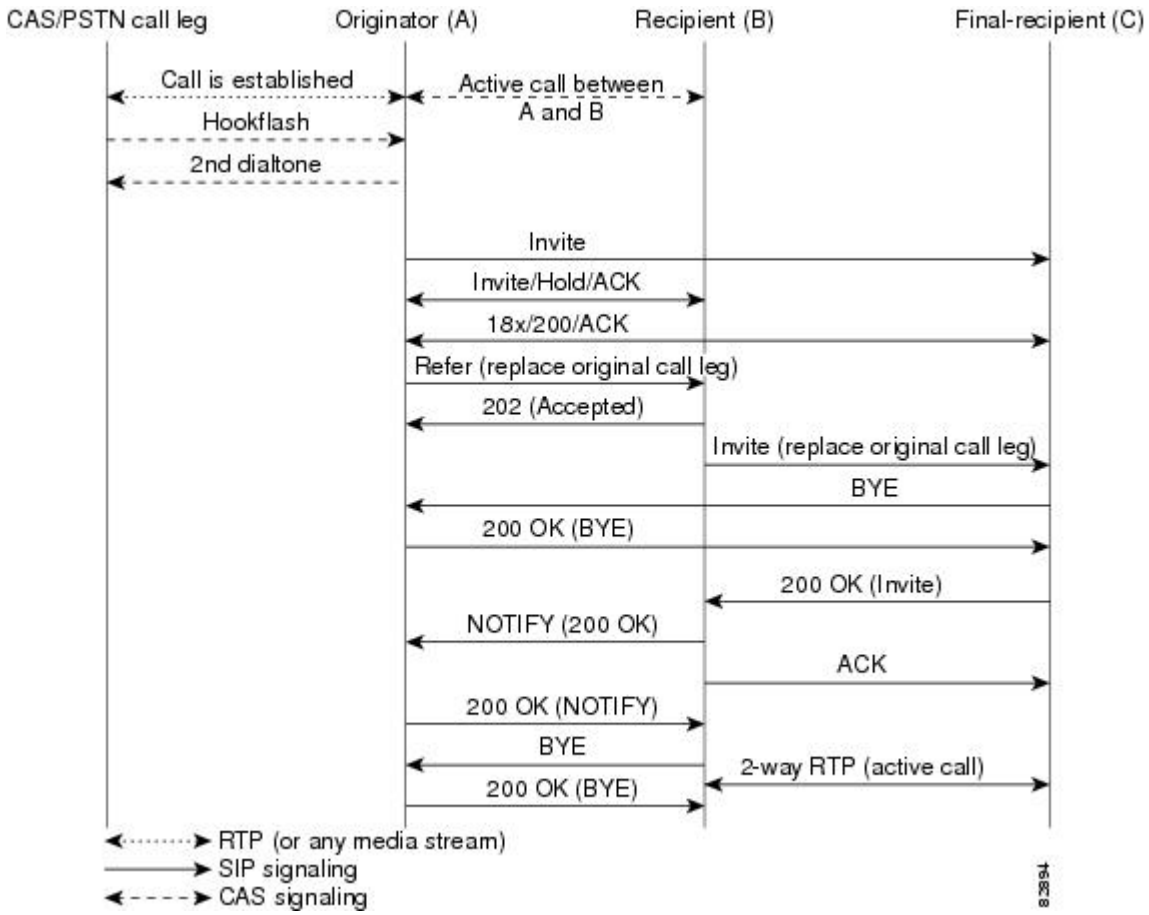
SIP gateways can also initiate, or originate, attended call transfers. The process begins when the originator establishes a call with the recipient. When the user on the PSTN call leg wants to transfer the call, the user uses hookflash to get a second dial tone and then enters the final-recipients number. The Tcl IVR script can then put the original call on hold and set up the call to the final-recipient, making the originator active with the final-recipient. The Refer request is sent out when the user hangs up to transfer the call. The Refer request contains a Replaces header that contains three tags: *SIP CallID* , *from* , and *to* . The tags are passed along in the Invite from the recipient to the final-recipient, giving the final-recipient adequate information to replace the call leg. The host portion of the Refer request is built from the established initial call. The following is an example of a Refer request that contains a Replaces header:

Note IP addresses and hostnames in examples are fictitious.

```
Refer sip:3100801@172.16.190.100:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  172.16.190.99:5060
From: "5550100" <sip:5550100@172.16.190.187>
To: <sip:3100801@172.16.190.187>;tag=A7C2C-1E8C
Date: Sat, 01 Jan 2000 05:15:06 GMT
Call-ID: c2943000-106ae5-1c5f-3428@172.16.197.182
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 946685709
CSeq: 103 Refer
Refer-To:
sip:3100802@10.102.17.217?Replaces=DD713380-339C11CC-80BCF308-92BA812C@172.16.195.77;to-tag=A5438-23E4;from-tag=C9122EDB-2408
Referred-By: <sip:3100802@172.16.190.99>
Content-Length: 0
```

After the NOTIFY is received by the originator, the Tcl IVR script can disconnect the call between the originator and the recipient. The call between the originator and the final-recipient is disconnected by the recipient sending a BYE to the originator. The figure below shows a call flow of a successful call transfer.

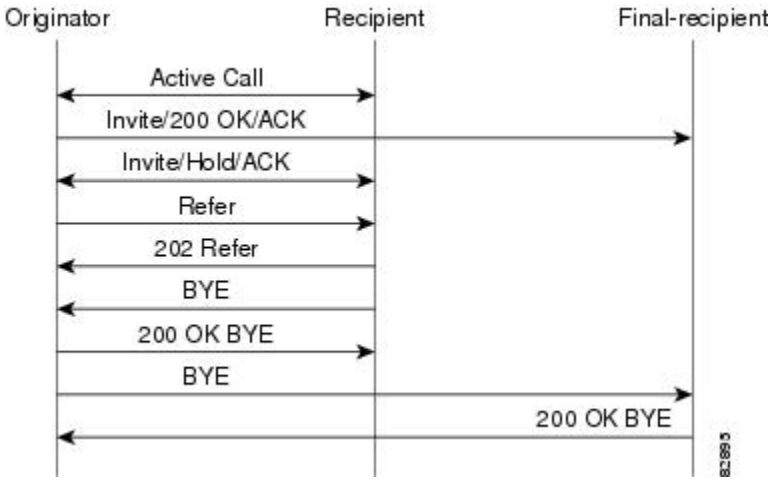
Figure 31: Successful Attended Call Transfer Initiated by the Originator



If the recipient does not support the Refer method, a 501 *Not implemented* message is returned.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down. The figure below shows the recipient hanging up the call before the transfer completes. The item to notice is that the NOTIFY message is never sent.

Figure 32: Unsuccessful Call Transfer--Recipient Hangs Up Before Transfer Completes



SIP Call Forwarding

SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS, FXO, T1, E1, and CAS phones are not supported.

With ephones, there are four different types of SIP call forwarding supported:

- Call Forward Unavailable
- Call Forward No Answer
- Call Forward Busy
- Call Forward Unconditional

In all four of these call forwarding types, a 302 *Moved Temporarily* response is sent to the user agent client. A Diversion header included in the 302 response indicates the type of forward.

The 302 response also includes a Contact header, which is generated by the calling number that is provided by the custom Tcl IVR script. The 302 response also includes the host portion found in the dial peer for that calling number. If the calling number cannot match a VoIP dial-peer or POTS dial-peer number, a 503 *Service Unavailable* message is sent, except in the case of the Call Forward No Answer. With Call Forward No Answer, call forwarding is ignored, the phone rings, and the expires timer clears the call if there is no answer.

Note By default, SIP credentials for forwarded calls on Cisco IOS voice gateways are based on the calling number. To globally enable a gateway to use the redirecting number, instead, use the **authenticate redirecting-number** command. To configure this behavior for a specific dial peer on a gateway, use the **voice-class sip authenticate redirecting-number** command. For detailed information, see these commands in the *Cisco IOS Voice Command Reference*.

Note In Cisco IOS Release 12.2(15)T and later releases, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

How to Configure SIP Call-Transfer Features

Note For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring SIP Call Transfer Using the Refer Method

Configure SIP Call Transfer on a POTS Dial Peer

Note To handle all call-transfer situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer for a POTS dial peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **pots**
4. **application** application-name
5. **destination-pattern** [+]*string*[**T**]
6. **port** *slot* / *port*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**dial-peer voice** *tag* **pots** **Example:** `Router(config)# dial-peer voice 25 pots`	Enters dial-peer configuration mode for the specified POTS dial peer.
Step 4	**application** application-name **Example:** `Router(config-dial-peer)# application session`	Enables a specific application on a dial peer. The argument is as follows: • *application-name* --Name of the predefined application that you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers.
Step 5	**destination-pattern** [+]*string*[**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer: Keywords and arguments are as follows: • + --(Optional) Character indicating an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable length dial string.
Step 6	**port** *slot* / *port* **Example:** `Router(config-dial-peer)# port 1/1`	Associates a dial peer with a voice slot number and a specific local voice port through which incoming VoIP calls are received. **Note** To find the correct port argument for your router, see the *Cisco IOS Voice Command Reference*.
Step 7	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure SIP Call Transfer on a VoIP Dial Peer

Note To handle all call-transfer situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer for a VoIP dial peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **application** *application-name*
5. **destination-pattern** [+]*string* [**T**]
6. **session target ipv4** :*destination-address*
7. **session protocol sipv2**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 4	**application** *application-name* **Example:** `Router(config-dial-peer)# application session`	Enables a specific application on a dial peer. The argument is as follows: • *application-name* --Name of the predefined application that you wish to enable on the dial peer. For SIP, the default Tcl application (from the Cisco IOS image) is session and can be applied to both VoIP and POTS dial peers.

	Command or Action	Purpose
Step 5	**destination-pattern** [+]*string* [**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows: • **+** --(Optional) Character that indicates an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 6	**session target ipv4** :*destination-address* **Example:** `Router(config-dial-peer)# session target ipv4:10.10.1.3`	**S** pecifies a network-specific address for a dial peer. Keyword and argument are as follows: • **ipv4** :*destination address* --IP address of the dial peer, in this format: *xxx.xxx.xxx.xxx*
Step 7	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use IETF SIP.
Step 8	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure the SIP Call-Transfer Session Target

To configure the SIP call-transfer session target, perform the following steps.

This task configures a SIP server as a session target. Although it is not required, configuring a SIP server as a session target is useful if there is a Cisco SIP proxy server (CSPS) present in the network. With a CSPS, you can configure the SIP server option and have the interested dial peers use the CSPS by default.

To determine the call-transfer destination on the originator, check if there is a matching dial peer:

- If yes, check the session target for the dial peer. If the session target is a SIP server, configure the SIP server as described in the task below. If the session target is not a SIP server, the session target configured in the VoIP dial peer is used.
- If no, a TEL URL is sent.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **sip-ua**
4. **sip-server dns** : *host-name*
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **destination-pattern** [+]*string* [**T**]
8. **session target sip-server**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**sip-server dns** : *host-name* **Example:** `Router(config-sip-ua)# sip-server dns:example.sip.com`	Sets the global SIP server interface to a Domain Name System (DNS) hostname. If you do not specify a hostname, the default DNS defined by the **ip name-server** command is used.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 7	**destination-pattern** [+]*string* [**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows: • + --(Optional) Character that indicates an E.164 standard number.

	Command or Action	Purpose
		• *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 8	**session target sip-server** **Example:** `Router(config-dial-peer)# session target sip-server`	**Instructs the dial-peer session target to use the global SIP server** . Doing so saves you from having to repeatedly enter the SIP server interface address for each dial peer.
Step 9	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure SIP Refer and Notify Message Settings

Note The Refer request is initiated by the originating gateway and signals the start of call transfer. Once the outcome of the SIP Refer transaction is known, the recipient of the Refer request notifies the originating gateway of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. Notification is accomplished using the Notify method.

Before you begin

- Configure dial peers for correct functioning of the Refer method.

Note For dial-peer configuration steps, see "Configure SIP Call Transfer on a POTS Dial Peer".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers notify** *milliseconds*
5. **retry notify** *number*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**timers notify** *milliseconds* **Example:** `Router(config-sip-ua)# timers notify 500`	Sets the amount time that the user agent waits before retransmitting the Notify message. The argument is as follows: • *milliseconds* --Time, in ms. Range: 100 to 1000. Default: 500.
Step 5	**retry notify** *number* **Example:** `Router(config-sip-ua)# retry notify 10`	Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows: • *number* --Number of notify message retries. Range: 1 to 10.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications

Load the Tcl IVR Application on the Gateway

Before you begin

- Before you implement SIP support of blind or attended call-transfer and call-forwarding requests from a Cisco IOS gateway, you must load a custom Tcl IVR 2.0 or VXML script on the gateway. Write a Tcl IVR 2.0 script that implements Cisco IOS call-transfer and call-forwarding services. The Tcl IVR script

is responsible for receiving the hookflash event, providing dial tone, matching against the dial plan, initiating the call transfer, and reestablishing the original call if the transfer attempt fails.

Note For information on writing a Tcl IVR script, see "Tcl IVR API Version 2.0 Programmer's Guide".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call application voice** *application-name location*
4. **call application voice** *application-name* **language** *number language*
5. **call application voice** *application-name* **set-location** *language category location*
6. **exit**
7. **all application voice load** *application-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call application voice** *application-name location* **Example:** `Router(config)# call application voice transfer_app flash:app_h450_transfer.tcl`	Loads the Tcl IVR script and specifies its application name. Arguments are as follows: • *application-name* --Name used to reference the call application. This is a user-defined name and does not have to match the document name. • *location* --Location of the Tcl IVR file in URL format. For example, flash memory (flash:filename), TFTP (tftp://../filename) or HTTP server paths (http://../filename) are valid locations.
Step 4	**call application voice** *application-name* **language** *number language* **Example:** `Router(config)# call application voice transfer_app language 1 en`	(Optional) Sets the language for dynamic prompts used by the application. Arguments are as follows: • *application-name* --Name of the Tcl IVR application to which the language parameters pass. • *number* --Number that identifies the language used by the audio files for the IVR application.

	Command or Action	Purpose
		• *language* --Language of the associated audio file. Valid values are as follows: • **en**--English • **sp**--Spanish • **ch**--Mandarin • **aa**--All
Step 5	**call application voice** *application-name* **set-location** *language category location* **Example:** `Router(config)# call application voice transfer_app set-location en 0 flash:/prompts`	(Optional) Defines the location and category of the audio files that are used by the application for dynamic prompts. Arguments are as follows: • *application-name* --Name of the Tcl IVR application. • *language* --Language of the associated audio file. Valid values are as above. • *category* --Category group (0 to 4) for the audio files from this location. For example, audio files for the days and months could be category 1, audio files for units of currency could be category 2, and audio files for units of time (seconds, minutes, and hours) could be category 3. Range is from 0 to 4. The value 0 means all categories. • *location* --URL of the directory that contains the language audio files used by the application, without filenames. For example, flash memory (flash) or a directory on a server (TFTP, HTTP, or RTSP) are valid locations.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits the current mode.
Step 7	**all application voice load** *application-name* **Example:** `Router# call application voice load transfer.app`	(Optional) Reloads the Tcl script after it has been modified. The argument is as follows: • *application-name* --Name of the Tcl IVR application to reload.

Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer

Note To handle all call-transfer and call-forwarding situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer and call forwarding for a POTS dial peer.

- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide* .

Note To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:http://www.cisco.com/web/psa/products/index.html.

Note In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **pots**
4. **application** *application-name*
5. **destination-pattern** [+]*string* [**T**]
6. **port** *slot* / *port*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **pots** **Example:** `Router(config)# dial-peer voice 25 pots`	Enters dial-peer configuration mode and for the specified POTS dial peer.
Step 4	**application** *application-name* **Example:** `Router(config-dial-peer)# application transfer_app`	Enables a specific application on a dial peer. The argument is as follows:

	Command or Action	Purpose
Step 5	**destination-pattern** [+]*string* [**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows: • + --(Optional) Character that indicates an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 6	**port** *slot* / *port* **Example:** `Router(config-dial-peer)# port 1/1`	Associates a dial peer with a voice slot number and a specific local voice port through which incoming VoIP calls are received. **Note** To find the correct port argument for your router, see the *Cisco IOS Voice Command Reference*.
Step 7	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure SIP Call Transfer and Call Forwarding on a VoIP Dial Peer

Note To handle all call-transfer and call-forwarding situations, configure both POTS and VoIP dial peers. This task configures SIP call transfer and call forwarding for a VoIP dial peer.

- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide*.

Note To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:http://www.cisco.com/web/psa/products/index.html .

Note

- RLT on CAS or analog (FXS) ports is necessary for initiating IP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **application** *application-name*
5. **destination-pattern** [+]*string* [**T**]
6. **session target ipv4:** *destination-address*
7. **session protocol sipv2**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 4	**application** *application-name* **Example:** `Router(config-dial-peer)# application transfer_app`	Enables a specific application on a dial peer. The argument is as follows:
Step 5	**destination-pattern** [+]*string* [**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:

	Command or Action	Purpose
		• + --(Optional) Character that indicates an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 6	**session target ipv4:** *destination-address* **Example:** `Router(config-dial-peer)# session target ipv4:10.10.1.3`	**S** pecifies a network-specific address for a dial peer. The argument is as follows: • *destination address* --IP address of the dial peer, in this format: *xxx.xxx.xxx.xxx*
Step 7	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use IETF SIP. The keyword is as follows: • **sipv2** --Causes the VoIP dial peer to use IETF SIP. Use this keyword with the SIP option.
Step 8	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure the SIP Call-Transfer and Call-Forwarding Session Target

Note To configure a SIP server as a session target, follow this task. Although configuring a SIP server as a session target is not required, it is useful if there is a Cisco SIP proxy server (CSPS) present in the network. With a CSPS, you can configure the SIP server option and have the interested dial peers use the CSPS by default.

- To determine the call-transfer destination on the originator, check if there is a matching dial peer:
  - If yes, check the session target for the dial peer. If the session target is a SIP server, configure the SIP server as described in the task below. If the session target is not a SIP server, the session target configured in the VoIP dial peer is used.
  - If no, a TEL URL is sent.
- To configure SIP call transfer and forwarding on a Cisco IOS gateway by using the CAS trunk, see the *Cisco IOS Dial Technologies Configuration Guide* .

Note To locate a release-specific configuration guide for your Cisco IOS software release, select the **Cisco IOS and NX-OS Software** category at the following Product Support page and navigate accordingly:http://www.cisco.com/web/psa/products/index.html .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **sip-server dns:** *host-name*
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **destination-pattern** [+]*string*[**T**]
8. **session target sip-server**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**sip-server dns:** *host-name* **Example:** `Router(config-sip-ua)# sip-server dns:example.sip.com`	Sets the global SIP server interface to a DNS hostname. The argument is as follows: • *host-name* --DNS hostname. If you do not specify a hostname, the default DNS defined by the **ip name-server** command is used.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

	Command or Action	Purpose
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 7	**destination-pattern** [+]*string*[**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows: • + --(Optional) Character that indicates an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character. • **T** --(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 8	**session target sip-server** **Example:** `Router(config-dial-peer)# session target sip-server`	**Instruct the dial-peer session target to use the global SIP server** . Doing so saves you from having to repeatedly enter the SIP server interface address for each dial peer.
Step 9	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure SIP Refer and Notify Message Settings

To configure SIP Refer and Notify message settings, perform the following steps.

Note The Refer request is initiated by the originating gateway and signals the start of call transfer. Once the outcome of the SIP Refer transaction is known, the recipient of the Refer request notifies the originating gateway of the outcome of the Refer transaction--whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify method.

Before you begin

- Custom scripting is necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.
- Configure the dial peers for correct functioning of the Refer method.

Note Dial-peer configuration steps are in "Configure SIP Call Transfer and Call Forwarding on a POTS Dial Peer".

Note

- Only RLT on CAS or analog (FXS) ports is supported with SIP call transfers.
- The Cisco AS5xxx platforms do not support hookflash detection for T1 CAS.
- SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. FXS and CAS phones are not supported.
- In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding. The standard configurations listed in this document work only when an ephone is the recipient or final-recipient.

Note Custom scripting is necessary for ephones to initiate call forwarding. The standard configurati this document work only when an ephone is the recipient or final-recipient.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers refer** *milliseconds*
5. **retry refer** *number*
6. **timers notify** *milliseconds*
7. **retry notify** *number*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:**	Enters SIP user-agent configuration mode.

	Command or Action	Purpose
	`Router(config)# sip-ua`	
Step 4	**timers refer** *milliseconds* **Example:** `Router(config-sip-ua)# timers refer 500`	Sets the amount time that the user agent waits before retransmitting the Refer request. The argument is as follows: • *milliseconds* --Time, in ms. Range: 100 to 1000. Default: 500.
Step 5	**retry refer** *number* **Example:** `Router(config-sip-ua)# retry refer 10`	Sets the number of times that the Refer request is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows: • *number* --Number of Notify message retries. Range: 1 to 10. Default: 10.
Step 6	**timers notify** *milliseconds* **Example:** `Router(config-sip-ua)# timers notify 500`	Sets the amount time that the user agent waits before retransmitting the Notify message. The argument is as follows: • *milliseconds* --Time, in ms. Range: 100 to 1000. Default: 500.
Step 7	**retry notify** *number* **Example:** `Router(config-sip-ua)# retry notify 10`	Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or refer request. The argument is as follows: • *number* --Number of notify message retries. Range: 1 to 10.
Step 8	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Verifying SIP Call Transfer

To verify SIP configurations, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show dial-peer voice**
2. **show ephone**
3. **show ephone-dn**
4. **show running-configuration**
5. **show sip-ua retry**
6. **show sip-ua statistics**
7. **show sip-ua timers**

8. **show telephony-service ephone-dn**
9. **show telephony-service voice-port**
10. **show voice port**

DETAILED STEPS

Step 1 **show dial-peer voice**

Use this command to display configuration information about voice dial peers. Use with the **summary** keyword to display a summary of all voice dial peers.

Step 2 **show ephone**

Use this command to display IP-phone output. Use with the **summary** keyword to display a summary of all IP phones.

Step 3 **show ephone-dn**

Use this command to display the IP-phone destination number. Use with the **summary** keyword to display a summary of all IP-phone destination numbers.

Step 4 **show running-configuration**

Use this command to verify your configuration.

Step 5 **show sip-ua retry**

Use this command to display SIP retry statistics including Notify responses.

Example:

```
Router# show sip-ua retry
SIP UA Retry Values
invite retry count = 6 response retry count = 1
bye retry count = 1 cancel retry count = 1
prack retry count = 10 comet retry count = 10
reliable 1xx count = 6 notify retry count = 10
```

Step 6 **show sip-ua statistics**

Use this command to display response, traffic, and retry statistics for the SIP user agent.

The following applies to the example below.

Field	Meaning
OkNotify1/0	Successful response to the Notify request.
202Accepted 0/1	Successful response to the Refer request.
Notify 0/1	Status.
Refer 1/0	Status.
Notify 0/1	No Notify requests were received from the gateway. One request was sent.
Refer 1/0	One request was received. No requests were sent.
Notify 0 under Retry Statistics	The Notify request was not retransmitted.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OKSubscribe 0/0, OkNotify 0/0,
      202Accepted 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, UseProxy 0,
      AlternateService 0
    Client Error:
      BadRequest 0/0, 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,
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, RequestCancel 0/0
      NotAcceptableMedia 0/0, BadEvent 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
    Miscellaneous counters:
      RedirectResponseMappedToClientError 1,
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0
Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0, Notify 0
SDP application statistics:
 Parses: 0,  Builds 0
 Invalid token order: 0,  Invalid param: 0
 Not SDP desc: 0,  No resource: 0
```

Tip To reset counters for the **sip-ua statistics**command, use the **clear sip-ua statistics** command.

Step 7 **show sip-ua timers**

Use this command to display the current settings for SIP user-agent timers, including Notify responses.

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 150000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
```

Step 8 **show telephony-service ephone-dn**

Use this command to display the Cisco-IP-phone destination number of the Cisco IOS telephony-service router.

Step 9 **show telephony-service voice-port**

Use this command to display the virtual voice-port configuration.

Step 10 **show voice port**

Use this command to display configuration information about a specific voice port.

Troubleshooting Tips

For general troubleshooting tips and a list of important **debug** commands, see the "General Troubleshooting Tips" section of the "Basic SIP Configuration" document.

Configuration Examples for SIP Call-Transfer Features

SIP Call Transfer Using the Refer Method Examples

Note Note that the **application session** command is set on all involved gateway dial peers. You must set the correct Cisco IOS session for call transfer.

```
Router# show running-config
Building configuration...
Current configuration : 4192 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
voice-card 2
!
ip subnet-zero
!
controller T1 2/0
framing esf
```

```
linecode b8zs
ds0-group 0 timeslots 1-24 type e&m-wink-start
!
interface FastEthernet3/0
ip address 172.18.200.36 255.255.255.0
speed 10
half-duplex
no shut
ip rsvp bandwidth 7500 7500
!
voice-port 2/0:0
timing hookflash-in 1500
!
dial-peer voice 3660110 voip
application session
incoming called-number 3660110
destination-pattern 3660110
session protocol sipv2
session target ipv4:172.18.200.24
codec g711ulaw
!
dial-peer voice 3640110 pots
application session
destination-pattern 3640110
direct-inward-dial
port 2/0:0
!
sip-ua
retry bye 1
retry refer 3
timers notify 400
timers refer 567
no oli
sip-server ipv4:172.18.200.21
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications Examples

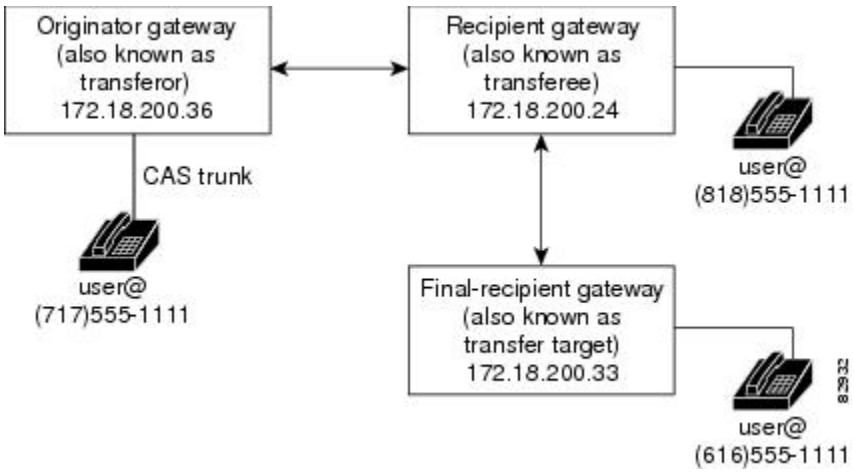
This section provides an end-to-end call-transfer configuration example.

Note IP addresses and hostnames in examples are fictitious.

Blind Call Transfer

The figure below shows the relationship of the gateways in the blind call transfer.

Figure 33: Blind Call Transfer



The following scenario is an example of a blind call transfer.

1. User at (818) 555-0111 calls user at (717) 555-0111, and they are in a conversation.
2. User at (717) 555-0111 decides to transfer user at (818) 555-0111 to user at (616) 555-0111.

Transfer takes place by the user at (717) 555-0111 going on-hook over the CAS trunk and dialing (616) 555-0111.

1. Call transfer is initiated from the originating gateway to the recipient gateway, and the originator releases the CAS trunk to (717) 555-0111.
2. Recipient gateway releases the call leg to the originator and initiates a new call to the final-recipient--user at (616) 555-0111.
3. Call transfer is complete, and user at (818) 555-0111 and user at (616) 555-0111 are in a conversation.

Originating Gateway

The following example shows a configuration of the originating gateway.

```
Router# show running-config
Building configuration...
Current configuration : 4192 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
voice-card 2
!
ip subnet-zero
!
controller T1 2/0
```

```
framing esf
linecode b8zs
ds0-group 0 timeslots 1-24 type e&m-wink-start
!
interface FastEthernet3/0
ip address 172.18.200.36 255.255.255.0
speed 10
half-duplex
no shut
ip rsvp bandwidth 7500 7500
!
voice-port 2/0:0
timing hookflash-in 1500
!
call application voice sample_RLT tftp://sample_RLT.tcl
call application voice sample_RLT uid-len 4
call application voice sample_RLT language 1 en
call application voice sample_RLT set-location en 0 tftp://prompts/en/
!
dial-peer voice 2 voip
application sample_rlt
destination-pattern 8183821111
session protocol sipv2
session target ipv4:172.18.200.24
codec g711ulaw
!
dial-peer voice 3 pots
destination-pattern 7173721111
direct-inward-dial
port 2/0:0
prefix 7173721111
!
dial-peer voice 3621111 voip
application sample_rlt
destination-pattern 6163621111
session protocol sipv2
session target sip-server
codec g711ulaw
!
sip-ua
retry bye 1
retry refer 3
timers notify 400
timers refer 567
no oli
sip-server ipv4:172.18.200.21
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Recipient Gateway

The following example shows a configuration of the recipient gateway.

```
Router# show running-config
Building configuration...
Current configuration : 2791 bytes
!
version 12.2
```

```
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
interface FastEthernet2/0
ip address 172.18.200.24 255.255.255.0
duplex auto
no shut
speed 10
ip rsvp bandwidth 7500 7500
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 8183821111
port 1/1/1
!
dial-peer voice 3 voip
application session
destination-pattern 7173721111
session protocol sipv2
session target ipv4:172.18.200.36
codec g711ulaw
!
dial-peer voice 4 voip
application session
destination-pattern 6163621111
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
sip-ua
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Final-Recipient

The following example shows a configuration of the final-recipient gateway.

```
Router# show running-config
!
version 12.2
no parser cache
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
no logging buffered
```

```
!
clock timezone GMT 0
aaa new-model
!
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
!
ip domain name example.com
ip dhcp smart-relay
!
!
voice class codec 1
codec preference 2 g711alaw
codec preference 3 g711ulaw
codec preference 5 g726r16
codec preference 6 g726r24
codec preference 7 g726r32
codec preference 8 g723ar53
codec preference 9 g723ar63
codec preference 10 g729r8
!
interface Ethernet0/0
ip address 172.18.200.33 255.255.255.0
no shut
half-duplex
ip rsvp bandwidth 7500 7500
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dial-peer voice 1 pots
application session
destination-pattern 6163621111
port 1/1/1
!
ip classless
no ip http server
ip pim bidir-enable
!
gateway
!
sip-ua
!
rtr responder
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
password password1
line vty 5 15
```

```
!
end
```

Additional References

General SIP References

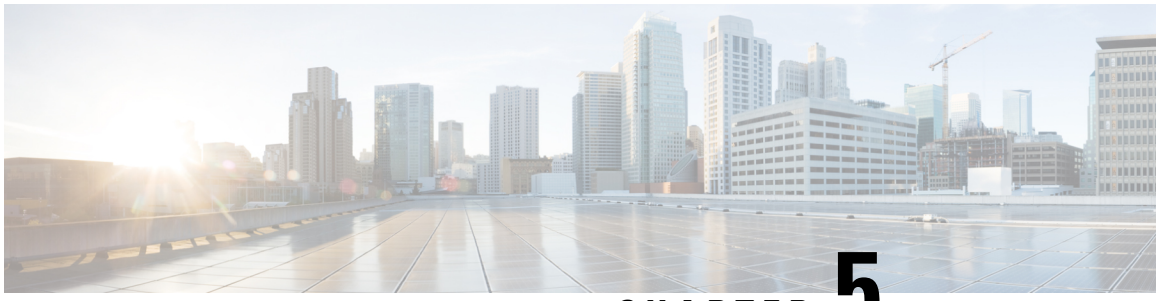
- “Overview of SIP” --Describes underlying SIP technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance.

References Mentioned in This Chapter (listed alphabetically)

- "Call Transfer Capabilities Using the Refer Method".
- *Cisco IOS Dial Technologies Configuration Guide*.

Note To locate a release-specific configuration guide for your Cisco IOS software release, select the Cisco IOS and NX-OS Software category at the following Product Support page and navigate accordingly:http://www.cisco.com/web/psa/products/index.html.

- *Cisco IOS Voice Command Reference*.
- "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms".
- " CDR Accounting for Cisco IOS Voice Gateways" guide.
- "Tcl IVR API Version 2.0 Programmer’s Guide".



CHAPTER 5

Configuring SIP Message Timer and Response Features

This chapter describes how to configure Session Initiation Protocol (SIP) message components, session timers, and responses. It describes the following features:

History for the Internal Cause Code Consistency Between SIP and H.323 Feature

Release	Modification
12.2(11)T	These features were introduced.

History for the SIP - Configurable PSTN Cause Code Mapping Feature

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB2	This feature was implemented on an additional platform.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

History for the SIP Accept-Language Header Support Feature

Release	Modification
12.3(1)	The SIP Accept-Language Header Support feature was introduced.

History for the SIP Enhanced 180 Provisional Response Handling Feature

Release	Modification
12.2(13)T	The features were introduced.

History for the SIP Extensions for Caller Identity and Privacy Feature

Release	Modification
12.2(13)T	The features were introduced.

History for the SIP INVITE Request with Malformed Via Header Feature

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	Support was added for additional platforms.

History for the SIP Session Timer Support Feature

Release	Modification
12.2(11)T	These features were introduced.
12.4(9)T	This feature was updated to support RFC 4028.

History for the SIP: Cisco IOS Gateway Reason Header and Buffered Calling Name Completion Feature

Release	Modification
12.3(8)T	This feature was introduced.

History for the SIP: SIP Header/URL Support and SUBSCRIBE/NOTIFY for External Triggers Feature

Release	Modification
12.3(4)T	This feature was introduced.

History for the SIP Stack Portability Feature

Release	Modification
12.4(1)T	This feature was introduced.

History for the SIP: Domain Name Support in SIP Headers Feature

Release	Modification
12.4(2)T	This feature was introduced.

History for the SIP Gateway Support for SDP Session Information and Permit Hostname CLI Feature

Release	Modification
12.4(9)T	This feature was introduced.

History for the Outbound Proxy Support for the SIP Gateway

Release	Modification
12.4(15)T	This feature was introduced.
12.4(20)T	Support was added for disabling outbound proxy support for SIP on a per dial peer basis

History for the SIP Support for PAI

Release	Modification
12.4(15)T	This feature was introduced.

History for the SIP History-info Header Support Feature

Release	Modification
12.4(22)T	This feature was introduced.
15.1(2)T	Support was added to enhance the privacy by enabling Cisco UBE to summarize the content of the history-info header in the outgoing message without letting out any internal topology information.

Feature History for Support for SIP 181 Call is Being Forwarded Message

Release	Modification
15.0(1)XA	Support for SIP 181 Call is Being Forwarded message was added to Cisco IOS SIP TDM gateways and Cisco UBEs.
15.1(1)T	This feature was integrated into this release.

Feature History for Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

Release	Modification
15.0(1)XA	Support for resetting the Expires timer upon receipt of SIP 183 messages on Cisco Unified Communications Manager Express (Cisco Unified CME), a Cisco IOS voice gateway, or a Cisco UBE.
15.1(1)T	This feature was integrated into this release.

Feature History for Support for Stripping off Progress Indication from Incoming ISDN Messages on SIP and H.323 TDM Gateways

Release	Modification
15.0(1)XA	Support for stripping off progress indication from incoming ISDN messages on Cisco IOS SIP and H.323 TDM gateways and on Cisco UBEs.
15.1(1)T	This feature was integrated into this release.

History for the SIP: Via Header Support Feature

Release	Modification
15.1(1)T	This feature was introduced.

History for the SIP: SIP Trunk Registration feature

Release	Modification
15.1(2)T	This feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Message Timer and Response Features

All SIP Message Components, Session Timers, and Responses Features

- Ensure that the gateway has voice functionality that is configurable for SIP.

- Establish a working IP network. Refer to the following Cisco IOS IP Configuration Guides by navigating to them from the Product Support page (http://www.cisco.com/web/psa/products/index.html?c=268438303) according to your Cisco IOS release):
  - *Cisco IOS IP Addressing Services Configuration Guide*
  - *Cisco IOS IP Mobility Configuration Guide*
  - *Cisco IOS IP Multicast Configuration Guide*
  - *Cisco IOS IP Routing Protocols Configuration Guide*
- Configure VoIP.
- Configure SIP voice functionality.

SIP: Cisco IOS Gateway Reason Header and Buffered Calling Name Completion Feature

- For the reason header, do the following:
  - Configure the CLI reason-header override, in SIP user-agent (SIP UA) configuration mode, if you want the Reason header to take precedence over existing cause-code-mapping tables on the gateway receiving Reason header.
- For buffered calling name completion (such as buffer-invite timers), do the following:
  - Complete the prerequisites associated with the Support for the ISDN Calling Name Display feature in release 12.3(4)T (refer to the "Configuring SIP DTMF Features" chapter).
  - Configure a buffer invite timer value.
- Ensure that the incoming ISDN setup contains a name-to-follow indication as described in Generic Requirements for ISDN Calling Name Identification Services for Primary Rate Interface (PRI) specification, GR-1367.

Restrictions for SIP Message Timer and Response Features

All SIP Message Components, Session Timers, and Responses Features

- Via handling for TCP is not implemented.

SIP Permit Hostname Command Features

- The maximum length of a hostname is 30 characters; SIP INVITE message support will truncate any hostname over 30 characters.

SIP Accept-Language Header Support Feature

- The Accept-Language header provided by the inbound SIP call leg is passed to the outbound call leg only if that call leg is SIP as well.

SIP Extensions for Caller Identity and Privacy Feature

- This feature does not support the Anonymity header described in the Internet Engineering Task Force (IETF) specification, draft-ietf-privacy-.02.txt. The feature implements presentation level anonymity at

Layer 5, rather than at the IP address level. Since the SIP gateway assumes that all adjacent signaling devices are trusted, it is recommended that border SIP proxy servers enforce anonymity policies at administrative boundaries.

- The IETF specification, draft-ietf-privacy-.02.txt, for mapping of North American Numbering Plan Area (NANPA) defined Automatic Number Identification Information Indicators (ANI II) or Originating Line Information (OLI) digits, is still under development. The current implementation of Cisco IOS VoiceXML supports carrying the ANI II digits as digits, rather than as a string representation of the numbering plan-tagged ANI II digits.

SIP INVITE Request with Malformed Via Header Feature

- Distributed Call Signaling (DCS) headers and extensions are not supported.

SIP Session Timer Support Feature

- This feature enables the SIP Portable stack and IOS gateway to comply with IETF RFC 4028 specification for SIP session timer.
- Cisco SIP gateways cannot initiate the use of SIP session timers but do fully support session timers if another user agent requests it.
- The Min-SE value can be set only by using the **min-se** command described in this document. It cannot be set using the CISCO-SIP-UA-MIB.

SIP: SIP Header/URL Support and SUBSCRIBE/NOTIFY for External Triggers Feature

- For outbound calls, an application is allowed to pass any extended or nonstandard header except for the following:
  - Call-ID
  - Supported
  - Require
  - Min-SE
  - Session-Expires
  - Max-Forwards
  - CSeq
  - The “Tag” parameter within From and To headers (From and To headers themselves are allowed)

All other headers may be overwritten by the application to create the header lines in the SIP INVITE message.

- SUBSCRIBE and NOTIFY methods are supported for Tool Command Language (Tcl) applications only.

SIP Gateway Support for SDP Session Information Feature

- The maximum length of a received session information line is 1000 characters; SIP gateway support truncates any session information line over 1000 characters.

SIP: SIP Support for PAI

- Privacy for REGISTER messages is not supported. When a gateway registers with another endpoint, the gateway assumes this endpoint is within the trusted domain, therefore privacy regarding this transaction is unnecessary.

SIP History-info Header Support Feature

- History-info header support is provided on Cisco IOS SIP time-division multiplexing (TDM) gateways and SIP-SIP Cisco Unified Border Elements (Cisco UBEs) only.
- Cisco IOS SIP gateways cannot use the information in the history-info header to make routing decisions.

Information About SIP Message Components Session Timers and Response Features

Internal Cause Code Consistency Between SIP and H.323

The Internal Cause Code Consistency Between SIP and H.323 feature establishes a standard set of categories for internal causes of voice call failures. Before this feature, the cause code that was passed when an internal failure occurred was not standardized or based on any defined rules. The nonstandardization lead to confusing or incorrect cause code information, and possibly contributed to billing errors.

This feature establishes a standard set of categories for internal causes of voice call failures. Internal cause-code consistency enables more efficient and effective operation of combined SIP and H.323 networks, which reduces operational expenses and improves service availability.

Note RFC 2543-bis-04 enhancements obsolete the SIP cause codes 303 *Redirection: See Other* and 411 *Client Error: Length required*. For information on RFC 2543-bis-04 enhancements, refer to the "Achieving SIP RFC Compliance" chapter.

H.323 and SIP standard cause codes that are now generated accurately reflect the nature of each internal failure. This capability makes the H.323 and SIP call control protocols consistent with cause codes that are generated for common problems. Also, for each internal failure, an ITU-T Q.850 release cause code is also assigned and the table below maps the new standard categories with the Q.850 release cause code and description that is assigned to each category.

Table 23: H.323 and SIP Standard Category and Q.850 Cause Code Mapping

Standard Category	Standard Category Description	Q.850 Cause Code	Q.850 Release Cause Description
Socket Failure	Typical scenarios: • Transmission Control Protocol (TCP) socket connection failure. • Problem sending an H.323 SETUP. • Problem sending a SIP INVITE. • Send or receive error occurs on connected socket.	27	CC_CAUSE_DESTINATION_OUT_OF_ORDER The destination indicated by the user cannot be reached because the destination's interface is not functioning correctly. The signaling message was unable to be delivered to the remote party.
Destination Address Resolution Failure	Typical scenarios: • Domain Name System (DNS) resolution failure. • Invalid session target in configuration.	3	CC_CAUSE_NO_ROUTE The called party cannot be reached because the network that the call has been routed through does not serve the desired destination.
Call Setup Timeout Failure	Typical scenarios: • No H.323 call proceeding. • No H.323 alerting or connect message received from the terminating gateway. • Invite expires timer reached maximum number of retries allowed.	102	CC_CAUSE_RECOVERY_ON_ TIMER_EXPIRY A procedure has been initiated by the expiry of a timer in association with error handling procedures.
Internal Resource Allocation Failure	Typical scenarios: • Out of memory. • Internal access to the TCP socket becomes unavailable.	47	CC_CAUSE_NO_RESOURCE A "resource unavailable" event has occurred.
Invalid Message Received Error	Typical scenarios: • An invalid message was received.	95	CC_CAUSE_INVALID_MESSAGE An invalid message event has occurred.

Standard Category	Standard Category Description	Q.850 Cause Code	Q.850 Release Cause Description
Mandatory IE Missing Error	Typical scenarios: • Mandatory Contact field missing in SIP message. • Session Description Protocol (SDP) body is missing.	96	CC_CAUSE_MANDATORY_IE_ MISSING The equipment sending this cause has received a message that is missing an information element (IE). This information element must be present in the message before the message can be processed.
Invalid IE Contents Error	Typical scenarios: • SIP Contact field is present, but format is bad.	100	CC_CAUSE_INVALID_IE_ CONTENTS The equipment sending this cause code has received an information element that it has implemented. However, the equipment sending this cause code has not implemented one or more of the specific fields.
Message in Invalid Call State	Typical scenarios: • An unexpected message was received that is incompatible with the call state.	101	CC_CAUSE_MESSAGE_IN_ INCOMP_CALL_STATE Indicates that a message has been received that is incompatible with the call state.
Internal Error	Typical scenarios: • Failed to send message to Public Switched Telephone Network (PSTN).	127	CC_CAUSE_INTERWORKING There has been interworking with a network that does not provide causes for actions it takes. Precise cause cannot be ascertained.
QoS Error	Typical scenarios: • Quality of service (QoS) error.	49	CC_CAUSE_QOS_UNAVAILABLE The requested QoS cannot be provided.
Media Negotiation Failure	Typical scenarios: • No codec match occurred. • H.323 or H.245 problem leading to failure in media negotiation.	65	CC_CAUSE_BEARER_CAPABILITY_NOT_IMPLEMENTED The equipment sending this cause does not support the bearer capability requested.

SIP - Configurable PSTN Cause Code Mapping

For calls to be established between a SIP network and a PSTN network, the two networks must be able to interoperate. One aspect of their interoperation is the mapping of PSTN cause codes, which indicate reasons for PSTN call failure or completion, to SIP status codes or events. The opposite is also true: SIP status codes or events are mapped to PSTN cause codes. Event mapping tables found in this document show the standard or default mappings between SIP and PSTN.

However, you may want to customize the SIP user-agent software to override the default mappings between the SIP and PSTN networks. The SIP - Configurable PSTN Cause Code Mapping feature allows you to configure specific map settings between the PSTN and SIP networks. Thus, any SIP status code can be mapped to any PSTN cause code, or vice versa.

When set, these settings can be stored in the NVRAM and are restored automatically on bootup.

Default Mappings

The table below lists PSTN cause codes and the corresponding SIP event mappings that are set by default. Any code other than the codes listed are mapped by default to *500 Internal server error* .

Table 24: Default PSTN Cause Code to SIP Event Mappings

PSTN Cause Code	Description	SIP Event
1	Unallocated number	404 Not found
2	No route to specified transit network	404 Not found
3	No route to destination	404 Not found
17	User busy	486 Busy here
18	No user response	480 Temporarily unavailable
19	No answer from the user	
20	Subscriber absent	
21	Call rejected	403 Forbidden
22	Number changed	410 Gone
26	Non-selected user clearing	404 Not found
27	Destination out of order	404 Not found
28	Address incomplete	484 Address incomplete
29	Facility rejected	501 Not implemented
31	Normal, unspecified	404 Not found
34	No circuit available	503 Service unavailable
38	Network out of order	503 Service unavailable
41	Temporary failure	503 Service unavailable
42	Switching equipment congestion	503 Service unavailable
47	Resource unavailable	503 Service unavailable
55	Incoming class barred within Closed User Group (CUG)	403 Forbidden

PSTN Cause Code	Description	SIP Event
57	Bearer capability not authorized	403 Forbidden
58	Bearer capability not presently available	501 Not implemented
65	Bearer capability not implemented	501 Not implemented
79	Service or option not implemented	501 Not implemented
87	User not member of Closed User Group (CUG)	503 Service Unavailable
88	Incompatible destination	400 Bad request
95	Invalid message	400 Bad request
102	Recover on Expires timeout	408 Request timeout
111	Protocol error	400 Bad request
Any code other than those listed above:	**500 Internal server error**	

The table below lists the SIP events and the corresponding PSTN cause codes mappings that are set by default.

Table 25: Default SIP Event to PSTN Cause Code Mapping

SIP Event	PSTN Cause Code	Description
400 Bad request	127	Interworking, unspecified
401 Unauthorized	57	Bearer capability not authorized
402 Payment required	21	Call rejected
403 Forbidden	57	Bearer capability not authorized
404 Not found	1	Unallocated number
405 Method not allowed	127	Interworking, unspecified
406 Not acceptable		
407 Proxy authentication required	21	Call rejected
408 Request timeout	102	Recover on Expires timeout
409 Conflict	41	Temporary failure
410 Gone	1	Unallocated number
411 Length required	127	Interworking, unspecified
413 Request entity too long		
414 Request URI (URL) too long		

SIP Event	PSTN Cause Code	Description
415 Unsupported media type	79	Service or option not implemented
420 Bad extension	127	Interworking, unspecified
480 Temporarily unavailable	18	No user response
481 Call leg does not exist	127	Interworking, unspecified
482 Loop detected		
483 Too many hops		
484 Address incomplete	28	Address incomplete
485 Address ambiguous	1	Unallocated number
486 Busy here	17	User busy
487 Request cancelled	127	Interworking, unspecified
488 Not acceptable here	127	Interworking, unspecified
500 Internal server error	41	Temporary failure
501 Not implemented	79	Service or option not implemented
502 Bad gateway	38	Network out of order
503 Service unavailable	63	Service or option unavailable
504 Gateway timeout	102	Recover on Expires timeout
505 Version not implemented	127	Interworking, unspecified
580 Precondition Failed	47	Resource unavailable, unspecified
600 Busy everywhere	17	User busy
603 Decline	21	Call rejected
604 Does not exist anywhere	1	Unallocated number
606 Not acceptable	58	Bearer capability not presently available

Benefits of SIP - Configurable PSTN Cause Code Mapping

The feature offers control and flexibility. By using CLI commands, you can easily customize the default or standard mappings that are currently available between PSTN and SIP networks. This allows for flexibility when setting up deployment sites.

SIP Accept-Language Header Support

The SIP Accept-Language Header Support feature introduces support for the Accept-Language header in SIP INVITE messages and in OPTIONS responses. This feature enables you to configure up to nine languages to be carried in SIP messages and to indicate multiple language preferences of first choice, second choice, and so on.

Feature benefits include the following:

- Allows service providers to support language-based features
- Allows VXML applications providers to support language-based services

To configure Accept-Language header support, you need to understand the following concepts:

Feature Design of SIP Accept-Language Header Support

Cisco implements this feature on SIP trunking gateways by supporting a new header, Accept-Language, as defined in the Internet Engineering Task Force (IETF) specification, draft-ietf-sip-rfc2543bis-09, SIP: Session Initiation Protocol. The Accept-Language header is used in SIP INVITEs, which establish media sessions between user agents, and in SIP OPTIONS responses, which list user-agent capabilities. The header specifies language preferences for reason phrases, session descriptions, or status responses. A SIP proxy may also use the Accept-Language header to route to a human operator.

The Accept-Language header supports 139 languages, as specified in the International Organization for Standardization (ISO) specification, ISO 639: Codes for Representation of Names of Languages. The SIP Accept-Language Header Support feature allows you to configure up to nine languages to be carried in INVITE messages and OPTIONS responses. This feature also supports the qvalue (q=) parameter, which allows you to indicate multiple language preferences, that is, first choice, second choice, and so on.

Sample INVITE Message

The following is a sample outgoing INVITE message for a gateway configured to support the Sindhi, Zulu, and Lingala languages.

```
11:38:42: Sent:
INVITE sip:36602@172.18.193.120:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.98:5060
From: <sip:172.18.193.98>;tag=27FB000-42E
To: <sip:36602@172.18.193.120>
Date: Mon, 01 Mar 1993 11:38:42 GMT
Call-ID: 23970D87-155011CC-8009E835-18264FDE@172.18.193.98
Supported: timer,100rel
Min-SE:  1800
Cisco-Guid: 0-0-0-0
User-Agent: Cisco-SIPGateway/IOS-12.x
Accept-Language: sd, zu, ln;q=0.123
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE, NOTIFY, INFO
CSeq: 101 INVITE
Max-Forwards: 10
Remote-Party-ID: <sip:172.18.193.98>;party=calling;screen=no;privacy=off
Timestamp: 730985922
Contact: <sip:172.18.193.98:5060>
Expires: 300
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 322
v=0
```

```
o=CiscoSystemsSIP-GW-UserAgent 5606 9265 IN IP4 172.18.193.98
s=SIP Call
c=IN IP4 172.18.193.98
t=0 0
m=audio 16434 RTP/AVP 18 100 101
c=IN IP4 172.18.193.98
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:10
```

Sample OPTIONS Response

The following is a sample OPTIONS response from a gateway configured to support the Yoruba, Sindhi, and English languages.

```
11:28:44: Received:
OPTIONS sip:36601@172.18.193.98:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060
From: "user" <sip:36602@172.18.193.120>
To: <sip:36601@172.18.193.98>
Date: Mon, 01 Mar 1993 02:55:01 GMT
Call-ID: BB8A5738-14EE11CC-8008B310-2C18B10E@172.18.193.120
Accept: application/sdp
CSeq: 110 OPTIONS
Contact: <sip:36601@172.18.193.98:5060>
Content-Length: 0
11:28:44: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  172.18.193.120:5060
From: "user" <sip:36602@172.18.193.120>
To: <sip:36601@172.18.193.98>;tag=2768F24-1DB2
Date: Mon, 01 Mar 1993 11:28:44 GMT
Call-ID: BB8A5738-14EE11CC-8008B310-2C18B10E@172.18.193.120
Server: Cisco-SIPGateway/IOS-12.x
Content-Type: application/sdp
CSeq: 110 OPTIONS
Supported: 100rel
Accept-Language: yo, sd;q=0.234, en;q=0.123
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE, NOTIFY, INFO
Accept: application/sdp
Allow-Events: telephone-event
Content-Length: 170
v=0
o=CiscoSystemsSIP-GW-UserAgent 7292 5756 IN IP4 172.18.193.98
s=SIP Call
c=IN IP4 172.18.193.98
t=0 0
m=audio 0 RTP/AVP 18 0 8 4 2 15 3
c=IN IP4 172.18.193.98
```

SIP Enhanced 180 Provisional Response Handling

The SIP Enhanced 180 Provisional Response Handling feature provides the ability to enable or disable early media cut-through on Cisco IOS gateways for SIP 180 response messages. The feature allows you to specify whether 180 messages with Session Description Protocol (SDP) are handled in the same way as 183 responses with SDP. The 180 Ringing message is a provisional or informational response used to indicate that the

INVITE message has been received by the user agent and that alerting is taking place. The 183 Session Progress response indicates that information about the call state is present in the message body media information. Both 180 and 183 messages may contain SDP which allow an early media session to be established prior to the call being answered.

Prior to the implementation of this feature, Cisco gateways handled a 180 Ringing response with SDP in the same manner as a 183 Session Progress response; that is, the SDP was assumed to be an indication that the far end would send early media. Cisco gateways handled a 180 response without SDP by providing local ringback, rather than early media cut-through. This feature provides the capability to ignore the presence or absence of SDP in 180 messages, and as a result, treat all 180 messages in a uniform manner. The SIP Enhanced 180 Provisional Response Handling feature introduces the new disable-early-media 180 command that enables you to specify which call treatment, early media or local ringback, is provided for 180 responses with SDP.

The table below shows the call treatments available with this feature.

Table 26: Call Treatments with SIP Enhanced 180 Provisional Response Handling Feature

Response Message	Cisco IOS VoiceXML Status	Treatment
180 response with SDP	Enabled (default)	Early media cut-through
180 response with SDP	Disabled	Local ringback
180 response without SDP	Not affected by the new feature	Local ringback
183 response with SDP	Not affected by the new feature	Early media cut-through

SIP Extensions for Caller Identity and Privacy

To configure the SIP Extensions for Caller Identity and Privacy feature, you must understand the following concepts:

Privacy Screening and Presentation Indicators

Cisco implements this feature on SIP trunking gateways by supporting a header, Remote-Party-ID, as defined in the IETF specification, draft-ietf-privacy-.02.txt, SIP Extensions for Caller Identity and Privacy. The Remote-Part-ID header identifies the calling party and carries presentation and screening information. In previous SIP implementations, the From header was used to indicate calling party identity, and once defined in the initial INVITE request, could not be modified for that session. Implementing the Remote-Part-ID header, which can be modified, added, or removed as a call session is being established, overcomes previous limitations and enables call participant privacy indication, screening, and verification. The feature uses the Remote-Part-ID header to support translation capability between Integrated Services Digital Networks (ISDN) messages and Remote-Party-ID SIP tags. The SIP header also enables support for certain telephony services, and some regulatory and public safety requirements, by providing screening and presentation indicators.

The SIP Extensions for Caller Identity and Privacy feature introduces command-line interface (CLI) commands to enable remote-party-id translations and to configure alternative calling information treatments for calls entering the SIP trunking gateway. Configurable treatment options are:

- Calling name and number pass-through (default).
- No calling name or number sent in the forwarded Setup message.
- Calling name unconditionally set to the configured string in the forwarded Setup message.

- Calling number unconditionally set to the configured string in the forwarded Setup message.

You can configure alternative calling information treatments for calls exiting the SIP trunking gateway. Configurable treatment options are as follows:

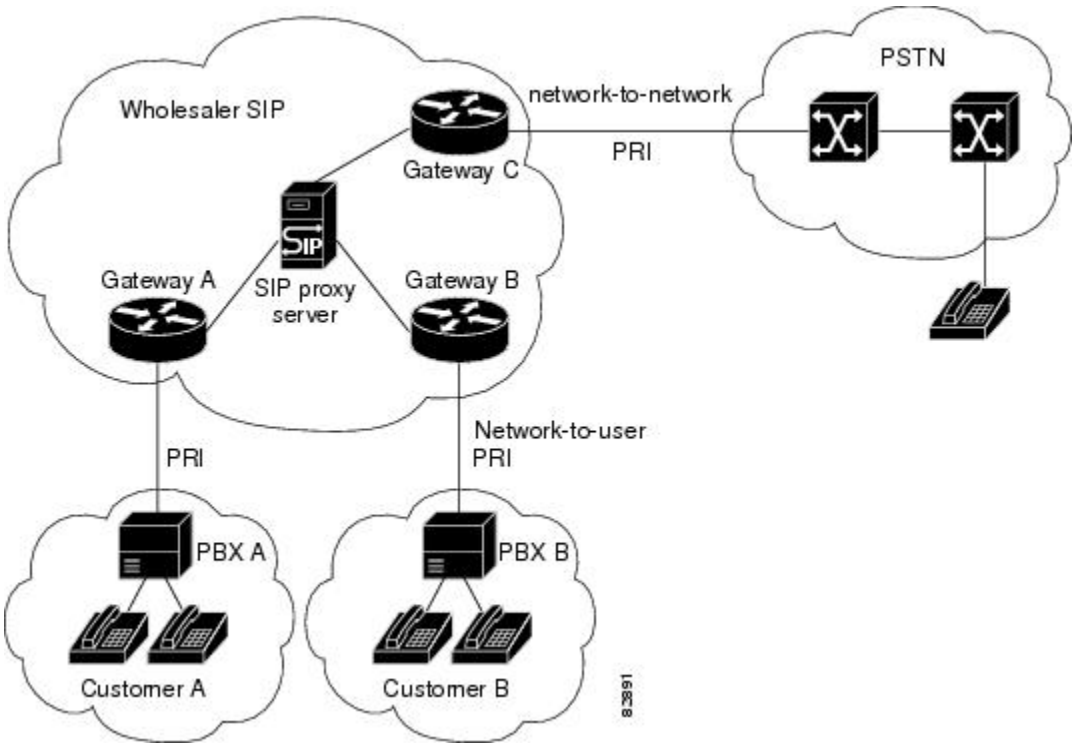
- Calling name and number pass-through (default).
- No calling name or number sent in the forwarded INVITE message.
- Display-name of the From header unconditionally set to the configured string in the forwarded INVITE message.
- User part of the From header unconditionally set to the configured string in the forwarded INVITE message.
- Display-name of the Remote-Party-ID header unconditionally set to the configured string in the forwarded INVITE message.
- User part of the Remote-Party-ID header unconditionally set to the configured string in the forwarded INVITE message.

Remote-Party-ID Implementation

This section discusses the implementation of the Remote-Party-ID feature in a SIP network. Before the implementation of this feature, there was no mechanism to modify the contents of the From header field. With the feature enabled, SIP gateways provide translation capability for ISDN screening and presentation identifiers in call setup messages. SIP gateways and proxy servers require configuration to support Remote-Party-ID feature.

The figure below shows a typical network where the feature is implemented. Gateway C is configured for unscreened discard, that is, if the incoming SIP INVITE request does not contain a screened Remote-Part-ID header (;screen=yes), no calling name or number is sent in the forwarded Setup message.

Figure 34: Wholesaler SIP Network



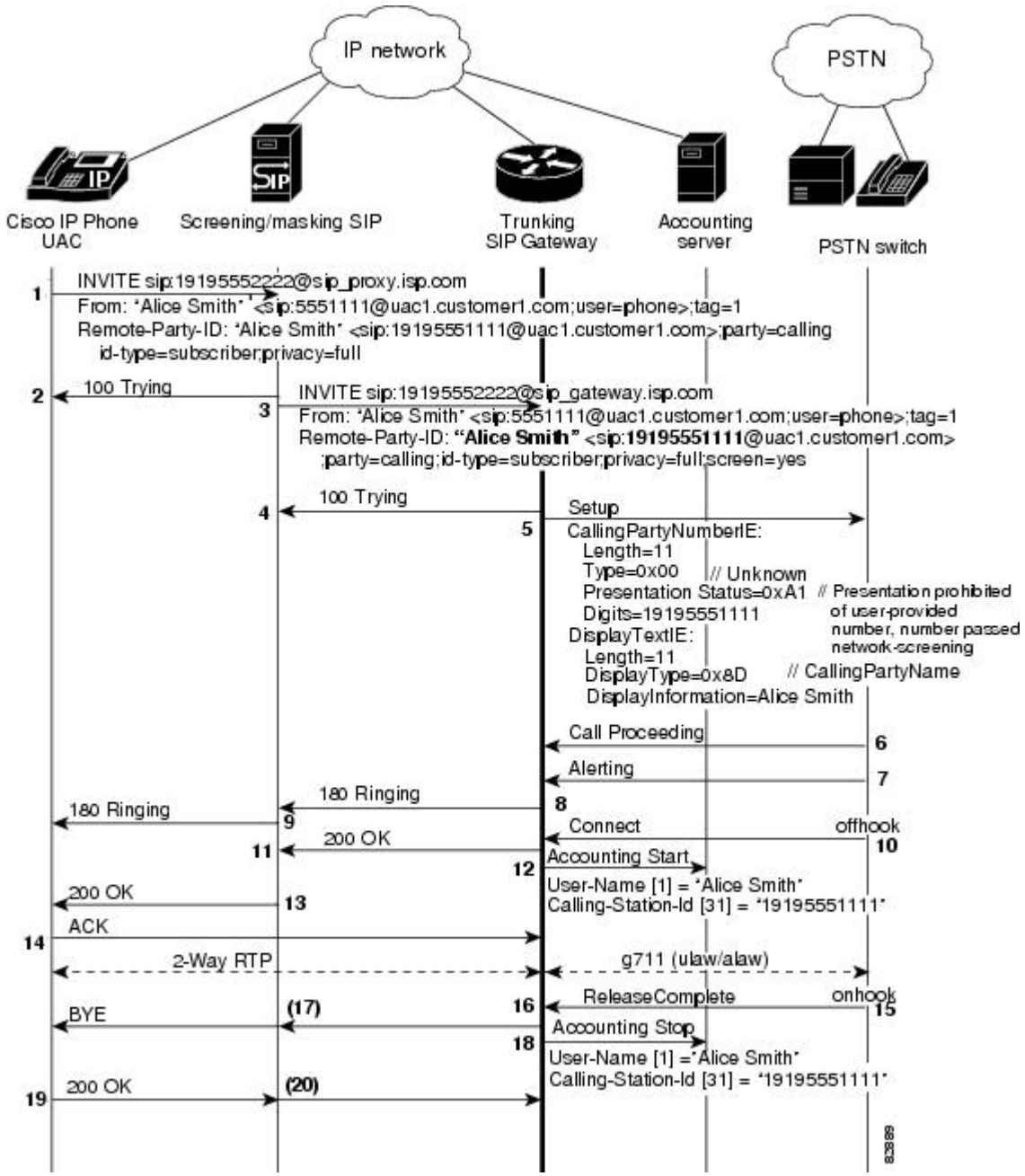
Note With respect to privacy and screening indication, it is the responsibility of the proxy server to protect display-name information and enforce privacy policies at the administrative boundary.

The figures in the Inbound and Outbound Call Flows section illustrate various calling information treatment options using the commands available with the feature. Calling information treatment is determined by the parameters specified in the Setup message and Remote-Party-ID configuration on the SIP gateway.

Inbound and Outbound Call Flows

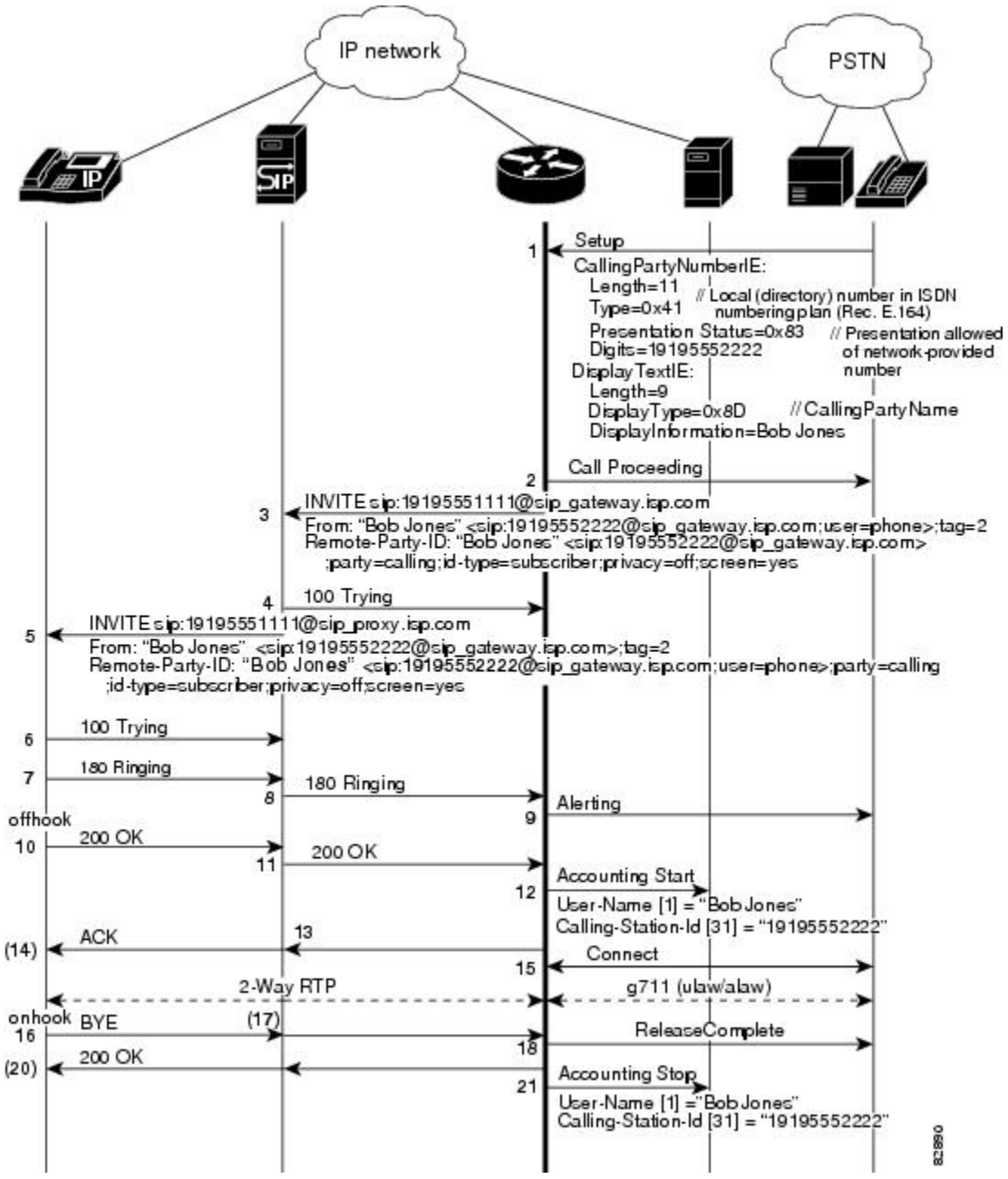
This section presents inbound and outbound call flows for the Remote-Party-ID feature. The figure below shows the SIP-to-PSTN default behavior where the calling party name and number are passed. The feature enables this treatment by default and no configuration is required.

Figure 35: SIP-to-PSTN Default Call Flow with Remote-Party-ID



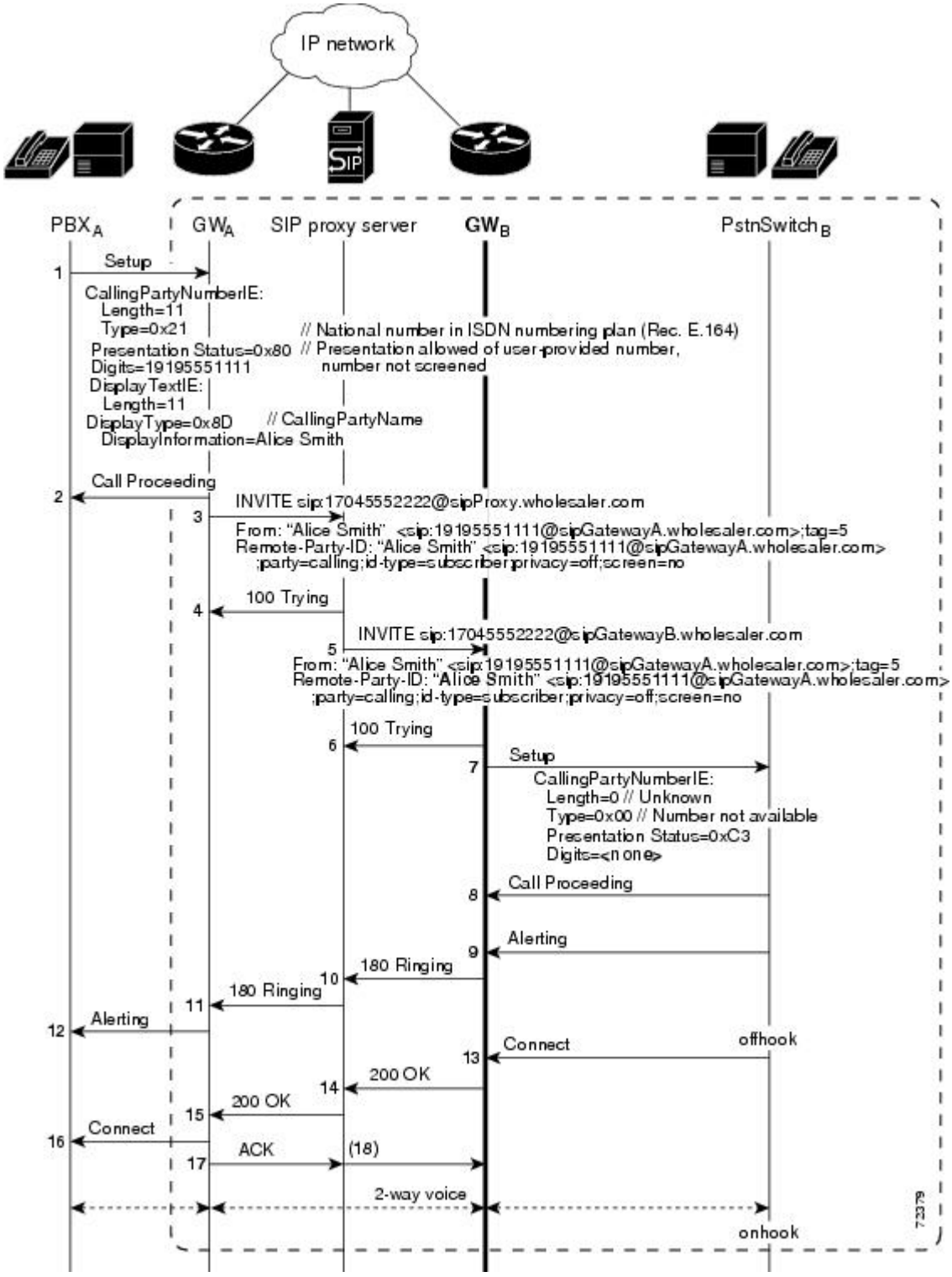
The figure below shows the PSTN-to-SIP default behavior where the calling party name and number are passed. This feature enables this treatment by default and no configuration is required.

Figure 36: PSTN-to-SIP Default Call Flow with Remote-Party-ID Translation, No Privacy Requested



The figure below shows the call flow for discarding the calling name and number at Gateway B. The Setup message includes ISDN information elements (IEs) that specify calling information treatment. The INVITE message from Gateway A includes the corresponding Remote-Party-ID SIP tags.

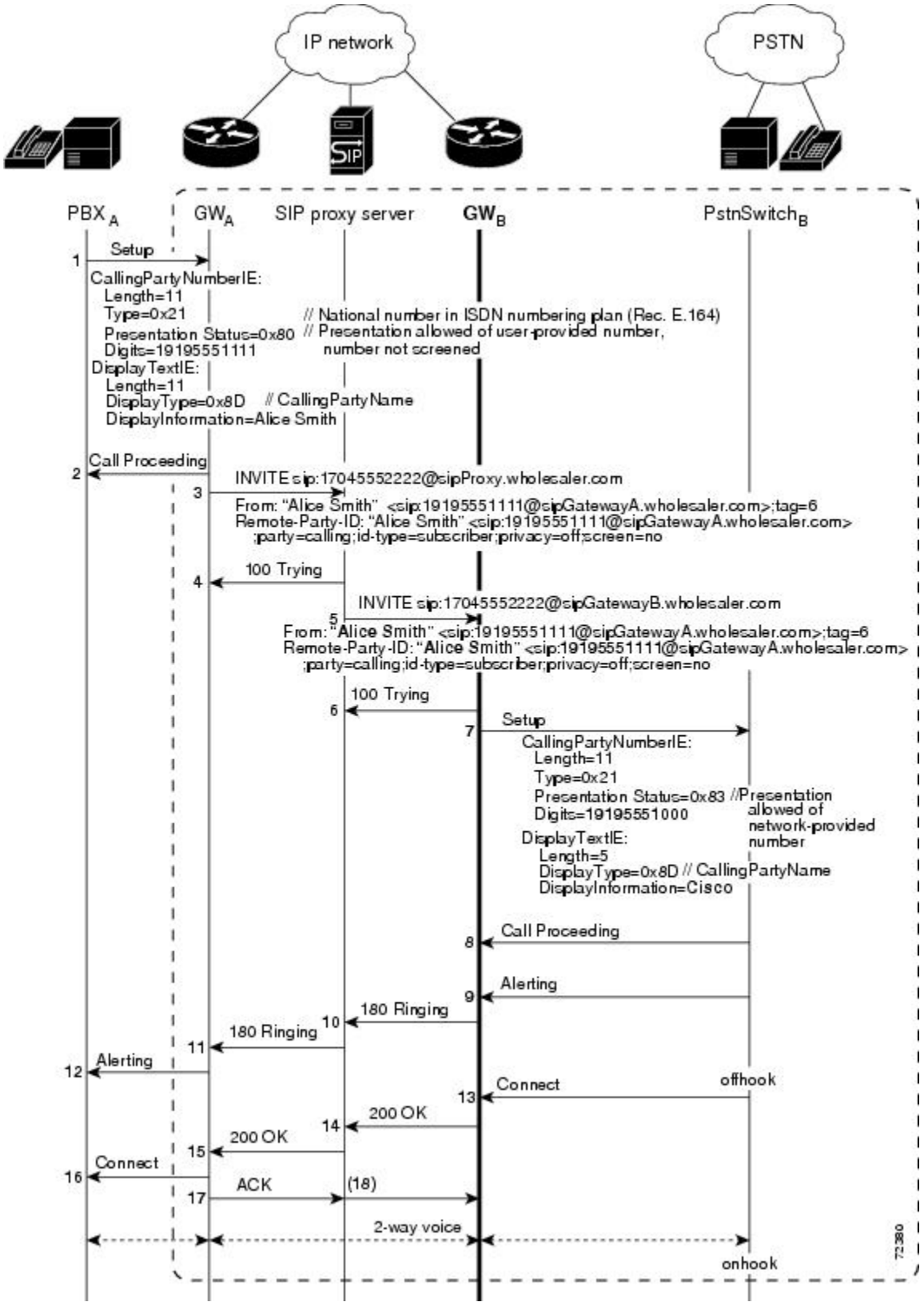
Figure 37: Discarding Calling Name and Number at Gateway



The figure below shows Gateway B overriding the calling name and number received in the Setup message from Gateway A. To configure Gateway B to override calling name and number, use the following commands:

- **remote-party-id**
- calling-info sip-to-pstn name set name
- calling-info sip-to-pstn number set number

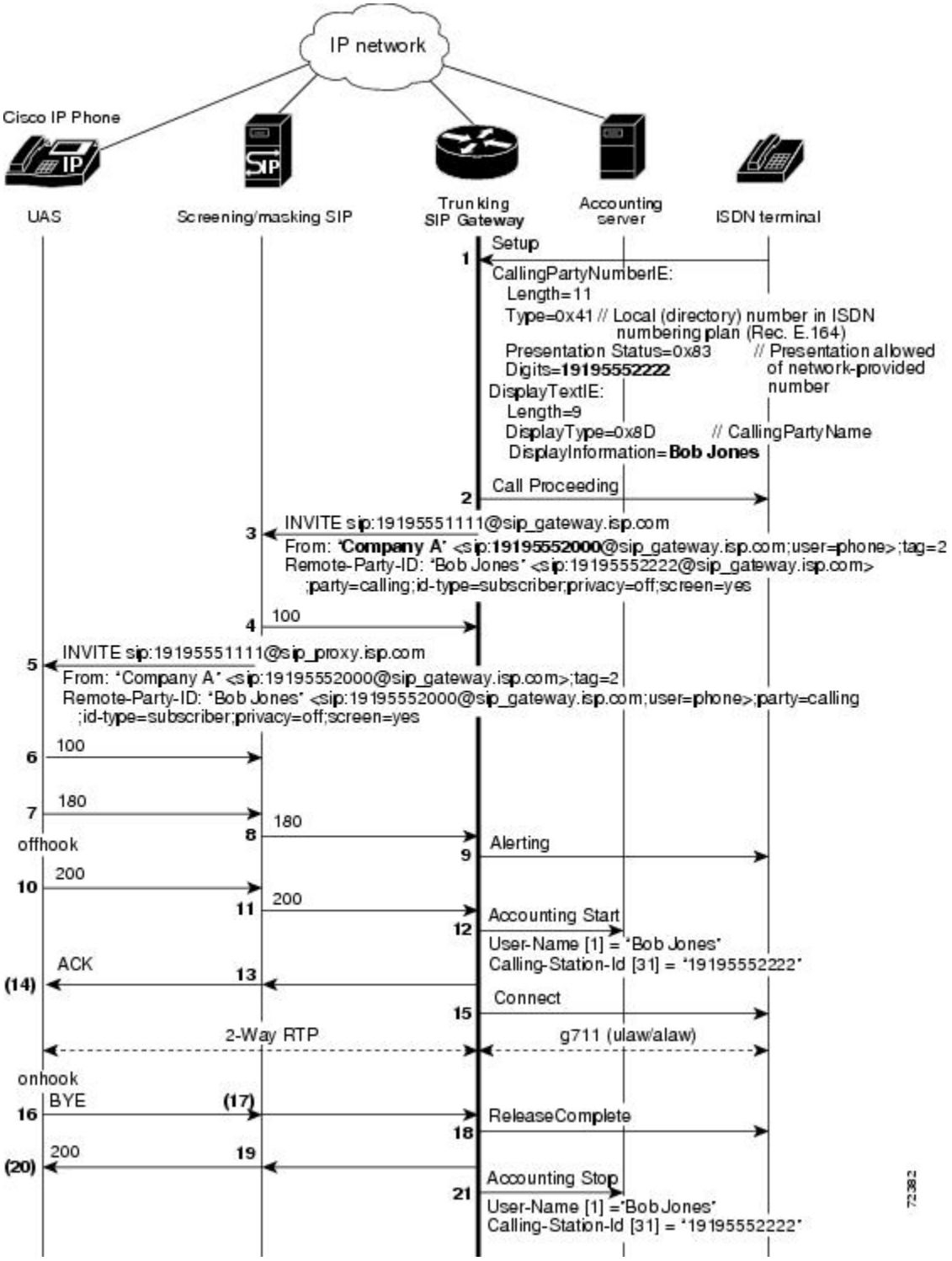
Figure 38: Overriding Calling Name and Number at Gateway


IP network
PSTN
SIP
PBX A
GW A
SIP proxy server
GW B
PstnSwitch B
1 Setup
CallingPartyNumberIE:
Length=11
Type=0x21
Presentation Status=0x80
Digits=19195551111
// National number in ISDN numbering plan (Rec. E.164)
// Presentation allowed of user-provided number, number not screened
DisplayTextIE:
Length=11
DisplayType=0x8D // CallingPartyName
DisplayInformation=Alice Smith
2 Call Proceeding
3 INVITE sip:17045552222@sipProxy.wholesaler.com
From: "Alice Smith" <sip:19195551111@sipGatewayA.wholesaler.com>;tag=6
Remote-Party-ID: "Alice Smith" <sip:19195551111@sipGatewayA.wholesaler.com>
;party=calling;id-type=subscriber;privacy=off;screen=no
4 100 Trying
5 INVITE sip:17045552222@sipGatewayB.wholesaler.com
From: "Alice Smith" <sip:19195551111@sipGatewayA.wholesaler.com>;tag=6
Remote-Party-ID: "Alice Smith" <sip:19195551111@sipGatewayA.wholesaler.com>
;party=calling;id-type=subscriber;privacy=off;screen=no
6 100 Trying
7 Setup
CallingPartyNumberIE:
Length=11
Type=0x21
Presentation Status=0x83
Digits=19195551000
//Presentation allowed of network-provided number
DisplayTextIE:
Length=5
DisplayType=0x8D // CallingPartyName
DisplayInformation=Cisco
8 Call Proceeding
9 Alerting
10 180 Ringing
11 180 Ringing
12 Alerting
13 Connect
offhook
14 200 OK
15 200 OK
16 Connect
17 ACK
(18)
2-way voice
onhook
72380

In the figure below the trunking SIP gateway is configured to override the calling name and number of the From header. To configure this call treatment option, use the following commands:

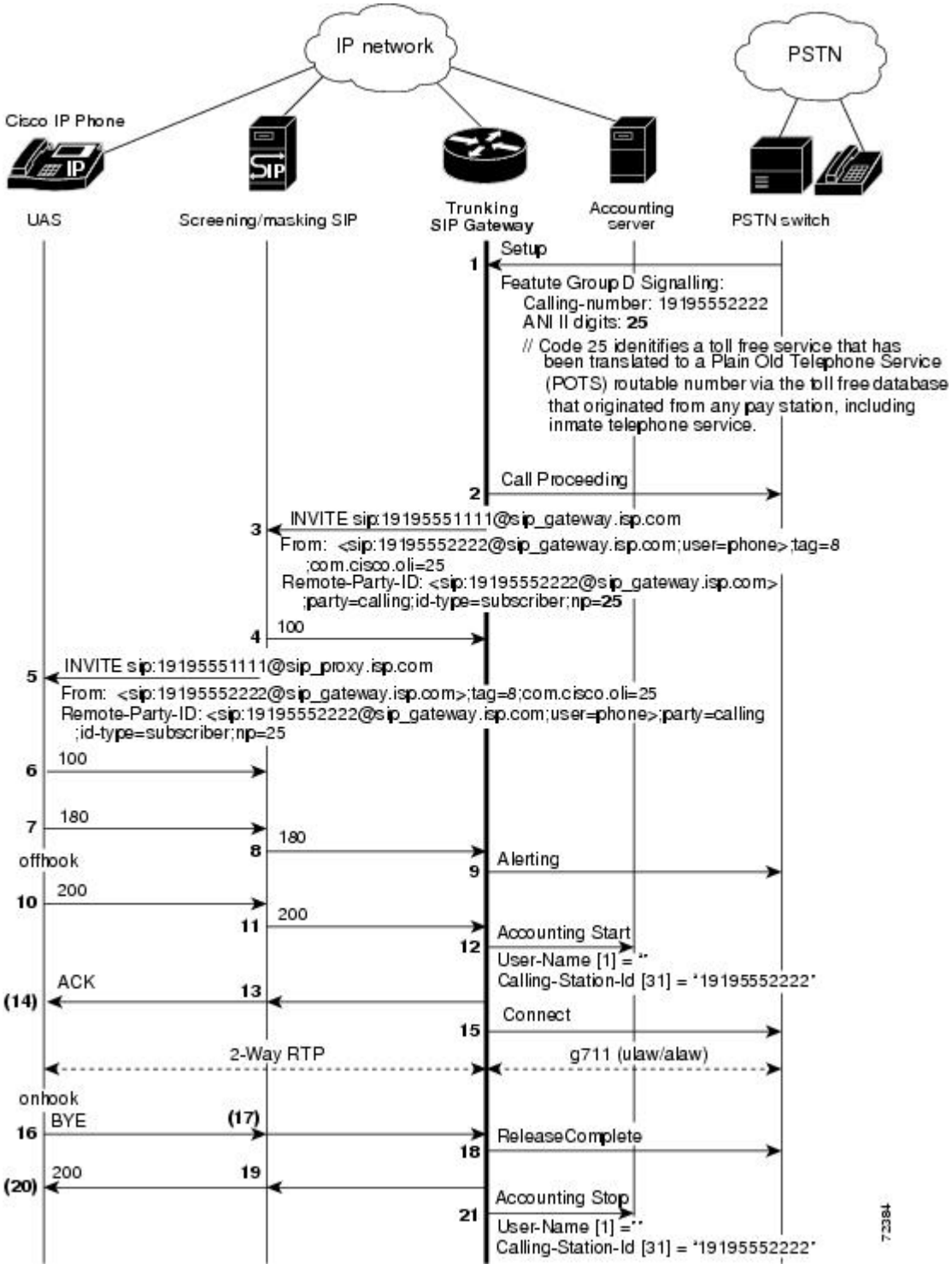
- remote-party-id
- calling-info pstn-to-sip from name set name
- calling-info pstn-to-sip from number set number

Figure 39: Overriding Calling Name and Number of From Header



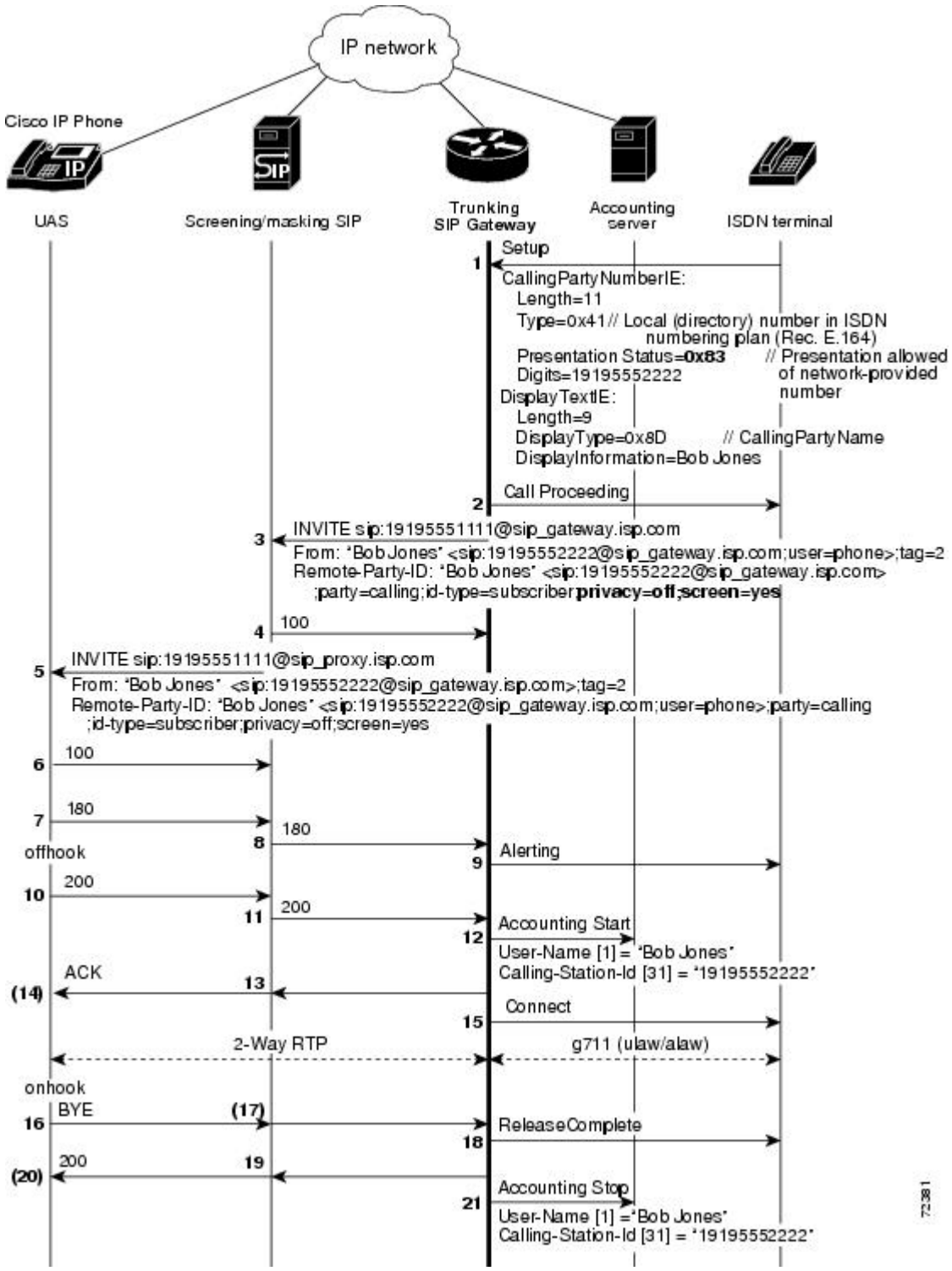
The figure below shows translation of OLI or ANI II digits for a billing application. The Remote-Party-ID feature enables this treatment by default; no configuration tasks are required. If the feature was disabled by using the no remote-party-id command, use the remote-party-id command to re-enable the feature.

Figure 40: Passing OLI from CAS to SIP



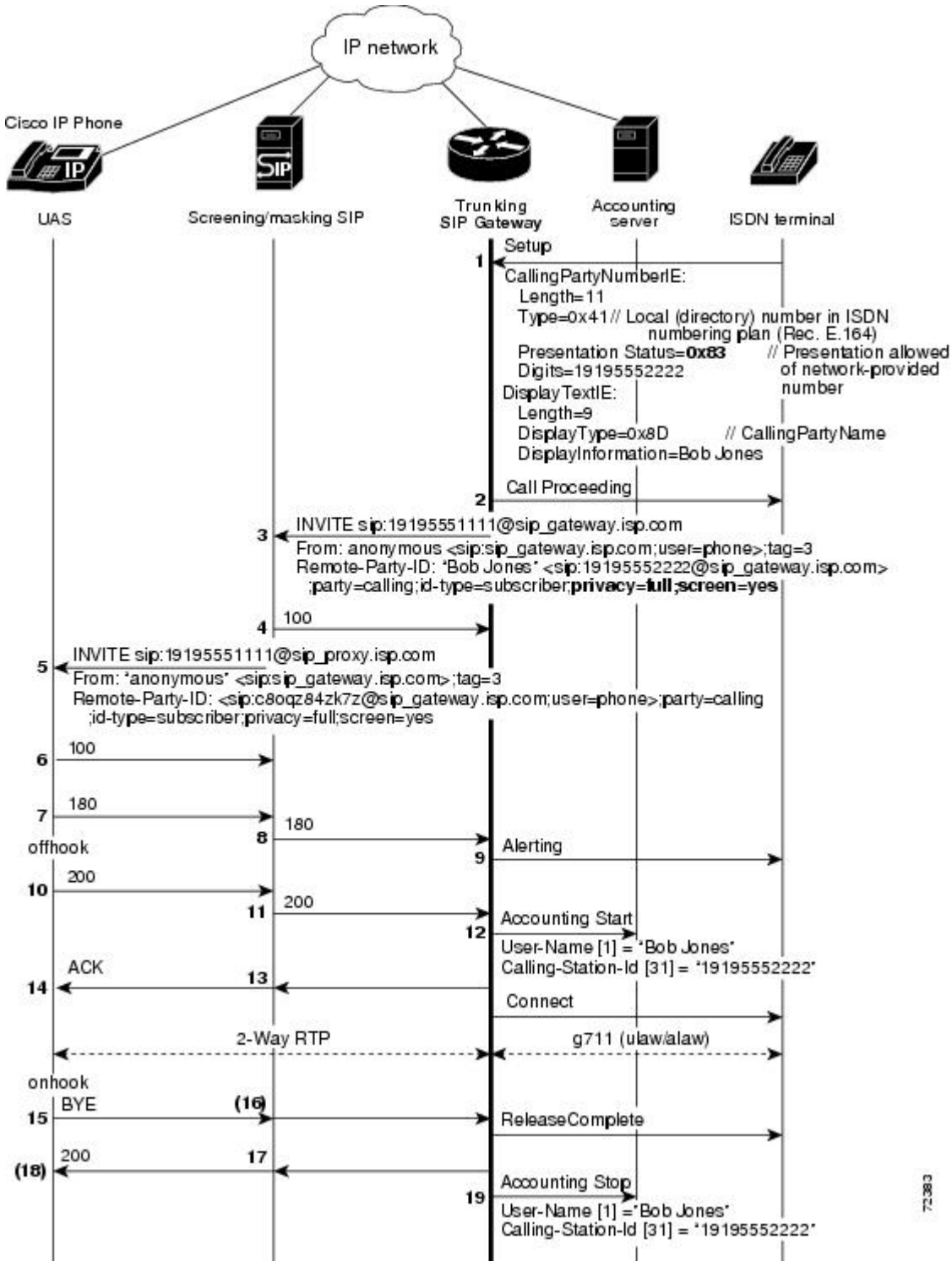
The next two figures show the SIP trunking gateway capability to provide translation between ISDN screening and presentation identifiers and SIP Remote-Party-ID extensions. The two figures show the difference in call treatment, with and without privacy requested. With no privacy requested, the calling party name and number are passed unchanged.

Figure 41: PSTN-to-SIP Call Flow with Remote-Party-ID Translation, No Privacy Requested



With privacy requested, as shown in the figure below, screened identity information is still logged in accounting records for billing information, but the user field is not populated in the From header of the outgoing INVITE message, and the display-name is populated with “anonymous.”

Figure 42: PSTN-to-SIP Call Flow with Remote-Party-ID, Privacy Requested



Remote-Party-ID in SIP and PSTN Messages

The ability to provide marking, screening, and PSTN translation of identity information to and from Remote-Party-ID extensions is supported in SIP INVITE and PSTN messages. This section discusses the formats of SIP INVITE and PSTN messages, and has the following subsections:

Remote-Party-ID Header

The SIP Remote-Party-ID header identifies the calling party and includes user, party, screen and privacy headers that specify how a call is presented and screened. The header contains a URL and an optional display name that identifies a user. A valid Remote-Party-ID header may be either a SIP URL or a TEL URL.

Note For information on header syntax, see "Remote-Party-ID Syntax" and "Screening and Presentation Information".

The following example shows representative Remote-Party-ID headers, including user, party, screen, and privacy.

```
02:32:17:Received:
INVITE sip:3331000@172.27.184.118:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP 10.0.0.1:5070
Supported:org.ietf.sip.100rel
From:"alice" <sip:555-1001@10.0.0.1:5070>
To:sip:555-1002@172.27.184.118:5060
Remote-Party-ID:"Alice Smith"
<sip:5551111@192.0.2.67;user=phone>;party=calling;screen=no;privacy=off
Call-ID:00000001@10.0.0.1:5070
CSeq:1 INVITE
Contact:"alice" <sip:10.0.0.1:5070>
Content-Type:application/sdp
v=0
o=- 2890844526 2890844526 IN IP4 A3C47F2146789F0
s=-
c=IN IP4 10.0.0.1
t=36124033 0
m=audio 49170 RTP/AVP 0
```

Remote-Party-ID Syntax

Remote-Party-ID fields identify the calling party depending upon how the field is marked. If the party is unmarked, a Remote-Party-ID in a header represents the identity of the calling party.

Remote-Party-ID follows the Augmented Backus-Naur Format (ABNF). Refer to draft-ietf-sip-privacy-02.txt for the definitive specification. Fields are as follows:

- Remote-Party-ID = "Remote-Party-ID" ":" [display-name] "<" addr-spec ">" *(";" rpi-token)
- rpi-token = rpi-screen | rpi-pty-type | rpi-id-type | rpi-privacy | other-rpi-token
- rpi-screen = "screen" "=" ("no" | "yes")
- rpi-pty-type = "party" "=" ("calling" | "called" | token)
- rpi-id-type = "id-type" "=" ("subscriber" | "user" | "alias" | "return" | "term" | token)
- rpi-privacy = "privacy" "=" 1#(("full" | "name" | "uri" | "off" | token) ["-" ("network" | token)])
- other-rpi-token = ["-"] token ["=" (token | quoted-string)]

ISDN Syntax

ISDN messages follow the format specified in ISDN Primary Rate Interface Call Control Switching and Signalling Generic Requirements for Class II Equipment, TR-NWT-001268, Revisions 1-4, Telcordia Technologies Technical Reference, 2001 and ISDN Basic Rate Interface Call Control Switching and Signalling Generic Requirements, GR-268-CORE, July 1998, to signal call control. ISDN messages are composed of information elements (IEs). The Cisco IOS VoiceXML feature uses Calling Party Number and Display Text IEs to provide specified screening and presentation treatment. The Calling Party Number IE specifies the origin of the calling number and presentation status, and the Display Text IE supplies calling party name information that is formatted for display by a terminal for a human user. See the Setup message in "ISDN Syntax" for sample IE information.

Screening and Presentation Information

The Remote-Part-ID header and ISDN Setup messages contain tags used to specify screened identity information. The table below lists translation of screening and presentation information included in the Remote-Party-ID SIP tags for SIP to PSTN networks.

Table 27: SIP to PSTN Translation of Screening and Presentation Information

Remote-Party-ID SIP Tags	PSTN Octet 3A
;privacy=off;screen=no	Presentation allowed of user-provided number, number not screened (0x80)
;privacy=off;screen=yes	Presentation allowed of user-provided number, number passed network screening (0x81)
;privacy=[full|uri|name];screen=no	Presentation prohibited of user-provided number, number not screened (0xA0)
;privacy=[full|uri|name];screen=yes	Presentation prohibited of user-provided number, number passed network screening (0xA1)
;screen=no	Presentation allowed of user-provided number, number not screened (0x80)
;screen=yes	Presentation allowed of user-provided number, number passed network screening (0x81)
;privacy=off	Presentation allowed of user-provided number, number not screened (0x80)
;privacy=[full|uri|name]	Presentation prohibited of user-provided number, number not screened (0xA0)
(no screen or privacy tags)	Presentation allowed of user-provided number, number not screened (0x80)

The table below provides the same translation for PSTN to SIP networks.

Table 28: PSTN to SIP Translation of Screening and Presentation Information

PSTN Octet 3A	Remote-Party-ID SIP Tags
Presentation allowed of user-provided number, number not screened (0x80)	;privacy=off;screen=no
Presentation allowed of user-provided number, number passed network screening (0x81)	;privacy=off;screen=yes
Presentation allowed of user-provided number, number failed network screening (0x82)	;privacy=off;screen=no
Presentation allowed of network-provided number (0x83)	;privacy=off;screen=yes
Presentation prohibited of user-provided number, number not screened (0xA0)	;privacy=full;screen=no
Presentation prohibited of user-provided number, number passed network screening (0xA1)	;privacy=full;screen=yes
Presentation prohibited of user-provided number, number failed network screening (0xA2)	;privacy=full;screen=no
Presentation prohibited of network-provided number (0xA3)	;privacy=full;screen=yes
Number not available (0xC3)	(no screen or privacy tags are sent)

The table below lists the corresponding translation for ISDN tags in binary and hex formats.

Table 29: ISDN Tags in Binary and Hex Formats

Binary (Bits) 8 7 6 5 4 3 2 1	Hex	Meaning
1 0 0 0 0 0 0 0	0x80	Presentation allowed of user-provided number, number not screened
1 0 0 0 0 0 0 1	0x81	Presentation allowed of user-provided number, number passed network screening
1 0 0 0 0 0 1 0	0x82	Presentation allowed of user-provided number, number failed network screening
1 0 0 0 0 0 1 1	0x83	Presentation allowed of network-provided number
1 0 1 0 0 0 0 0	0xA0	Presentation prohibited of user-provided number, number not screened
1 0 1 0 0 0 0 1	0xA1	Presentation prohibited of user-provided number, number passed network screening
1 0 1 0 0 0 1 1	0xA3	Presentation prohibited of network-provided number
1 1 0 0 0 0 1 1	0xC3	Number not available

Benefits of SIP Extensions for Caller Identity and Privacy

- Expands PSTN interoperability

- Supports the ability to override privacy and screening indicators
- Enables network verification and screening of a call participant identity by SIP proxy servers
- Supports logging of screened identity information in accounting records for billing information
- Provides enhanced subscriber information that supports the enabling of service creation platforms and application servers for service providers
- Allows the service provider enhanced control of the ability to identify a subscriber and its qualifications within the network

SIP Via Header Support

Each SIP request includes a Via header that may have an maddr (multiple address) parameter. The maddr parameter indicates an alternate address for where to send the SIP response.

The value of the maddr parameter can be an IP address or a hostname. If a maddr parameter is a hostname, the SIP stack performs a DNS query to resolve the name to an IP address. In compliance with RFC 3261, this feature allows the SIP request sender to specify the response destination using the maddr parameter in the Via header.

Note Prior to Cisco IOS Release 15.1(1)T, replies to SIP requests could be sent only to the source IP address (the IP address where the SIP request originated).

The sender of the SIP request is a far-end SIP endpoint with which the Cisco IOS gateway is communicating. The far-end endpoint manages SIP dialogs across multiple nodes. The far-end endpoint is a SIP INVITE request to initiate a new dialog with the Cisco IOS gateway. It uses the maddr parameter in the Via header to specify a destination address for SIP responses. SIP dialog to the Cisco IOS gateway can originate from one IP address, but subsequent responses go to the destination address specified in the maddr parameter.

The following SIP request shows a destination address specified using the maddr parameter in the Via header (192.168.199.200). The SIP response is sent to this address.

```
INVITEsip:1234@10.105.209.114:5060 SIP/2.0
Via: SIP/2.0/TCP 192.168.199.200:5060; branch=z9hG4K0245fc9a5; maddr=192.168.10.11
From: <sip:1234@10.105.209.114>
Date: Tue, 23 Mar 2010 21:42:14 GMT
Call-ID: f06cc480-ba9135b6-1-c8c71ac@192.168.199.200
Supported: timer, resource-priority, replaces
Min-SE: 1800
User-Agent: Cisco-CUCM8.0
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY
CSeq: 101 INVITE
Contact: <sip:3001@192.168.199.200:5060;transport=tcp>
Expires: 1800
Allow- Events: presence, kpml
Supported: X-Cisco-srtp-fallback
Support: Geolocation
Call-Info: <sip:192.168.199.200:5060>;method=”NOTIFY; Event=telephone-event;Duration=500”
Cisco-Guid: 4033660032-3130078646-0000000001-3368489644
Session Expires: 1800
P-Asserted-Identity: <sip:3001@192.168.199.200>
Content-Length: 0
```

SIP INVITE Request with Malformed Via Header

A SIP INVITE requests that a user or service participate in a session. Each INVITE contains a Via header that indicates the transport path taken by the request so far, and where to send a response.

In the past, when an INVITE contained a malformed Via header, the gateway would print a debug message and discard the INVITE without incrementing a counter. However, the printed debug message was often inadequate, and it was difficult to detect that messages were being discarded.

The SIP INVITE Request with Malformed Via Header feature provides a response to the malformed request. A counter, *Client Error: Bad Request,* increments when a response is sent for a malformed Via field. *Bad Request* is a class 400 response and includes the explanation *Malformed Via Field* . The response is sent to the source IP address (the IP address where the SIP request originated) at User Datagram Protocol (UDP) port 5060.

Note This feature applies to messages arriving on UDP, because the Via header is not used to respond to messages arriving on TCP.

Feature benefits include the following:

- The system now increments a counter and sends a response, rather than simply discarding an INVITE message that contains a malformed Via header.
- The counter provides a useful and immediate indication that an INVITE message has been discarded, and the response allows the result to be propagated back to the sender.

SIP Session Timer Support

The SIP Session Timer Support feature adds the capability to periodically refresh SIP sessions by sending repeated INVITE requests. The repeated INVITE requests (re-INVITEs), are sent during an active call leg to allow user agents or proxies to determine the status of a SIP session. Without this keepalive mechanism, proxies that remember incoming and outgoing requests (stateful proxies) may continue to retain call state needlessly. If a user agent fails to send a BYE message at the end of a session or if the BYE message is lost because of network problems, a stateful proxy does not know that the session has ended. The re-INVITEs ensure that active sessions stay active and completed sessions are terminated.

In addition to re-INVITEs, UPDATE can also used as a method for session keepalives. The SIP stack supports both re-INVITE and UPDATE. The gateway continues to use re-INVITE for session refresh.

The SIP Session Timer Support feature also adds two new general headers that are used to negotiate the value of the refresh interval.

- A Session-Expires header is used in an INVITE if the user-agent client (UAC) wants to use the session timer.
- The Minimum Session Expiration (Min-SE) header conveys the minimum allowed value for the session expiration.

Role of the User Agents

The initial INVITE request establishes the duration of the session and may include a Session-Expires header and a Min-SE header. These headers indicate the session timer value required by the UAC. A receiving

user-agent server (UAS) or proxy can lower the session timer value, but not lower than the value of the Min-SE header. If the session timer duration is lower than the configured minimum, the proxy or UAS can also send out a 422 response message. If the UAS or proxy finds that the session timer value is acceptable, it copies the Session-Expires header into the 2*xx* class response.

A UAS or proxy can also insert a Session-Expires header in the INVITE if the UAC did not include one. Thus a UAC can receive a Session-Expires header in a response even if none was present in the request.

In the 2*xx* response, the *refresher* parameter in the Session-Expires header indicates who performs the re-INVITEs or UPDATE. For example, if the parameter contains the value UAC, the UAC performs the refreshes. For compatibility issues, only one of the two user agents needs to support the session timer feature, and in that case, the user agent that supports the feature performs the refreshes.

Re-INVITEs are processed identically to INVITE requests, but go out in predetermined session intervals. Re-INVITEs carry the new session expiration time. The user agent that is responsible for generating re-INVITE requests sends a re-INVITE out before the session expires. If there is no response, the user agent sends a BYE request to terminate the call before session expiration. If a re-INVITE is not sent before the session expiration, either the UAC or the UAS can send a BYE.

If the 2*xx* response does not contain a Session-Expires header, there is no session expiration and re-INVITEs do not need to be sent.

Session-Expires Header

The Session-Expires header conveys the session interval for a SIP call. It is placed in an INVITE request and is allowed in any 2*xx* class response to an INVITE. Its presence indicates that the UAC wishes to use the session timer for this call. Unlike the SIP-Expires header, it can only contain a delta-time, which is the current time, plus the session interval from the response.

For example, if a UAS generates a 200 OK response to a INVITE that contained a Session-Expires header with a value of 90 seconds (1.5 minutes), the UAS computes the session expiration as 1.5 minutes after the time when the 200 OK response was sent. For each proxy, the session expiration is 1.5 minutes after the time when the 2*xx* was received or sent. For the UAC, the expiration time is 1.5 minutes after the receipt of the final response.

When the gateway acts as an UAS, it is responsible for refreshes. The refresh interval is a minimum of 32 seconds, or one-third the refresh interval. When the gateway act as an UAC, the refresh interval is one-half the refresh interval.

If the session is not refreshed, the minimum time to send a BYE before the session expires is 32 seconds.

The recommended value for the Session-Expires header is 90 seconds.

The syntax of the Session-Expires header is as follows:

```
Session-Expires  =  ("Session-Expires" |
"x"
) ":" delta-seconds
                          [refresher]
refresher        =  ";" "refresher" "=" "UAS"|"UAC"
```

The *refresher* parameter is optional in the initial INVITE, although the UAC can set it to UAC to indicate that it will do the refreshes. The 200 OK response must have the refresher parameter set.

Min-SE Header

Because of the processing load of INVITE requests, the proxy, UAC, and UAS can have a configured minimum timer value that they can accept. The **min-se** (SIP) command sets the minimum timer, and it is conveyed in the Min-SE header in the initial INVITE request.

When making a call, the presence of the Min-SE header informs the UAS and any proxies of the minimum value that the UAC accepts for the session timer duration, in units of delta-seconds. The default value is 90 seconds (1.5 minutes). By not reducing the session interval below the value set, the UAS and proxies prevent the UAC from having to reject a call with a 422 error. Once set, the **min-se** command value affects all calls originated by the router. If the Min-SE header is not present, the user agent accepts any value.

The syntax of the Min-SE header is:

```
Min-SE  =  "Min-SE" ":" delta-seconds
```

422 Response Message

If the value of the Session-Expires header is too small, the UAS or proxy rejects the call with a 422 *Session Timer Too Small* response message. With the 422 response message, the proxy or UAS includes a Min-SE header indicating the minimum session value it can accept. The UAC may then retry the call with a larger session timer value.

If a 422 response message is received after an INVITE request, the UAC can retry the INVITE.

Supported and Require Headers

The presence of the *timer* argument in the Supported header indicates that the user agent supports the SIP session timer. The presence of the *timer* argument in the Require header indicates that the opposite user agent must support the SIP session timer for the call to be successful.

Benefits of SIP Session Timer Support

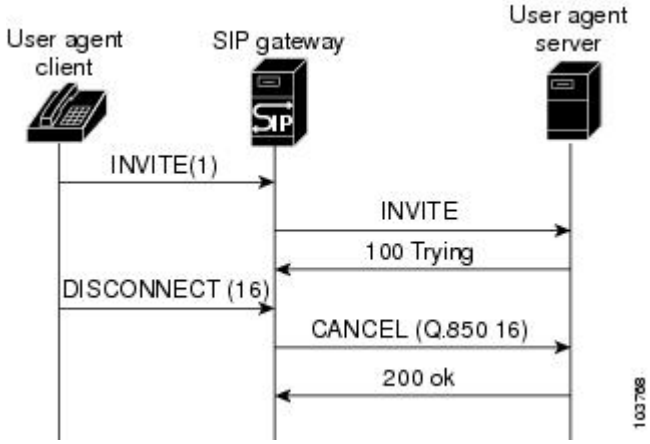
- This feature provides a periodic refresh of SIP sessions. The periodic refresh allows user agents and proxies to monitor the status of a SIP session, preventing hung network resources when network failures occur.
- Only one of the two user-agent or proxy participants in a call needs to have the SIP Session Timer Support feature implemented. This feature is easily compatible with older SIP networks.

SIP Cisco IOS Gateway Reason Header and Buffered Calling Name Completion

Reason Header

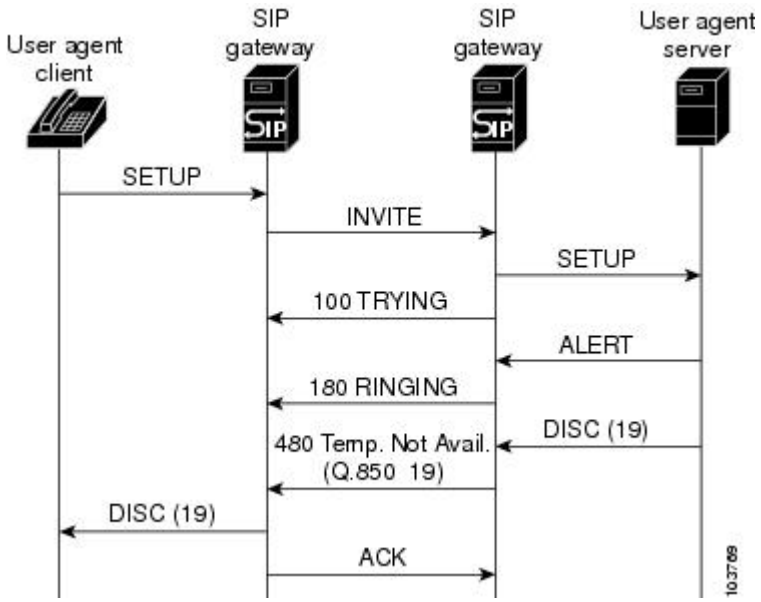
The Reason header facilitates PSTN interworking. This is achieved by having the side receiving a Disconnect message response append a Reason header to the outgoing Bye or Cancel message request and 4xx, 5xx, or 6xx message response, indicating the Q.850 cause code that passed down from the PSTN (see the figure below).

Figure 43: PSTN Interworking Using Reason Header Example



SIP implementations on PSTN gateways are plagued with issues related to mapping ISDN-disconnect message-request cause codes to SIP response status codes, which stem from the mapping on the gateway receiving the disconnect. Specifically, more than one ISDN-disconnect message-request cause code maps to one SIP status code. For example, on SIP gateways, ISDN cause codes 18, 19, and 20 all map to the SIP status code of 480 message response. This makes it impossible to deterministically relay the cause-code value on the remote end. The Reason header can carry the actual cause code (see the figure below).

Figure 44: Reason Header in Action; Extinguishing the Ambiguity in SIP Status Codes



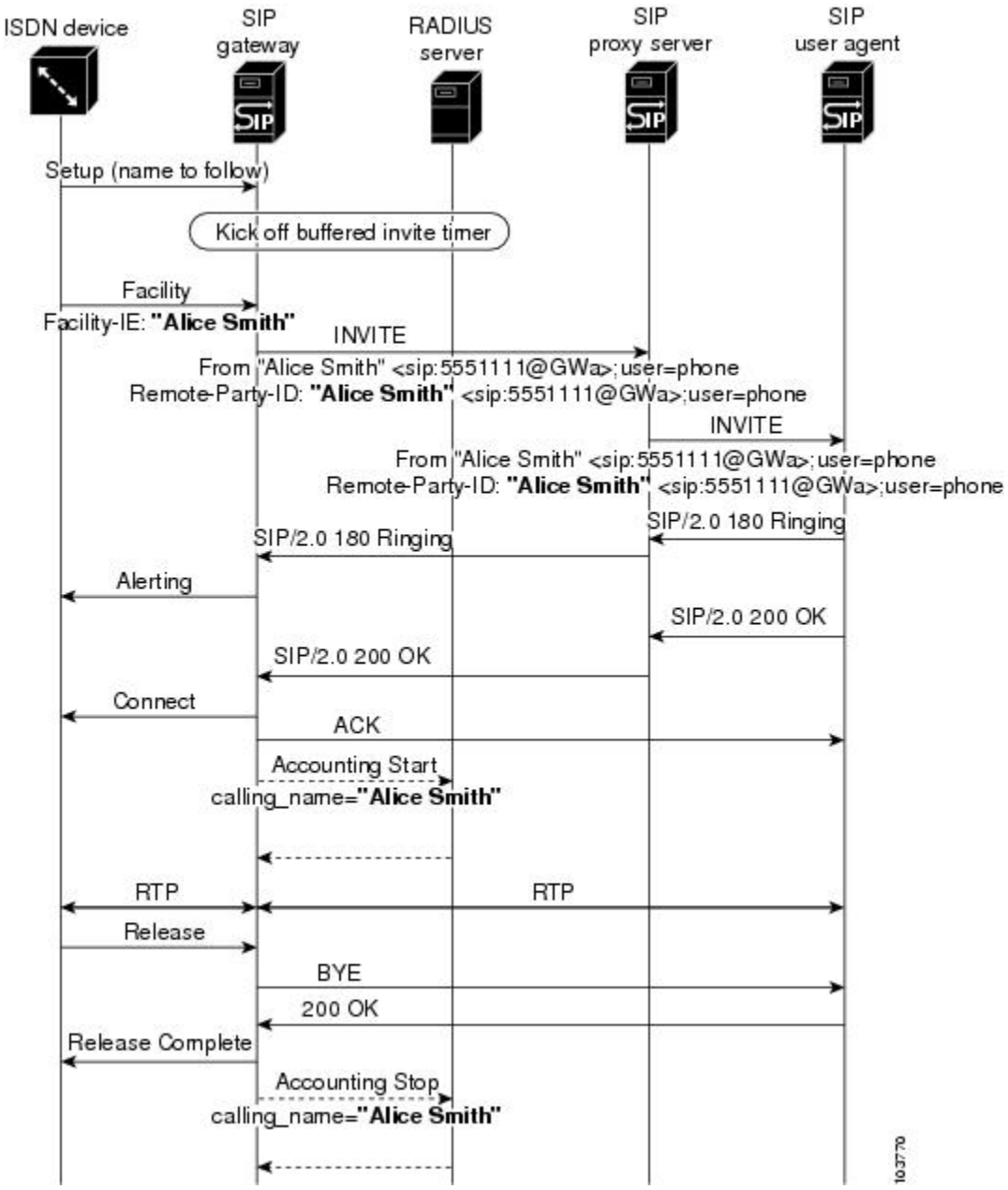
Buffered Calling-Name Completion

As shown in the figure below, Cisco IOS SIP has always supported receiving calling-name information in the display information element (IE) of a Setup message request. Support for receiving calling-name information in the facility IE of a Setup message request, of a Facility message request, and of a NOTIFY message request were supported through the Support for the ISDN Calling Name Display feature in release 12.3(4)T (refer to the “Configuring SIP DTMF Features” chapter).

The Buffered Calling Name Completion feature adds support for buffering the INVITE message request when the calling-name information is going to arrive in a subsequent facility IE of a Facility message request.

When an originating gateway (OGW) receives a Setup message with an indication that calling-name information is enabled, the configuration is checked for INVITE-message display-name buffering. When buffering is enabled, the INVITE message is buffered until the time specified in the configuration. If a Facility message with display information in the From and Remote Party ID headers of the INVITE message is received, then send it out. If no Facility message is received in the specified time, send out only the INVITE message.

Figure 45: Calling Name in Facility IE of Facility



SIP SIP Header URL Support and SUBSCRIBE NOTIFY for External Triggers

The SIP: SIP Header/URL Support and SUBSCRIBE/NOTIFY for External Triggers feature provides a mechanism for applications to send and receive SIP headers and to send SUBSCRIBE messages and receive NOTIFY events. Where appropriate, this section discusses separately the features that make up this feature set, the SIP Header Support feature along with the SUBSCRIBE and NOTIFY for External Triggers feature.

Feature benefits include the following:

- Enables the creation of presence-based, subscribe-to-be-notified services that are triggered by events external to a session
- Allows service providers to expand services to include VoiceXML-driven voice browser applications
- Allows the SIP gateway to subscribe to triggered applications and custom event-packages
- Supports distributed voice-web scenarios and call and contact center integration applications by providing access to SIP headers

Feature Design of SIP Header Support

Prior to the implementation of this feature, voice applications running on the gateway did not have access to headers sent in SIP requests. The SIP Header Passing feature makes SIP headers, the fields which specify session details in SIP messages, available to applications. This feature supports the following capabilities for VoiceXML and Tcl IVR 2.0 applications:

- Set SIP headers for outgoing SIP INVITE messages.
- Obtain information about SIP headers for incoming calls and create session variables to access the headers in VoiceXML document or Tcl IVR 2.0 script.
- Set and obtain extended and non-standard headers (user-defined header attribute-value pairs)

Using headers in SIP INVITE messages, voice applications can pass information about a call to an application on another server. For example, if the caller has entered an account number and the application transfers the call to another application on another platform, the account number can be passed in a SIP Header. An example scenario is an airline application transferring the call to a hotel reservation application hosted at a different service provider. This feature enables the respective sites to share context information about the caller.

This feature introduces a new command, the **header-passing** command, to either enable or disable passing headers from INVITE messages to applications.

The SIP Header Passing feature also provides enhanced inbound and outbound dial-peer matching services.

Feature Design of SIP SUBSCRIBE and NOTIFY for External Triggers

This feature implements support for two SIP methods, SUBSCRIBE and NOTIFY, and for a new Event header, as defined in the IETF draft, draft-roach-sip-subscribe-notify-02.txt, Event Notification in SIP.

Overview of the Application

The SIP event notification mechanism uses the SUBSCRIBE method to request notification of an event at a later time. The NOTIFY method provides notification that an event which has been requested by an earlier SUBSCRIBE method has occurred, or provides further details about the event. The new feature makes headers in incoming SIP INVITE, SUBSCRIBE, and NOTIFY messages available to applications for use in event

subscription. Similarly, to allow an application to place an outbound call using SIP, this feature passes headers in the URL for use by the SIP service provider interface (SPI) to create an outgoing INVITE request.

The new feature also supports the capability to subscribe to standard event packages, such as Message Waiting Indicator and Presence, and to application-specific custom event packages, as defined in SIP-Specific Event Notification, an earlier draft of RFC 3265, *Session Initiation Protocol (SIP)-Specific Event Notification.*

For information on these capabilities, see the following:

- "Cisco IOS Tcl IVR and VoiceXML Application Guide".
- "Cisco VoiceXML Programmer's Guide".
- "Tcl IVR API Version 2.0 Programming Guide".

Cisco implements the SUBSCRIBE and NOTIFY for External Triggers feature using the Application SUBSCRIBE/NOTIFY Layer (ASNL). ASNL is a software interface layer between the application and signaling protocol stacks that allows the application to subscribe to interested events and to pass notification when it is received.

The SUBSCRIBE and NOTIFY for External Triggers allows external SIP servers to trigger a particular voice application, behavior or activity on Cisco voice gateways. For example, a client application on the gateway subscribes to a particular event in a server. When the event takes place, the server notifies the client of that event. On receiving this event notification, the client application triggers a particular action in the gateway. The client and server must mutually agree on the events they can handle and the processing of those events.

Example of a Application

The SUBSCRIBE and NOTIFY for External Triggers feature supports various applications of external triggers. In the following scenario, a user requests a stock reminder service, for example "Let me know if Stock X reaches 100. Here is a phone number to reach me." The SUBSCRIBE and NOTIFY for External Triggers feature supports an application like this in the following manner:

- The user dials into the gateway.
- The gateway sends a subscription request to the server on the user's behalf. The subscription request contains details of the event: event name, expiration time, and other information related to the event. The request can contain any application specific headers and content.
- When the server determines, through some other means, that Stock X has reached 100, it sends a notification request to the client. The SIP NOTIFY request from the server can contain any application specific headers and content.
- This notification request triggers the client on the gateway to call the specified user or destination.

Other external trigger applications include mid-call triggers such as call center queuing and subscription to a wake-up call service.

RFC 3265 Compliance for the Feature

The Cisco implementation of SIP SUBSCRIBE and NOTIFY methods is based on an earlier draft of SIP-Specific Event Notification, and deviates from *RFC 3265, Session Initiation Protocol (SIP)-Specific Event Notification* in the following capabilities:

- The Cisco client does not support the following:
    - Embedded parameters in event package names.

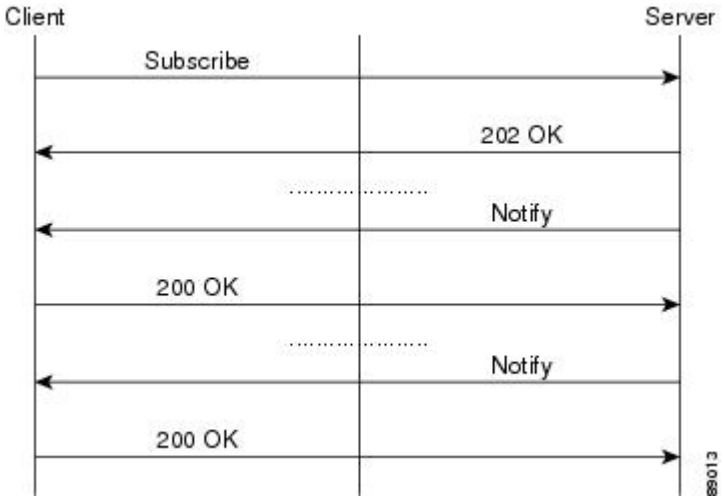
- Subscription-State header.To terminate a subscription, the notifier or user agent sends a NOTIFY request to the Cisco gateway with the Expires header set to zero.
- Forking.
- State deltas.

- In the Cisco SIP implementation, a subscription request always creates a new dialog, and cannot send a SUBSCRIBE request for an existing dialog.
- The Cisco SIP implementation does not prevent man-in-the-middle attacks as defined in RFC 3265.
- Event package registration with the IANA is not required; instead you have the flexibility to specify your own event package.

SUBSCRIBE and NOTIFY Message Flow

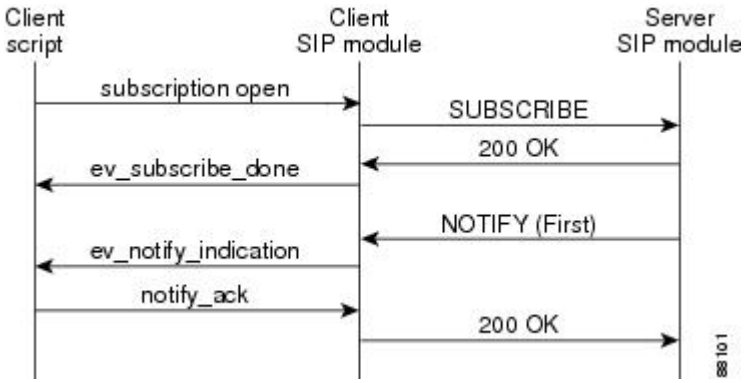
The figure below shows a typical message flow for SUBSCRIBE and NOTIFY messages.

Figure 46: SUBSCRIBE and NOTIFY Message Flow



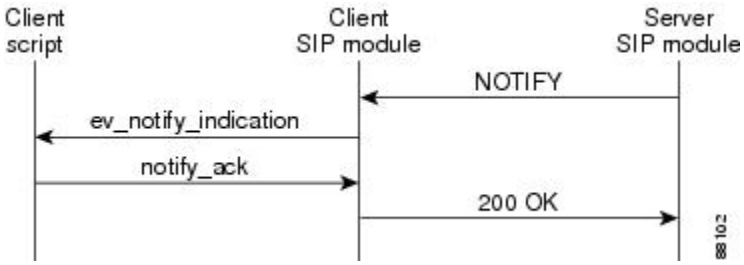
The figure below shows the message flow for a successful subscription.

Figure 47: Successful Subscription



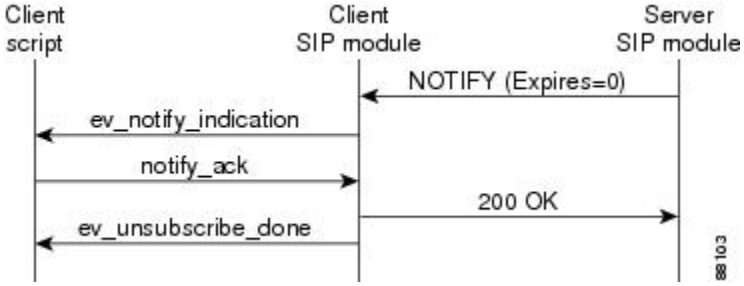
The figure below shows a completed subscription. The server can send any number of NOTIFY messages as long as the subscription is active.

Figure 48: Subscription Completed



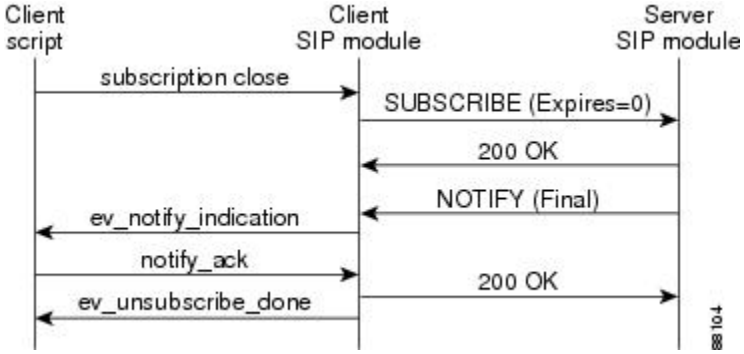
The figure below shows the message flow for subscription termination by the server.

Figure 49: Subscription Termination by the Server



The figure below shows the message flow for subscription termination by the client.

Figure 50: Subscription Termination by the Client



Sample Messages

This section presents a sequence of SIP messages sent and received between gateways during the message flow shown in "Sample Messages" in the preceding section.

Example: Subscription Request Sent From Client

This example shows a SUBSCRIBE request sent to the server. The example includes a nonstandard Subject header and an Event header.

```
*Apr 19 08:38:52.525: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
Sending MWI client request to server
*Apr 19 08:38:52.525:
*Apr 19 08:38:52.529: Sent:
```

```
SUBSCRIBE sip:user@10.7.104.88:5060 SIP/2.0
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:10.7.102.35>;tag=1C24D44-20FD
To: <sip:user@10.7.104.88>
Date: Wed, 19 Apr 2000 08:38:52 UTC
Call-ID: C4BB7610-150411D4-802186E3-AD119804
CSeq: 101 SUBSCRIBE
Timestamp: 956133532
Subject: Hi There
Contact: <sip:10.7.102.35:5060>
Event: message-summary
Expires: 500 )
Content-Type: text/plain
Content-Length: 21
This is from client
```

Example: Subscription Response Received from the Server

This example shows a response from the server to a subscription request.

```
*Apr 19 08:38:52.537: Received:
SIP/2.0 202 Accepted
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:10.7.102.35>;tag=1C24D44-20FD
To: <sip:user@10.7.104.88>;tag=1D80E90-2072
Date: Sun, 17 Nov 2002 02:59:19 GMT
Call-ID: C4BB7610-150411D4-802186E3-AD119804
Server: Cisco-SIPGateway/IOS-12.x
Timestamp: 956133532
Content-Length: 0
CSeq: 101 SUBSCRIBE
Expires: 500
*Apr 19 08:38:52.541: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: ***** act_Subscribe : SUBSCRIPTION
 DONE received
*Apr 19 08:38:52.541: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: *** act_subscribe: subscription
status=sn_000
```

Example: NOTIFY Request from the Server

This example shows the initial NOTIFY request from a server and includes an application-specific nonstandard Hello header.

```
*Apr 19 08:38:52.545: Received:
NOTIFY sip:10.7.102.35:5060 SIP/2.0
Via: SIP/2.0/UDP  10.7.104.88:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Sun, 17 Nov 2002 02:59:19 GMT
Call-ID: C4BB7610-150411D4-802186E3-AD119804
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 1037501959
CSeq: 101 NOTIFY
Event: message-summary
Hello: Hello world
Contact: <sip:user@10.7.104.88:5060>
Content-Length: 43
Content-Type: text/plain
This is content(message body) from server
```

Example: An Application Reads Header and Body Information in a NOTIFY Request

This example shows an application accessing the From and Hello headers in the NOTIFY request.

```
*Apr 19 08:38:52.549: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: *** act_Notify : NOTIFY RECEIVED
 t
*Apr 19 08:38:52.549: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
From header is: <sip:user@10.7.104.88>;tag=1D80E90-2072
*Apr 19 08:38:52.549: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
Hello header is: Hello world
*Apr 19 08:38:52.549: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
content_type received=text/plain
*Apr 19 08:38:52.549:
*Apr 19 08:38:52.553: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
content received=This is content(message body) from server
```

Example: NOTIFY Request Sent From the Client

This example shows a NOTIFY request sent from a client.

```
*Apr 19 08:38:52.553: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Wed, 19 Apr 2000 08:38:52 UTC
Call-ID: C4BB7610-150411D4-802186E3-AD119804
CSeq: 101 NOTIFY
Timestamp: 956133532
Event: message-summary
Content-Length: 0
```

Example: The Client receives a NOTIFY Message

This example shows a NOTIFY message received by a client.

```
c5300-5#
*Apr 19 08:38:57.565: Received:
NOTIFY sip:10.7.102.35:5060 SIP/2.0
Via: SIP/2.0/UDP  10.7.104.88:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Sun, 17 Nov 2002 02:59:19 GMT
Call-ID: C4BB7610-150411D4-802186E3-AD119804
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 1037501964
CSeq: 102 NOTIFY
Event: message-summary
Hello: Hello world
Contact: <sip:user@10.7.104.88:5060>
Content-Length: 35
Content-Type: text/plain
this is just a notify from server
*Apr 19 08:38:57.569: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: *** act_Notify : NOTIFY RECEIVED
*Apr 19 08:38:57.569: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
From header is: <sip:user@10.7.104.88>;tag=1D80E90-2072
*Apr 19 08:38:57.569: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
Hello header is: Hello world
*Apr 19 08:38:57.569: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
content_type received=text/plain
```

```
*Apr 19 08:38:57.569: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
content received=this is just a notify from server
```

Example: The Client Sends a NOTIFY Message

This example shows a client sending a NOTIFY message.

```
*Apr 19 08:38:57.573: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Wed, 19 Apr 2000 08:38:57 UTC
Call-ID: C4BB7610-150411D4-802186E3-AD119804
CSeq: 102 NOTIFY
Timestamp: 956133537
Event: message-summary
Content-Length: 0
```

Example: The Client Initiates a Subscription Termination

This example shows a a client initiating a subscription termination request using the Expires header set to zero.

```
*Apr 19 08:38:57.577: Sent:
SUBSCRIBE sip:user@10.7.104.88:5060 SIP/2.0
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:10.7.102.35>;tag=1C24D44-20FD
To: <sip:user@10.7.104.88>;tag=1D80E90-2072
Date: Wed, 19 Apr 2000 08:38:57 UTC
Call-ID: C4BB7610-150411D4-802186E3-AD119804
CSeq: 102 SUBSCRIBE
Timestamp: 956133537
Subject: Hi There
Contact: <sip:10.7.102.35:5060>
Event: message-summary
Expires: 0
Content-Type: text/plain
Content-Length: 21
This is from client
```

Example: The Client Receives a Response to a Subscription Termination Request

This example shows a client receiving a response to a subscription termination request.

```
*Apr 19 08:38:57.589: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:10.7.102.35>;tag=1C24D44-20FD
To: <sip:user@10.7.104.88>;tag=1D80E90-2072
Date: Sun, 17 Nov 2002 02:59:24 GMT
Call-ID: C4BB7610-150411D4-802186E3-AD119804
Server: Cisco-SIPGateway/IOS-12.x
Timestamp: 956133532
Content-Length: 0
CSeq: 102 SUBSCRIBE
Expires: 0
Contact: <sip:user@10.7.104.88:5060>
```

Example: The Client Receives a Final NOTIFY Message

This example shows a client receiving a final NOTIFY message that a subscription is finished.

```
c5300-5#
*Apr 19 08:39:02.585: Received:
NOTIFY sip:10.7.102.35:5060 SIP/2.0
Via: SIP/2.0/UDP  10.7.104.88:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Sun, 17 Nov 2002 02:59:24 GMT
Call-ID: C4BB7610-150411D4-802186E3-AD119804
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 1037501969
CSeq: 103 NOTIFY
Event: message-summary
Hello: Hello world
Contact: <sip:user@10.7.104.88:5060>
Content-Length: 35
Content-Type: text/plain
this is just a notify from server
*Apr 19 08:39:02.589: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: *** act_Notify : FINAL NOTIFY
RECEIVED
*Apr 19 08:39:02.589: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: status=sn_004
*Apr 19 08:39:02.589: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd:
From header is: <sip:user@10.7.104.88>;tag=1D80E90-2072
*Apr 19 08:39:02.593: //-1//TCL2:HN01C24D3C:/tcl_PutsCmd: act_UnsubcribeDone : !!!
SUBSCRIPTION IS  OVER !!!
```

Example: A Final NOTIFY Message to a Server

This example shows a final NOTIFY message to a server.

```
*Apr 19 08:39:02.593: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP  10.7.102.35:5060
From: <sip:user@10.7.104.88>;tag=1D80E90-2072
To: <sip:10.7.102.35>;tag=1C24D44-20FD
Date: Wed, 19 Apr 2000 08:39:02 UTC
Call-ID: C4BB7610-150411D4-802186E3-AD119804
CSeq: 103 NOTIFY
Timestamp: 956133542
Event: message-summary
Content-Length: 0
```

SIP Stack Portability

The SIP Stack Portability feature implements the following capabilities to the Cisco IOS SIP gateway stack:

- It receives inbound Refer message requests both within a dialog and outside of an existing dialog from the user agents (UAs).
- It sends and receives SUBSCRIBE or NOTIFY message requests via UAs.
- It receives unsolicited NOTIFY message requests without having to subscribe to the event that was generated by the NOTIFY message request.
- It supports outbound delayed media.

It sends an INVITE message request without Session Description Protocol (SDP) and provides SDP information in either the PRACK or ACK message request for both initial call establishment and mid-call re-INVITE message requests.

- It sets SIP headers and content body in requests and responses.

The stack applies certain rules and restrictions for a subset of headers and for some content types (such as SDP) to protect the integrity of the stack's functionality and to maintain backward compatibility. When receiving SIP message requests, it reads the SIP header and any attached body without any restrictions.

To make the best use of SIP call-transfer features, you should understand the following concepts:

SIP Call-Transfer Basics

Basic Terminology of SIP Call Transfer

Call transfer allows a wide variety of decentralized multiparty call operations. These decentralized call operations form the basis for third-party call control, and thus are important features for VoIP and SIP. Call transfer is also critical for conference calling, where calls can transition smoothly between multiple point-to-point links and IP-level multicasting.

Refer Message Request

The SIP Refer message request provides call-transfer capabilities to supplement the SIP BYE and ALSO message requests already implemented on Cisco IOS SIP gateways. The Refer message request has three main roles:

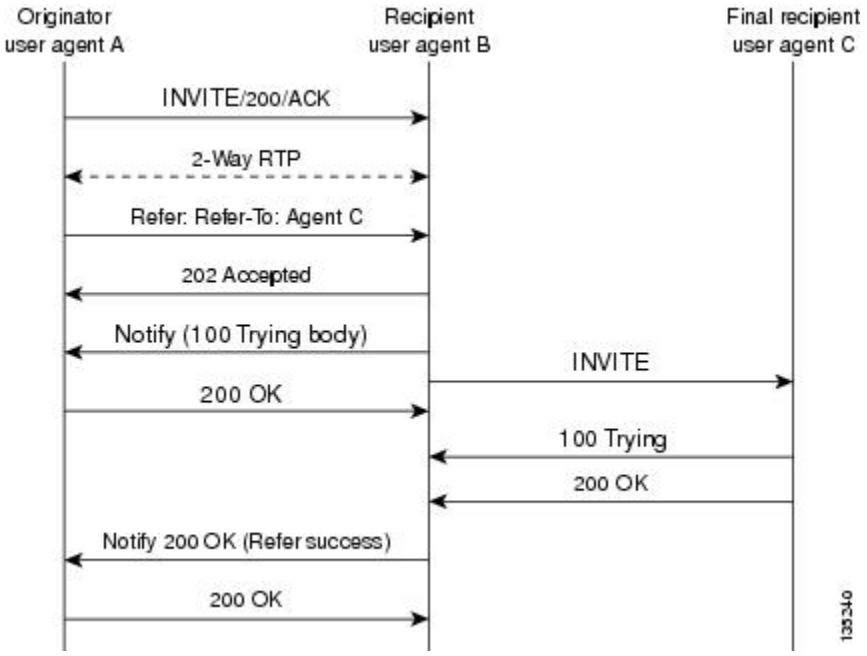
- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

Note A gateway can be a recipient or final recipient, but not an originator.

The Refer message request always begins within the context of an existing call and starts with the *originator* . The originator sends a Refer request to the *recipient* (user agent receiving the Refer request) to initiate a triggered INVITE request. The triggered INVITE request uses the SIP URL contained in the Refer-To header as the destination of the INVITE request. The recipient then contacts the resource in the Refer-To header (*final recipient*), and returns a SIP 202 (Accepted) response to the originator. The recipient also must notify the originator of the outcome of the Refer transaction--whether the final recipient was successfully contacted or not. The notification is accomplished using the SIP NOTIFY message request, SIP's event notification mechanism. A NOTIFY message with a message body of SIP 200 OK indicates a successful transfer, and a message body of SIP 503 Service Unavailable indicates an unsuccessful transfer. If the call was successful, a call between the recipient and the final recipient results.

The figure below represents the call flow of a successful Refer transaction initiated within the context of an existing call.

Figure 51: Successful Refer transaction



Refer-To Header

The recipient receives from the originator a Refer request that always contains a single Refer-To header. The Refer-To header includes a SIP URL that indicates the party to be invited and must be in SIP URL format.

Note The TEL URL format cannot be used in a Refer-To header, because it does not provide a host portion, and without one, the triggered INVITE request cannot be routed.

The Refer-To header may contain three additional overloaded headers to form the triggered INVITE request. If any of these three headers are present, they are included in the triggered INVITE request. The three headers are:

- Accept-Contact--Optional in a Refer request. A SIP Cisco IOS gateway that receives an INVITE request with an Accept-Contact does not act upon this header. This header is defined in draft-ietf-sip-callerprefs-03.txt and may be used by user agents that support caller preferences.
- Proxy-Authorization--Nonstandard header that SIP gateways do not act on. It is echoed in the triggered INVITE request because proxies occasionally require it for billing purposes.
- Replaces--Header used by SIP gateways to indicate whether the originator of the Refer request is requesting a blind or attended transfer. It is required if the originator is performing an attended transfer, and not required for a blind transfer.

All other headers present in the Refer-To are ignored, and are not sent in the triggered INVITE.

Note The Refer-To and Contact headers are required in the Refer request. The absence of these headers results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4*xx* class response.

Referred-By Header

The Referred-By header is required in a Refer request. It identifies the originator and may also contain a signature (included for security purposes). SIP gateways echo the contents of the Referred-By header in the triggered INVITE request, but on receiving an INVITE request with this header, gateways do not act on it.

Note The Referred-By header is required in a Refer request. The absence of this header results in a 4*xx* class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4*xx* class response.

NOTIFY Message Request

Once the outcome of the Refer transaction is known, the recipient of the Refer request must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully contacted or not. The notification is accomplished using the NOTIFY message request, SIP's event notification mechanism. The notification contains a message body with a SIP response status line and the response class in the status line indicates the success or failure of the Refer transaction.

The NOTIFY message must do the following:

- Reflect the same To, From, and Call-ID headers that were received in the Refer request.
- Contain an Event header refer.
- Contain a message body with a SIP response line. For example: SIP/2.0 200 OK to report a successful Refer transaction, or SIP/2.0 503 Service Unavailable to report a failure. To report that the recipient disconnected before the transfer finished, it must use SIP/2.0 487 Request Canceled.

Two Cisco IOS commands pertain to the NOTIFY message request:

- The **timers notify** command sets the amount of time that the recipient should wait before retransmitting a NOTIFY message to the originator.
- The **retry notify** command configures the number of times a NOTIFY message is retransmitted to the originator.

Note For information on these commands, see the *Cisco IOS Voice Command Reference*.

Types of SIP Call Transfer Using the Refer Message Request

This section discusses how the Refer message request facilitates call transfer.

There are two types of call transfer: blind and attended. The primary difference between the two is that the Replaces header is used in attended call transfers. The Replaces header is interpreted by the final recipient and contains a Call-ID header, indicating that the initial call leg is to be replaced with the incoming INVITE request.

As outlined in the Refer message request, there are three main roles:

- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

A gateway can be a recipient or final recipient, but not an originator.

Blind Call-Transfer Process

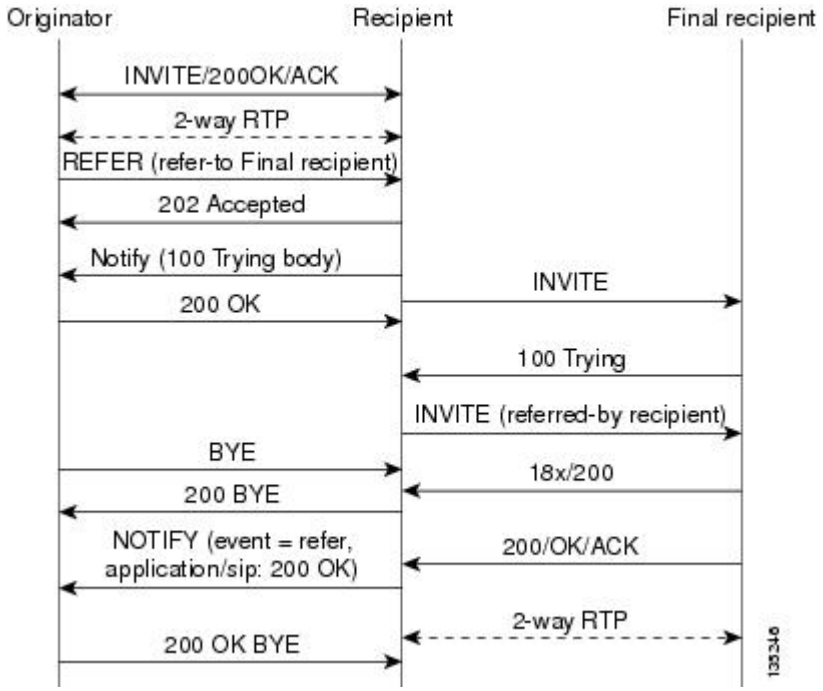
A blind, or unattended, transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. This is different from a consultative, or attended, transfer in which one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller to the third party. Blind transfers are often preferred by automated devices that do not have the capability to make consultation calls.

Blind transfer works as described in the Refer Message Request section. The process is as follows:

1. Originator (user agent that initiates the transfer or Refer request) does the following:
    1. Sets up a call with recipient (user agent that receives the Refer request)
    2. Issues a Refer request to recipient
2. Recipient does the following:
    1. Sends an INVITE request to final recipient (user agent introduced into a call with the recipient)
    2. Returns a SIP 202 (Accepted) response to originator
    3. Notifies originator of the outcome of the Refer transaction--whether final recipient was successfully (SIP 200 OK) contacted or not (SIP 503 Service Unavailable)
3. If successful, a call is established between recipient and final recipient.
4. The original signaling relationship between originator and recipient terminates when either of the following occurs:
5. One of the parties sends a Bye request.
6. Recipient sends a Bye request after successful transfer (if originator does not first send a Bye request after receiving an acknowledgment for the NOTIFY message).

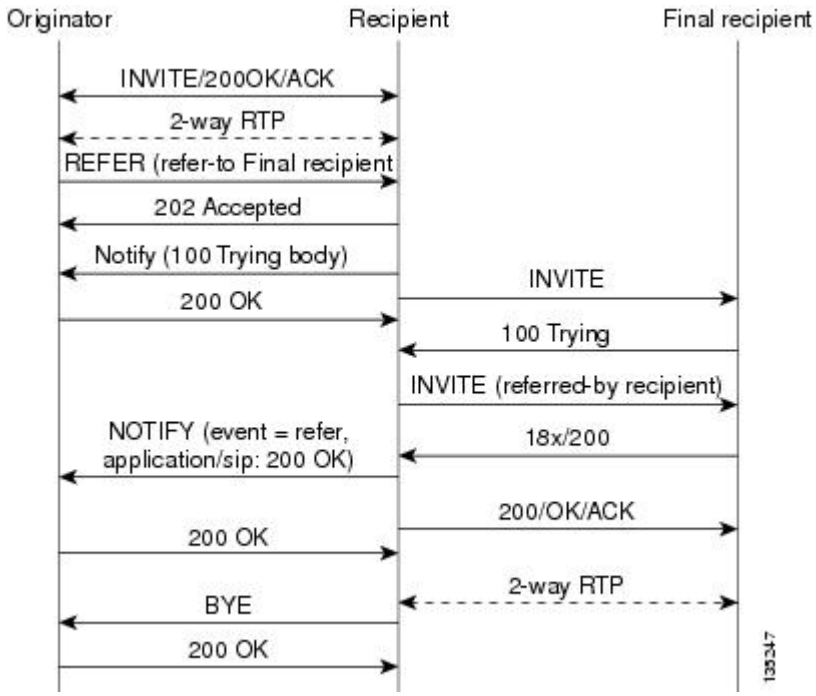
The figure below shows a successful blind or unattended call transfer in which the originator initiates a Bye request to terminate signaling with the recipient.

Figure 52: Successful Blind or Unattended Transfer--Originator Initiating a Bye Request



The figure below shows a successful blind or unattended call transfer in which the recipient initiates a Bye request to terminate signaling with the originator. A NOTIFY message is always sent by the recipient to the originator after the final outcome of the call is known.

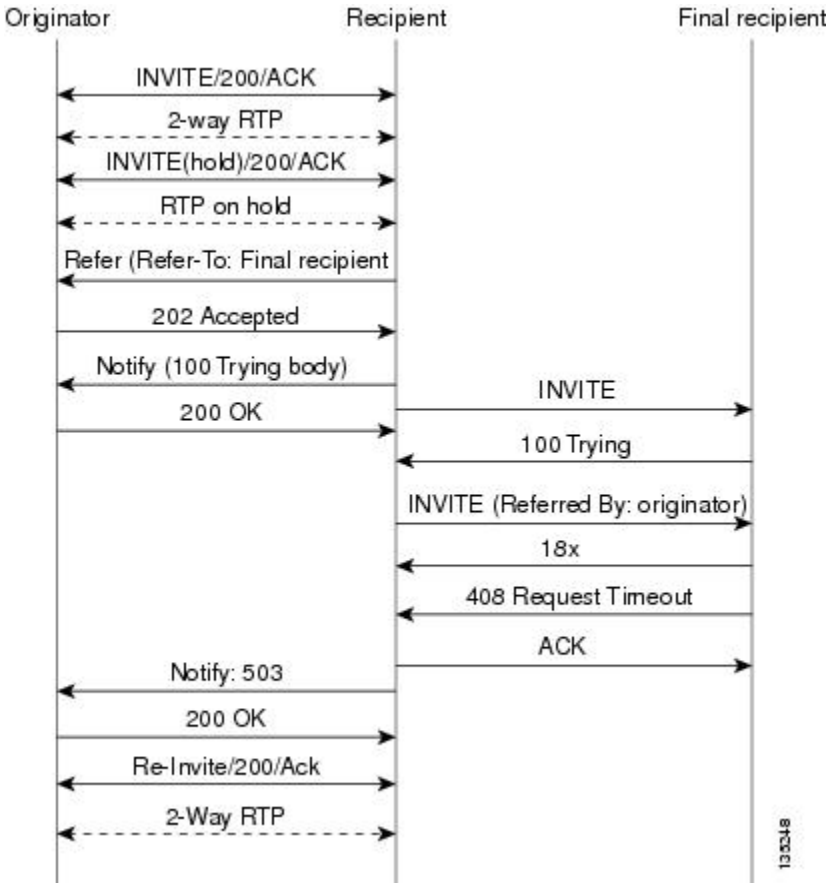
Figure 53: Successful Blind or Unattended Transfer--Recipient Initiating a Bye Request



If a failure occurs with the triggered INVITE to the final recipient, the call between originator and recipient is not disconnected. Rather, with blind transfer the process is as follows:

1. Originator sends a re-INVITE that takes the call off hold and returns to the original call with recipient.
2. Final recipient sends an 18x informational response to recipient.
3. The call fails; the originator cannot recover the call with recipient. Failure can be caused by an error condition or timeout.
4. The call leg between originator and recipient remains active (see the figure below).
5. If the INVITE to final recipient fails (408 Request Timeout), the following occurs:
    1. Recipient notifies originator of the failure with a NOTIFY message.
    2. Originator sends a re-INVITE and returns to the original call with the recipient.

Figure 54: Failed Blind Transfer--Originator Returns to Original Call with Recipient



Attended Transfer

In attended transfers, the Replaces header is inserted by the initiator of the Refer message request as an overloaded header in the Refer-To and is copied into the triggered INVITE request sent to the final recipient. The header has no effect on the recipient, but is interpreted by the final recipient as a way to distinguish between blind transfer and attended transfer. The attended transfer process is as follows:

1. Originator does the following:
    1. Sets up a call with recipient.
    2. Places recipient on hold.
    3. Establishes a call to final recipient.
    4. Sends recipient a Refer message request with an overloaded Replaces header in the Refer-To header.
2. Recipient does the following:
    1. Sends a triggered INVITE request to final recipient. (Request includes the Replaces header, identifying the call leg between the originator and the final recipient.)
    2. Recipient returns a SIP 202 (Accepted) response to originator. (Response acknowledges that the INVITE has been sent.)
3. Final recipient establishes a direct signaling relationship with recipient. (Replaces header indicates that the initial call leg is to be shut down and replaced by the incoming INVITE request.)
4. Recipient notifies originator of the outcome of the Refer transaction. (Outcome indicates whether or not the final recipient was successfully contacted.)
5. Recipient terminates the session with originator by sending a Bye request.

Replaces Header

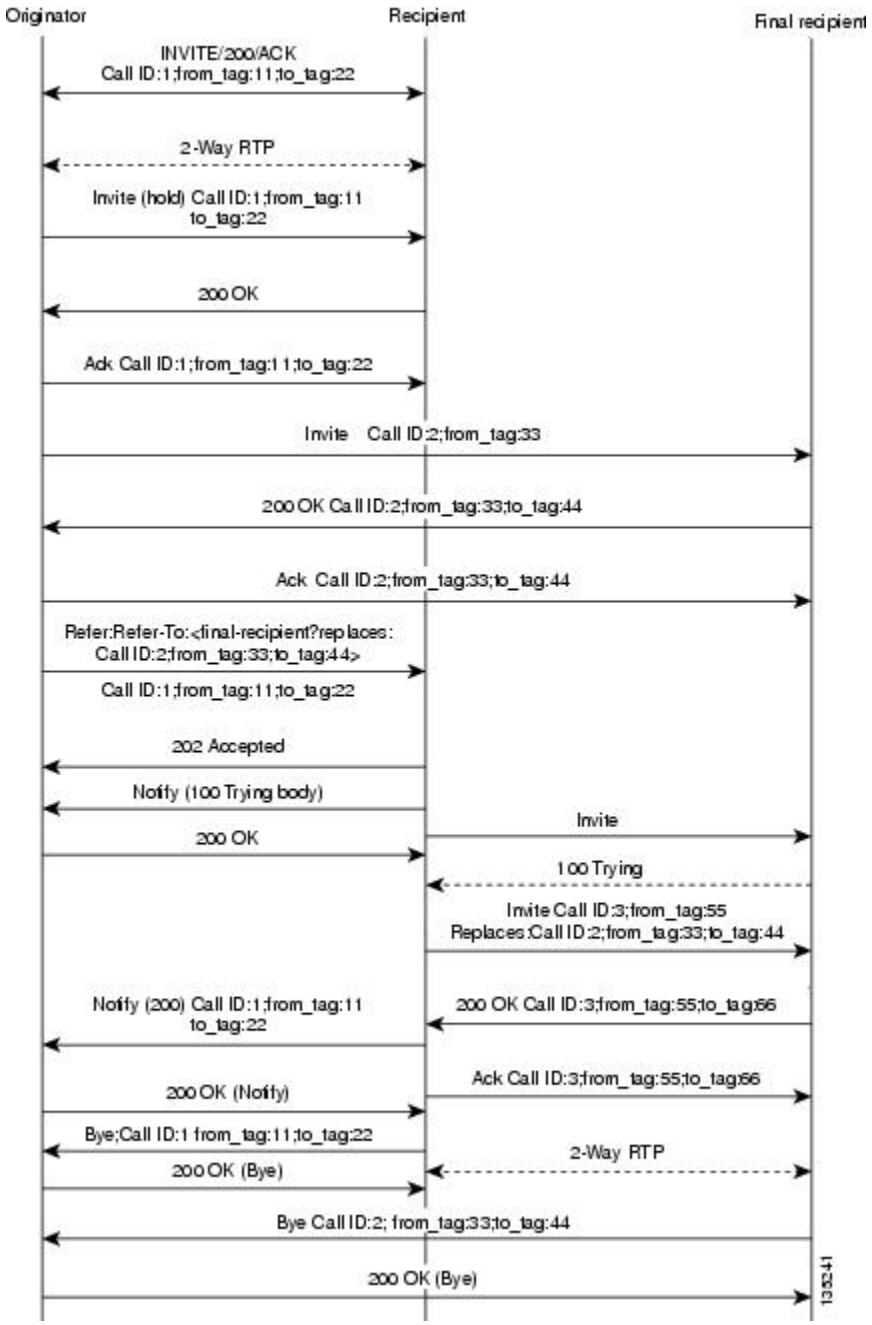
The Replaces header is required in attended transfers. It indicates to the final recipient that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming INVITE request. The final recipient sends a Bye request to the originator to terminate its session.

If the information provided by the Replaces header does not match an existing call leg, or if the information provided by the Replaces header matches a call leg but the call leg is not active (a Connect, 200 OK to the INVITE request has not been sent by the final-recipient), the triggered INVITE does not replace the initial call leg and the triggered INVITE request is processed normally.

Any failure resulting from the triggered INVITE request from the recipient to the final recipient does not drop the call between the originator and the final recipient. In these scenarios, all calls that are active (originator to recipient and originator to final recipient) remain active after the failed attended transfer attempt

The figure below shows a call flow for a successful attended transfer.

Figure 55: Successful Attended Transfer



Attended Transfer with Early Completion

Attended transfers allow the originator to have a call established between both the recipient and the final recipient. With attended transfer with early completion, the call between the originator and the final recipient does not have to be active, or in the talking state, before the originator can transfer it to the recipient. The originator establishes a call with the recipient and only needs to be setting up a call with the final recipient.

The final recipient may be ringing, but has not answered the call from the originator when it receives a re-INVITE to replace the call with the originator and the recipient.

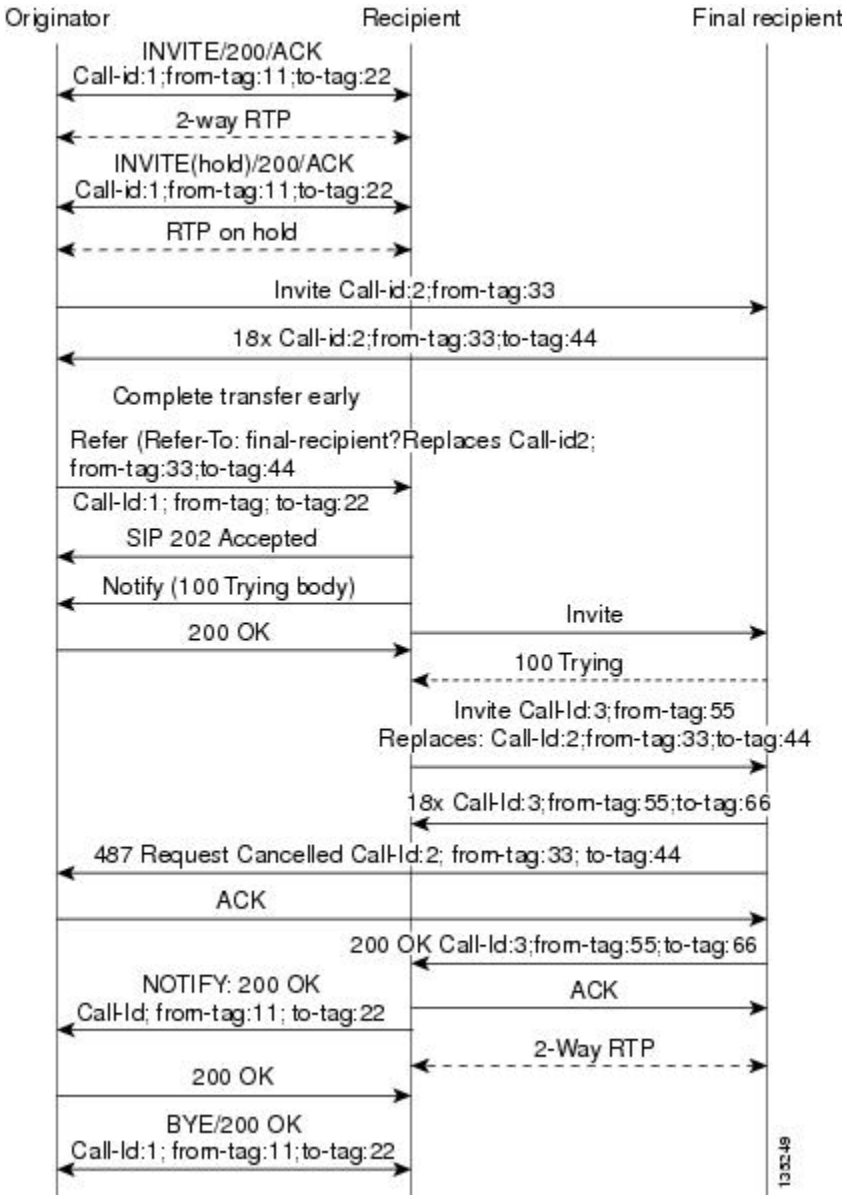
The process for attended transfer with early completion is as follows (see the figure below):

1. Originator does the following:
    1. Sets up a call with recipient.
    2. Places the recipient on hold.
    3. Contacts the final recipient.
    4. After receiving an indication that the final recipient is ringing, sends recipient a Refer message request with an overloaded Replaces header in the Refer-To header. (The Replaces header is required in attended transfers and distinguishes between blind transfer and attended transfers.)
2. Recipient does the following:
    1. Returns a SIP 202 (Accepted) response to the originator. (to acknowledge that the INVITE has been sent.)
    2. Upon receipt of the Refer message request, sends a triggered INVITE request to final recipient. (The request includes the Replaces header, which indicates that the initial call leg, as identified by the Call-ID header and tags, is to be shut down and replaced by the incoming INVITE request.)
3. Final recipient establishes a direct signaling relationship with recipient.
4. Final recipient tries to match the Call-ID header and the To or From tag in the Replaces header of the incoming INVITE with an active call leg in its call control block. If a matching active call leg is found, final recipient replies with the same status as the found call leg. However, it then terminates the found call leg with a 487 Request Cancelled response.

Note If early transfer is attempted and the call involves quality of service (QoS) or Resource Reservation Protocol (RSVP), the triggered INVITE from the recipient with the Replaces header is not processed and the transfer fails. The session between originator and final recipient remains unchanged.

1. Recipient notifies originator of the outcome of the Refer transaction--that is, whether final recipient was successfully contacted or not.
2. Recipient or originator terminates the session by sending a Bye request.

Figure 56: Attended Transfer with Early Completion



VSA for Call Transfer

You can use a vendor-specific attribute (VSA) for SIP call transfer.

Referred-By Header

For consistency with existing billing models, Referred-By and Requested-By headers are populated in call history tables as a VSA. Cisco VSAs are used for VoIP call authorization. The new VSA tag **supp-svc-xfer-by**helps to associate the call legs for call-detail-record (CDR) generation. The call legs can be originator-to-recipient or recipient-to-final-recipient.

The VSA tag **supp-svc-xfer-by** contains the user@host portion of the SIP URL of the Referred-By header for transfers performed with the Refer message request. For transfers performed with the Bye/Also message request, the tag contains user@host portion of the SIP URL of the Requested-By header. For each call on the gateway, two RADIUS records are generated: start and stop. The **supp-svc-xfer-by**VSA is generated only for stop records and is generated only on the recipient gateway--the gateway receiving the Refer or Bye/Also message.

The VSA is generated when a gateway that acts as a recipient receives a Refer or Bye/Also message with the Referred-By or Requested-By headers. There are usually two pairs of start and stop records. There is a start and stop record between the recipient and the originator and also between the recipient to final recipient. In the latter case, the VSA is generated between the recipient to the final recipient only.

Business Group Field

A new business group VSA field has been added that assists service providers with billing. The field allows service providers to add a proprietary header to call records. The VSA tag for business group ID is **cust-biz-grp-id** and is generated only for stop records. It is generated when the gateway receives an initial INVITE with a vendor dial-plan header to be used in call records. In cases when the gateway acts as a recipient, the VSA is populated in the stop records between the recipient and originator and the final recipient.

Note For information on VSAs, see the *RADIUS VSA Voice Implementation Guide*.

SIP Call Transfer and Call Forwarding Using Tcl IVR 2.0 and VoiceXML Applications

SIP Call Transfer and Call Forwarding with a Tcl IVR Script

When using a Tcl IVR 2.0 application, you can implement SIP support of blind, or attended, call-transfer and call-forwarding requests from a Cisco IOS gateway. A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. This is different from a consultative transfer in which one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller to the third party. Blind transfers are often preferred by automated devices that do not have the capability to make consultation calls.

Before implementing blind transfer and call forwarding, you must write a custom Tcl IVR 2.0 script that implements call transfer and call forwarding. The script is responsible for receiving the hookflash event, providing dial tone, matching against the dial plan, initiating call transfer, and reestablishing the original call if the transfer attempt fails.

Note For information on writing a Tcl IVR script, see "Tcl IVR API Version 2.0 Programming Guide".

When the Tcl IVR script runs on the Cisco gateway, it can respond to requests to initiate blind call transfer (transfer without consultation) on a SIP call leg. SIP call forwarding on ephones (IP phones that are not configured on the gateway) is also supported.

Note SIP call transfer and call forwarding are compliant with VoiceXML. VoiceXML scripts can also be used to implement call transfer and call forwarding.

Release Link Trunking on SIP Gateways

Release link trunking (RLT) functionality has been added to Cisco IOS SIP gateways. With RLT functionality, SIP call transfer now can be triggered by channel associated signaling (CAS) trunk signaling, which the custom Tcl IVR application can monitor. After a SIP call transfer has transpired and the CAS interface is no longer required, the CAS interface can be released.

The RLT functionality can be used to initiate blind transfers on SIP gateways. Blind call transfer uses the Refer message request. A full description of blind transfer and the Refer message request can be found in the "Configuring SIP Call-Transfer Features" chapter.

RLT and SIP Call Transfers

Call transfer can be triggered by CAS trunk signaling and then captured by the Tcl IVR script on a gateway. The process begins with the originator (the SIP user agent that initiates the transfer or Refer message request) responding with a dial tone once the originator receives the signal or hookflash from the PSTN call leg. The originator then prepares to receive dual-tone multifrequency (DTMF) digits that identify the final recipient (the user agent introduced into a call with the recipient).

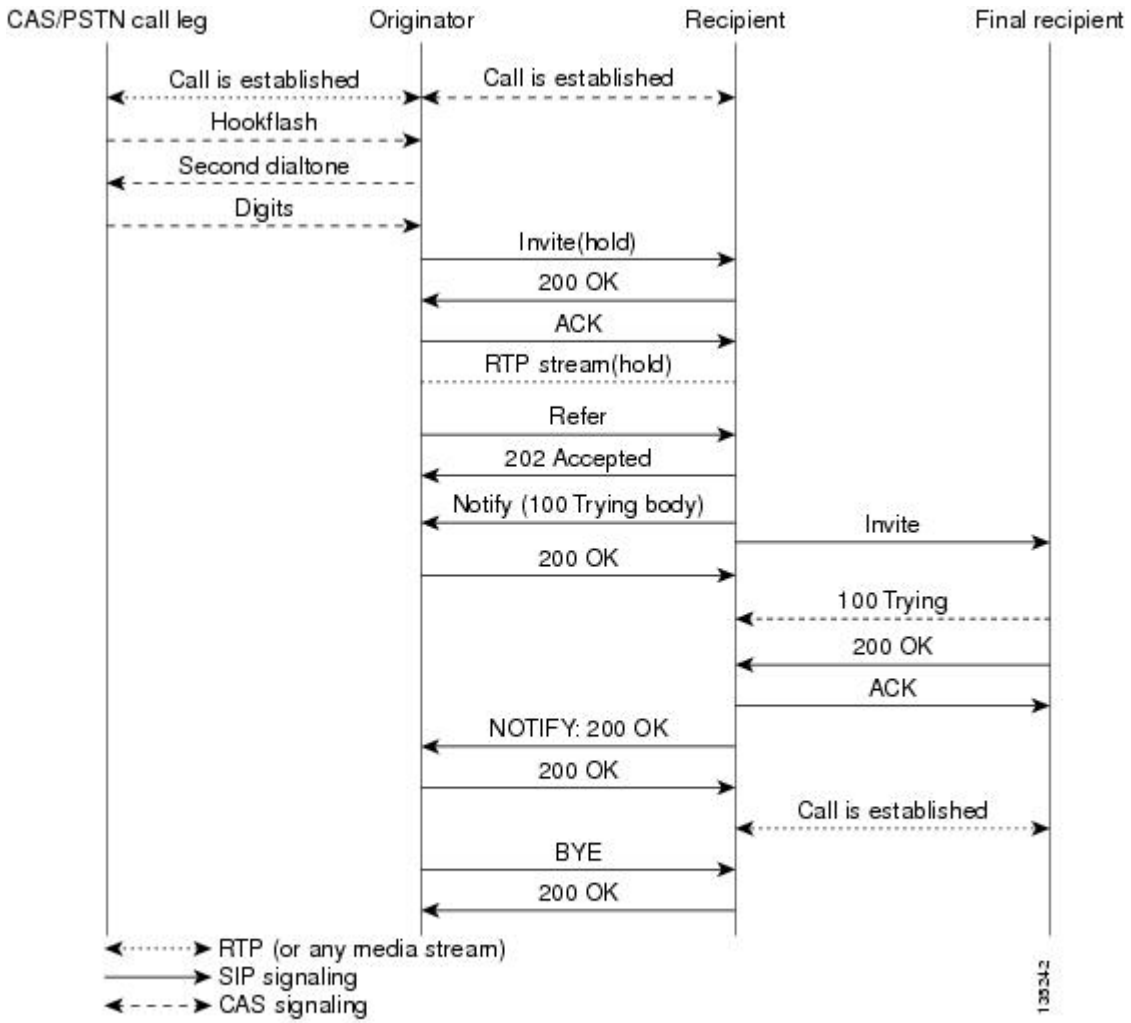
Once the first DTMF digit is received, the dial tone is discontinued. DTMF-digit collection is not completed until a 4-second interdigit timeout occurs, or an on-hook is received on that specific CAS time slot. Call transfer starts when DTMF-digit collection is successful. If digit collection fails, for example, if not enough DTMF digits or invalid digits are collected, the initial call is reestablished.

Once the DTMF digits are successfully collected, the Tcl IVR script can initiate call transfer. SIP messaging begins when the transfer is initiated with the Refer message request. The originator sends an INVITE to the recipient (the user agent that receives the Refer message request and is transferred to the final recipient) to hold the call and request that the recipient not return Real-Time Transport Protocol (RTP) packets to the originator. The originator then sends a SIP Refer message request to the recipient to start the transfer process. When the recipient receives the request, the recipient returns a 202 *Accepted* acknowledgment to the originator. The Tcl IVR script run by the originator can then release the CAS trunk and close the primary call. See the figure below.

If the recipient does not support the Refer message request, a 501 *Not implemented* message is returned. However, for backward compatibility purposes, the call transfer is automatically continued with the Bye/Also message request. The originator sends a Bye/Also request to the recipient and releases the CAS trunk with the PSTN call leg. The primary call between the originator and the recipient is closed when a 200 OK response is received.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down.

Figure 57: Call Transfer Using the Refer Message Request



SIP and TEL URLs in Call Transfers

When the SIP call-transfer originator collects DTMF digits from the CAS trunk, it attempts to find a dial peer. If a dial peer is found, the session target in the dial peer is used to formulate a SIP URL. This URL can be used with both the Refer message request and the Bye/Also message request. A SIP URL is in the following form:

```
sip:JohnSmith@example.com
```

If a valid dial peer is not found, a Telephone Uniform Resource Locator (TEL URL) is formulated in the Refer-To header. A TEL URL is in the following form:

```
tel:+11235550100
```

The choice of which URL to use is critical when correctly routing SIP calls. For example, the originating gateway can send out a Bye with an Also header, but the Also header can carry only a SIP URL. The Also header cannot carry a TEL URL. That is, if the gateway decides to send a Bye/Also but cannot find a matched dial peer, the gateway reports an error on the transfer gateway and sends a Bye without the Also header.

If the recipient of a SIP call transfer is a SIP phone, the phone must have the capability to interpret either the Refer message request or the Bye/Also message request for the call transfer to work. If the recipient is a Cisco IOS gateway, there needs to be a matching dial peer for the Refer-To *user* . *User,* looking at the previous example, can be either *JohnSmith* or *11235550100* . The dial peer also needs to have an application session defined, where session can be the name of a Tcl IVR application. If there is no match, a 4*xx* error is sent back and no transfer occurs. If there is a POTS dial-peer match, a call is made to that POTS phone. Before Cisco IOS Release 12.2(15)T, if there is a VoIP match, the Refer-To URL is used to initiate a SIP call. In Release 12.2(15)T and later releases, the application session target in the dial peer is used for the SIP call.

SIP Gateway Initiation of Call Transfers

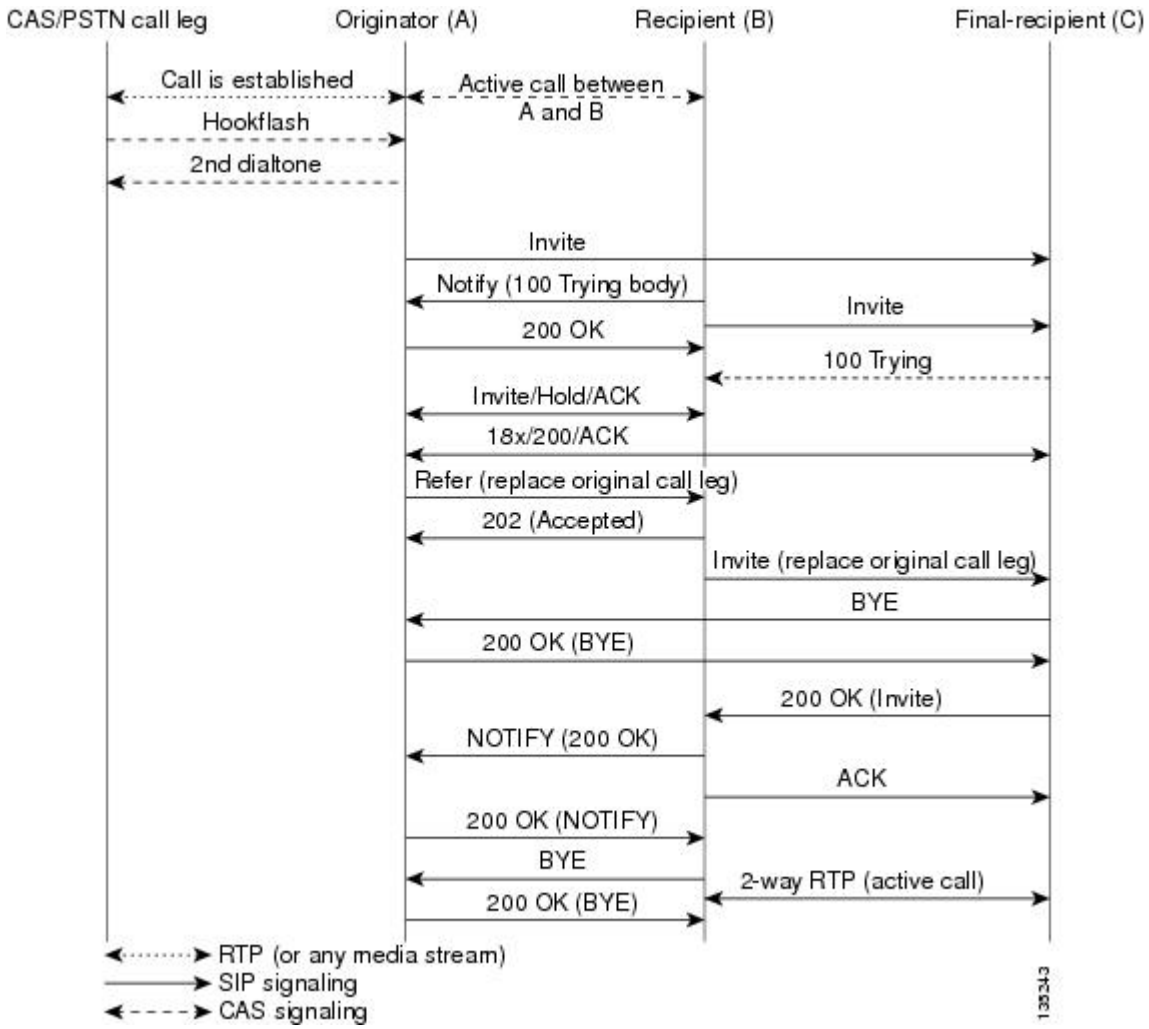
SIP gateways can also initiate, or originate, attended call transfers. The process begins when the originator establishes a call with the recipient. When the user on the PSTN call leg wants to transfer the call, the user uses hookflash to get a second dial tone and then enters the final recipient's number. The Tcl IVR script can then put the original call on hold and set up the call to the final-recipient, making the originator active with the final-recipient. The Refer message request is sent out when the user hangs up to transfer the call. The Refer message request contains a Replaces header that contains three tags: *SIP CallID* , *from* , and *to* . The tags are passed along in the INVITE from the recipient to the final recipient, giving the final recipient adequate information to replace the call leg. The host portion of the Refer message request is built from the established initial call. The following is an example of a Refer message request that contains a Replaces header:

Note IP addresses and host names in examples are fictitious.

```
Refer sip:5550100@172.16.190.100:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.16.190.99:5060
From: "5550101" <sip:5555555@172.16.190.187>
To: <sip:5550100@172.16.190.187>;tag=A7C2C-1E8C
Date: Sat, 01 Jan 2000 05:15:06 GMT
Call-ID: c2943000-106ae5-1c5f-3428@172.16.197.182
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 6
Timestamp: 946685709
CSeq: 103 Refer
Refer-To:
sip:5550101@10.102.17.217?Replaces=DD713380-339C11CC-80BCF308-92BA812C@172.16.195.77;to-tag=A5438-23E4;from-tag=C9122EDB-2408
Referred-By: <sip:5550101@172.16.190.99>
Content-Length: 0
```

Once the NOTIFY is received by the originator, the Tcl IVR script can disconnect the call between the originator and the recipient. The call between the originator and the final recipient is disconnected by the recipient sending a BYE to the originator. See the figure below for a call flow of a successful call transfer.

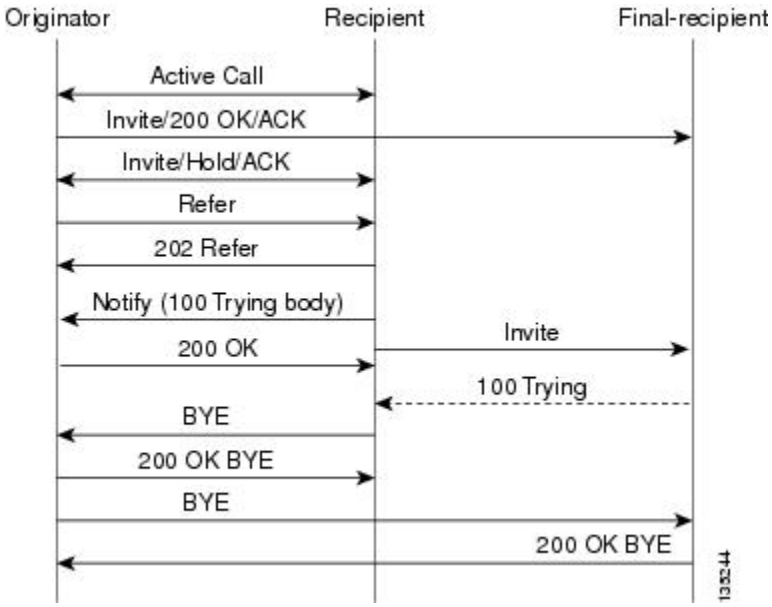
Figure 58: Successful Attended Call Transfer Initiated by the Originator



If the recipient does not support the Refer message request, a 501 *Not implemented* message is returned.

In all other cases of call-transfer failures, the primary call between the originator and the recipient is immediately shut down. The figure below shows the recipient hanging up the call before the transfer completes. The item to notice is that the NOTIFY message is never sent.

Figure 59: Unsuccessful Call Transfer--Recipient Hangs Up Before Transfer Completes



SIP Call Forwarding

SIP call forwarding is supported only on ephones--IP phones that are not configured on the gateway. Foreign exchange station (FXS), foreign exchange office (FXO), T1, E1, and CAS phones are not supported.

With ephones, four different types of SIP call forwarding are supported:

- Call Forward Unavailable
- Call Forward No Answer
- Call Forward Busy
- Call Forward Unconditional

In all four of these call-forwarding types, a 302 *Moved Temporarily* response is sent to the user-agent client. A Diversion header included in the 302 response indicates the type of forward.

The 302 response also includes a Contact header, which is generated by the calling number that is provided by the custom Tcl IVR script. The 302 response also includes the host portion found in the dial peer for that calling number. If the calling number cannot match a VoIP dial-peer or POTS dial-peer number, a 503 *Service Unavailable* message is sent, except in the case of the Call Forward No Answer. With Call Forward No Answer, call forwarding is ignored, the phone rings, and the expires timer clears the call if there is no answer.

Note In Cisco IOS Release 12.2(15)T, when SIP with ephones is used, DTMF is not supported. Voice can be established, but DTMF cannot be relayed in- or out-of-band. Custom scripting is also necessary for ephones to initiate call forwarding.

SUBSCRIBE or NOTIFY Message Request Support

The Cisco IOS gateway accepts in dialog the SUBSCRIBE message requests with the same Call-Id and tags (to and from) for out-of-band (OOB) DTMF for Event header: telephone event. There can be an ID parameter in it, but the gateway supports in-dialog subscription for only one event. After the subscription is accepted, an initial NOTIFY message request is sent and includes a Subscription-State header as per RFC 3265.

When a digit is pressed on the PSTN end, the digit event is sent in the NOTIFY message requests. The Subscription-State header in these requests is active.

When the subscription expires before it is refreshed, the gateway terminates it by sending a NOTIFY message request with a Subscription-State header value set to terminated. The subscriber can always refresh the subscription by sending another SUBSCRIBE message request with the same Call-Id and tags as in the initial SUBSCRIBE message request.

If the INVITE message request dialog is terminated before the subscription expires, the subscription is terminated by sending a NOTIFY message request with a Subscription-State header value set to terminated. The gateway does not support generating in-dialog SUBSCRIBE message request.

SIP NOTIFY-Based Out-of-Band DTMF Relay

The Skinny Client Control Protocol (SCCP) IP phones do not support in-band DTMF digits; they are capable of sending only out-of-band DTMF digits. To support SCCP devices, originating and terminating SIP gateways can use Cisco-proprietary NOTIFY-based out-of-band DTMF relay. In addition, NOTIFY-based out-of-band DTMF relay can also be used by analog phones attached to analog voice ports (FXS) on the router.

NOTIFY-based out-of-band DTMF relay sends messages bidirectionally between the originating and terminating gateways for a DTMF event during a call. If multiple DTMF relay mechanisms are enabled on a SIP dial peer and are negotiated successfully, NOTIFY-based out-of-band DTMF relay takes precedence.

The originating gateway sends an INVITE message with a SIP Call-Info header to indicate the use of NOTIFY-based out-of-band DTMF relay. The terminating gateway acknowledges the message with an 18x or 200 Response message, also using the Call-Info header. The Call-Info header for NOTIFY-based out-of-band relay appears as follows:

Call-Info: <sip: address>; method="NOTIFY;Event=telephone-event;Duration=msec"

Note Duration is the interval between NOTIFY messages sent for a single digit and is set by means of the **notify telephone-event** command.

The NOTIFY-based out-of-band DTMF relay mechanism is negotiated by the SIP INVITE and 18x/200 Response messages. Then, when a DTMF event occurs, the gateway sends a SIP NOTIFY message for that event. In response, the gateway expects to receive a 200 OK message.

The NOTIFY-based out-of-band DTMF relay mechanism is similar to the DTMF message format described in RFC 2833. NOTIFY-based out-of-band DTMF relay consists of 4 bytes in a binary encoded format. The message format is shown in the figure below; the table below describes the fields.

Figure 60: Message Format of NOTIFY-Based Out-of-Band DTMF Relay

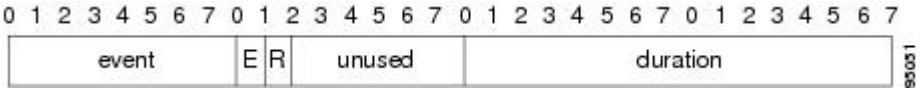


Table 30: Fields in NOTIFY-Based Out-of-Band DTMF relay Message

Field	Description
event	The DTMF event that is between 0--9, A, B, C, D, #, * and flash.
E	E signifies the end bit. If E is set to a value of 1, the NOTIFY message contains the end of the DTMF event. Thus, the duration parameter in this final NOTIFY message measures the complete duration of the event.
R	Reserved.
unused	In RFC 2833, unused corresponds to the volume field, but is not used in NOTIFY-based out-of-band DTMF relay.
duration	Duration of this DTMF event, in milliseconds.

Sending NOTIFY Messages

As soon as the DTMF event is recognized, the gateway sends out an initial NOTIFY message for this event with the duration negotiated in the Call-Info header of the SIP INVITE. For the initial NOTIFY message, the end bit is set to zero. Afterward, one of the following actions can occur:

- If the duration of the DTMF event is less than the negotiated duration, the originating gateway sends an end NOTIFY message for this event with the duration field containing the exact duration of the event and the end bit set to 1.
- If the duration of the DTMF event is greater than the negotiated duration, the originating gateway sends another NOTIFY message for this event after the initial timer runs out. The updated NOTIFY message has a duration of twice the negotiated duration. The end bit is set to 0 because the event is not yet over. If the event lasts beyond the duration specified in the first updated NOTIFY message, another updated NOTIFY message is sent with three times the negotiated duration.
- If the duration of the DTMF event is exactly the negotiated duration, either of the preceding two actions occurs, depending on whether the end of the DTMF event occurred before or after the timer ran out.

For example, if the negotiated duration is 600 ms, as soon as a DTMF event occurs, the initial NOTIFY message is sent with duration as 600 ms. Then a timer starts for this duration.

- If the DTMF event lasts only 300 ms, the timer stops and an end NOTIFY message is sent with the duration as 300 ms.
- If the DTMF event lasts longer than 600 ms (such as 1000 ms), when the timer expires an updated NOTIFY message is sent with the duration as 1200 ms and the timer restarts. When the DTMF event ends, an end NOTIFY message is sent with the duration set to 1000 ms.

Every DTMF event corresponds to at least two NOTIFY message requests: an initial NOTIFY message and an end NOTIFY message. There might also be some update NOTIFY message requests involved, if the total duration of the event is greater than the negotiated max-duration interval. Because DTMF events generally last for less than 1000 ms, setting the duration using the **notify telephone-event** command to more than 1000 ms reduces the total number of NOTIFY messages sent. The default value of the **notify telephone-event** command is 2000 ms.

Receiving NOTIFY Messages

Once a NOTIFY message is received by the terminating gateway, the DTMF tone plays and a timer is set for the value in the duration field. Afterward, one of the following actions can occur:

- If an end NOTIFY message for a DTMF event is received, the tone stops.
- If an update is received, the timer is updated according to the duration field.
- If an update or end NOTIFY message is not received before the timer expires, the tone stops and all subsequent NOTIFY messages for the same DTMF event or DTMF digit are ignored until an end NOTIFY message is received.
- If a NOTIFY message for a different DTMF event is received before an end NOTIFY message for the current DTMF event is received (which is an unlikely case), the current tone stops and the new tone plays. This is an unlikely case because for every DTMF event there needs to be an end NOTIFY message, and unless this is successfully sent and a 200 OK is received, the gateway cannot send other NOTIFY messages.

Note In-band tones are not passed while NOTIFY-based out-of-band DTMF relay is used as the DTMF relay message request.

Two commands allow you to enable or disable NOTIFY-based out-of-band DTMF relay on a dial peer. The functionality is advertised to the other end using INVITE messages if it is enabled by the commands, and must be configured on both the originating and terminating SIP gateways. A third command allows you to verify DTMF relay status:

- **dtmf-relay (VoIP)**
- **notify telephone-event**
- **show sip-ua status**

The NOTIFY message request has a Subscription-State header per RFC 3265. Refer to the "Configuring SIP DTMF Features" module for additional information that relates to the DTMF feature.

Support for RFC 3312--QoS

This feature provides implementation on the gateway with suitable enhancements to the common stack to support quality of service (QoS) RSVP calls adhering to RFC 3312. This feature changes the existing implementation and follows RFC 3312 to provide QoS services using RSVP.

SIP Portable Stack Considerations for QoS

The portable SIP stack is unaware of the type of call (QoS or regular). All QoS-related information carried by SIP or SDP are passed by or to the application. The application takes the necessary steps to distinguish the type of call and handle it accordingly, transparent to the portable SIP stack. From the portable SIP stack's perspective, the call flow for establishing a QoS call is similar to that of a non-QoS call. The only additions to the portable SIP stack application for establishing or modifying QoS calls are as follows:

- Ability to send the UPDATE message request
- Support for initiating and handling 183 and PRACK message request for midcall INVITE message requests

Behavior for QoS with RFC 3312 for Cisco IOS Gateways

The following lists the behavior that SIP QoS calls exhibits on Cisco IOS gateways, with RFC 3312 complaint stack as opposed to existing ICEE implementation:

- The QoS information is conveyed and confirmed through the following set of SDP attributes as proposed by RFC 3312.

Current Status--This attribute carries the current status of the network resources for a particular media stream in either offer or answer SDP. The gateways generates the following values depending on the state of the reservation.

Desired Status--This attribute states the preconditions for a particular media stream. For the Cisco IOS gateway the reservation is always applicable end-to-end status with resources reserved in either direction. The strength tag is configurable.

Confirmation Status--This attribute carries the information for the other gateway to notify back (using the UPDATE message request) once resource reservations are done on its end. On Cisco IOS gateways the originating gateways never request confirmation from the terminating gateway and if that fails, then the call is not presented and is terminated with a 580 (Precondition Failure) message response. The terminating gateway always asks for confirmation from the originating gateway when its reservations are done using the UPDATE message request. This is requested through the 183 message response for the INVITE message request.

- RFC 3312 requires the UA to use an UPDATE message request to notify the other gateway with the confirmation once the reservations are done on its end. The UPDATE message request transaction happens only if the received 183 message response contained the confirmation status attribute. The COMET message request is being used to convey that the reservations are met. With the RFC 3312 compliancy the COMET message request usage is obsoleted.
- The originated INVITE message request contains the precondition option tag for use in Require and Supported header fields as in RFC 3312. With this the Content-Disposition header and Session=qos headers for QoS calls are no longer used.
- RFC 3312 suggests that the UA includes SDP (indicating QoS failure) in 580 Precondition Not Met message response. If a UAC does not make an QoS offer in the INVITE message request or gets a bad QoS offer in 18x or 2xx message response, then corresponding CANCEL or BYE message request contains an SDP body indicating QoS failure. This behavior is recommended but not mandatory as per RFC. This is kept as similar to existing implementation; 580, CANCEL, and BYE message requests continue to be sent, without SDP.
- RFC 3312 suggests the use of reliable provision responses (183/PRACK/200OK) for doing midcall QoS modifications. The current stack implementation uses the offer or answer model (Re-INVITE/200 OK/ACK) to do QoS modifications after the call is active. The new RFC recommendation for midcall does not give any advantage or extra functionality over the existing implementation. It complicates the midcall handling done by the stack. Midcall reliable provisional responses are not used by any other SIP feature, and there is no application that has an immediate need for this midcall functionality. Hence this feature continues using the existing stack's midcall INVITE offer/answer transaction for doing RSVP modifications for QoS calls.

Backward Compatibility

The QoS call flows are not backward compatible on Cisco IOS gateways. SIP continues to use existing RSVP Cisco IOS subsystem and its APIs but the SIP or SDP signalling involved is different from the existing implementation.

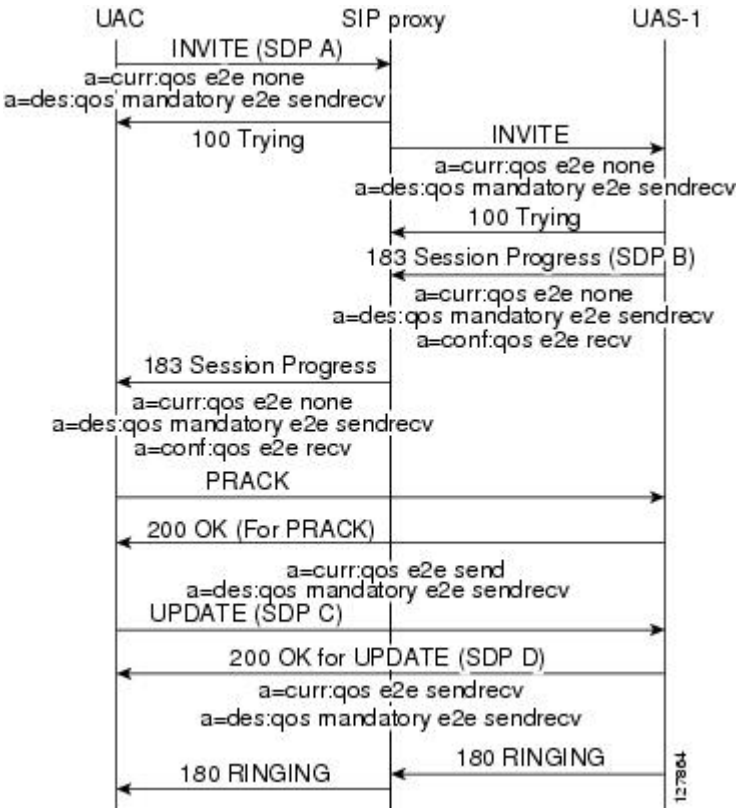
COMET Message Request Obsolescence

This feature stack is obsoletes the usage (sending or receiving) of COMET message requests. This message request is replaced by UPDATE message request. This change has minor impact on the Call Admission Control feature on Cisco IOS gateways. QoS is the only other feature to use COMET. The CAC feature is using the UPDATE message request instead of COMET.

QoS Call Flow

The flows shown in the figure below show a two-party call that invokes RSVP services to reserve resources before the called party is alerted. On the Cisco IOS gateway for this feature implementation the originating gateway does not need confirmation for INVITE message request preconditions. All the QoS SDP attributes shown are media-level attributes. If multiple media lines are associated with their own QoS attributes, then only the first media line QoS is honored.

Figure 61: Successful QoS Call Establishment



Support for the Achieving SIP RFC Compliance Feature

The Achieving SIP RFC Compliance feature enhances SIP gateway compliance for RFCs 3261, 3262, and 3264. This feature inherits these enhancements for the portable stack. Refer to the “Achieving SIP RFC Compliance” chapter for a description of introduced enhancements.

Enhanced Redirect Handling

The portable stack handles redirections (3xx or 485 message responses) internally. When a 3xx or 485 class message response is received by the SIP stack, the stack sends out a new INVITE message request to the

contact in the 3xx message response, without notification to the application. In this feature, the functionality is opened up to the application. Upon receipt of a 3xx or 485 message response the application has the ability to take over the redirect response. When the application decides to handle the redirect, the SIP stack disconnects the original call that the 3xx or 485 message response received, and the application takes over responsibilities for setting up the new call.

Cisco IOS Behavior

There are no changes in the handling of redirects in Cisco IOS software. The stack continues to perform the redirections.

Diversion Header Draft 06 Compliance

This feature upgrades the Diversion header draft implementation to the draft-levy-diversion-06.txt version. This upgrade adds the capability to send or receive two new parameters in the Diversion header. The stack adds two new fields to set or pass this information to and from the application.

Note The draft-levy-diversion-06.txt version has since expired. Current standard uses History-Info header (refer to RFC 4244 , An Extension to the Session Initiation Protocol (SIP) for Request History Information .

SIP Domain Name Support in SIP Headers

The SIP: Domain Name Support in SIP Headers feature adds a command line interface (CLI) switch to provide a host or domain name in the host portion of the locally generated SIP headers (for example, From, RPID, and Call-ID). This feature also affects outgoing dialog-initiating SIP requests (for example, INVITE and SUBSCRIBE message requests).

To configure this feature, you should understand the following concepts:

Vendor-specific attribute (VSA) is introduced to generate information about the locally configured host or domain name in the accounting records generated by the gateway. For a complete list of VSA changes, see the RADIUS VSA Voice Implementation Guide .

Call Active and History VoIP Records

Call active and history VoIP records present the local hostname. They have the following format:

```
#show call active voice
VOIP:
LocalHostname=example.com
```

These records are generated for calls created in the context of the INVITE message request.

SIP Headers

The CLI affects the host portion of the following SIP headers generated for an outbound VoIP call from the SIP gateway:

- Call-ID--The Call-ID header in the SIP messages has an existing format of unique-string@ipaddr. With the CLI, the Call-ID has a value in the form of unique-string@localhostname or unique-string@domain-name. The dialog initiating the SIP requests that are affected namely are the INVITE and SUBSCRIBE message requests.

- From--The From header in the following dialog initiating requests. The INVITE and SUBSCRIBE message requests originating from the gateway have host or domain name in the host portion of the SIP URI. When the CLI is configured, the Remote-Party-ID header also has a hostname in the host portion of the SIP URI. The Remote-Party-ID header is sent out in the INVITE and INFO message requests from the gateway.

Other SIP headers such as Contact and Via are not affected by configuring the new CLI. Those headers continue to have IP addresses even when the CLI is configured.

These changes do not affect the Session Definition Protocol (SDP).

SIP headers that are provided by the application to SIP via header passing mechanisms always override headers generated by SIP.

Sample SIP Header Messages

This section contains the following sample SIP header messages with the SIP: Domain Name Support in SIP Headers feature disabled and enabled:

Feature Disabled--INVITE Message Request Sent from the Gateway

```
Sent:
INVITE sip:9002@example.sip.com:5060 SIP/2.0
Via:SIP/2.0/TCP 172.18.195.49;branch=z9hG4bK597
Remote-Party-ID:<sip:9001@172.18.195.49>;party=calling;screen=no;privacy=off
From:<sip:9001@172.18.195.49>;tag=3AA7574-11BA
To:<sip:9002@example.sip.com>
Date:Tue, 31 Aug 2004 13:40:57 GMT
Call-ID:3924408D-FA8A11D8-80208D32-72E3122E@172.18.195.49
Supported:100rel,timer,resource-priority
Min-SE:1800
Cisco-Guid:940277299-4203352536-2149420338-1927483950
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, Refer , SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq:101 INVITE
Max-Forwards:70
Timestamp:1093959657
Contact:<sip:9001@172.18.195.49:5060;transport=tcp>
Expires:180
Allow-Events:telephone-event
Content-Type:multipart/mixed;boundary=uniqueBoundary
Mime-Version:1.0
Content-Length:418
--uniqueBoundary
Content-Type:application/sdp
Content-Disposition:session;handling=required
v=0
o=CiscoSystemsSIP-GW-UserAgent 4780 5715 IN IP4 172.18.195.49
s=SIP Call
c=IN IP4 172.18.195.49
t=0 0
m=audio 18336 RTP/AVP 18 101 19
c=IN IP4 172.18.195.49
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:19 CN/8000
```

```
a=ptime:20
--uniqueBoundary--
```

Feature Enabled--INVITE Message Request Sent from the Gateway

```
Sent:
INVITE sip:9002@example.sip.com:5060 SIP/2.0
Via:SIP/2.0/TCP 172.18.195.49;branch=z9hG4bK22C7
Remote-Party-ID:<sip:9001@gw11.example.com>;party=calling;screen=no;privacy=off
From:<sip:9001@gw11.example.com>;tag=39CF740-FFC
To:<sip:9002@example.sip.com>
Date:Tue, 31 Aug 2004 13:26:13 GMT
Call-ID:2A101AD3-FA8811D8-801C8D32-72E3122E@gw11.example.com
Supported:100rel,timer,resource-priority
Min-SE:1800
Cisco-Guid:686218050-4203221464-2149158194-1927483950
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, Refer , SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq:101 INVITE
Max-Forwards:70
Timestamp:1093958773
Contact:<sip:9001@172.18.195.49:5060;transport=tcp>
Expires:180
Allow-Events:telephone-event
Content-Type:multipart/mixed;boundary=uniqueBoundary
Mime-Version:1.0
Content-Length:418
--uniqueBoundary
Content-Type:application/sdp
Content-Disposition:session;handling=required
v=0
o=CiscoSystemsSIP-GW-UserAgent 5250 7833 IN IP4 172.18.195.49
s=SIP Call
c=IN IP4 172.18.195.49
t=0 0
m=audio 18998 RTP/AVP 18 101 19
c=IN IP4 172.18.195.49
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:19 CN/8000
a=ptime:20
--uniqueBoundary--
```

SIP Gateway Support for SDP Session Information and Permit Hostname CLI

The SIP GW Support for SDP Session Information and Permit Hostname CLI Feature adds support to Cisco IOS SIP gateways for both SDP session information and validation of hostnames in initial INVITE requests.

SDP Changes for Session Information Line

The SDP Session Information line can exist multiple times within a session description. The line, represented by “i=” in the SDP, can be present at the session-level as well as the media-level. You can have only one session description per packet. The session description contains one session-level, but can have multiple media-levels.

The following is a sample SDP description. The highlighted lines represent the updates to reflect RFC 2327:

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
i=media-information 1
m=video 51372 RTP/AVP 31
i=media-information 2
m=application 32416 udp wb
a=orient:portrait
```

The session information is optional, therefore internal structures are not built to expect this parameter. Specifically, internal memory is only allocated for this parameter when it is present in SDP, or when the application specifies that it be built into an outgoing message. In order to protect the internal operation of the Cisco IOS gateway, the maximum allowable length of a received session information line is 1000 characters. Session information lines over 1000 characters are truncated.

While the RFC detailing SDP indicates to only expect one session information line at the appropriate level, the Cisco IOS gateway will not “drop” the SDP in the event that this rule is violated. In the event that multiple “i=” lines are received at a particular level, the first parsed line that contains data is stored. All subsequent lines for that level are dropped.

Validating Hostname in Initial INVITE Request URI

Beginning with Cisco IOS Software Release 12.4(9)T, administrators can validate hostnames of incoming initial INVITE messages. When the gateway processes an initial INVITE, a determination is made whether or not the host portion is in ipv4 format or a domain name.

If the host portion is an IP address, its IP address is compared with the interfaces on the gateway. If a match is found, the INVITE is processed as normal. If there is not a match, the gateway sends a **400 Bad Request - ‘Invalid IP Address’** message.

If the initial INVITE has a domain name in the host of the request URI, the gateway checks this domain name against a list of configured hostnames. If you configure no hostnames, existing behavior executes and the INVITE is processed. If you configure hostnames for this gateway, the gateway compares the host name in the request URI to the configured hostname list. If a match is found, the INVITE is processed as normal. If there is not a match, the gateway sends a **400 Bad Request - ‘Invalid host’** message.

You can configure up to 10 hostnames by re-entering the **permit hostname dns**command. Use the **no** form of this command to remove any configured hostnames.

The following example shows a configured list of hostnames. The highlighted lines represent the updates to reflect RFC 2327.

```
sip-ua
retry invite 1
registrar ipv4:172.18.193.97 expires 3600
permit hostname dns:sinise.sip.com
permit hostname dns:liotta.sip.com
permit hostname dns:sipgw.sip.com
permit hostname dns:yourgw.sip.com
permit hostname dns:csps.sip.com
!
```

The following example shows an initial INVITE message with a hostname. The highlighted line represents the updates to reflect RFC 2327.

```
INVITE sip:777@sinise.sip.com;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.201.173:5060;branch=z9hG4bK2C419
To: <sip:777@172.18.197.154>
From: <sip:333@64.102.17.246>;tag=B87C0-B65
Date: Thu, 23 Feb 2006 16:49:26 GMT
Call-ID: 4EAF670B-A3C311DA-80148B65-6E225A8E@172.18.197.154
Contact: <sip:333@172.18.201.173>
Supported: 100rel, eatit
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer, SUBSCRIBE, NOTIFY, INFO
Max-Forwards: 70
Cseq: 104 INVITE
Expires: 60
Timestamp: 730947404
Content-Length: 211
Content-Type: application/sdp
^M
v=0
o=CiscoSystemsSIP-GW-UserAgent 6109 4520 IN IP4 172.18.201.173
s=SIP Call
c=IN IP4 111.11.111.111
t=0 0
m=audio 16880 RTP/AVP 0 19
c=IN IP4 111.11.111.111
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20
```

Outbound Proxy Support for the SIP Gateway

The Outbound Proxy Support for the SIP Gateway feature allows you to configure an outbound proxy server on a SIP gateway. You can use the **outbound-proxy** command to globally configure a SIP gateway to send all dialog initiating requests (such as INVITE, SUBSCRIBE, and REGISTER) to a specified destination. You can also use the **voice-class sip outbound-proxy** command to configure these settings on an individual dial peer, overriding the global gateway settings (refer to the *Cisco IOS Voice Command Reference*).

The request-uri of these dialog initiating requests are extracted from the session-target and does not reflect that the request is sent to a configured outbound-proxy server. The outbound-proxy server, based on the host in the request-uri, routes it accordingly. However, in some scenarios, it is possible that calls coming in over a SIP trunk to Cisco Unified CME get forwarded to the outbound SIP proxy rather than directly to the phone. To correct this behavior, use the **outbound-proxy system** command to configure SIP line-side phones on Cisco Unified CME (refer to the *Cisco Unified Communications Manager Express Command Reference*).

SIP SIP Support for PAI

The SIP Support for PAI feature allows you to configure privacy headers into associated SIP request messages, as defined in RFC 3323 and RFC 3325. This feature introduces the **privacy and asserted-id commands which you can use to build various privacy-header requests into common SIP messages,** as shown in the table below.

Table 31: Privacy Header Request Options

Cisco IOS Command	SIP Message Header Options
privacy	ACK, BYE, CANCEL, INVITE, OPTIONS, SUBSCRIBE, NOTIFY, PRACK, IFO, and UPD
privacy pai	BYE, INVITE, OPTIONS, SUBSCRIBE, NOTIFY, and Refer
privacy ppi	BYE, INVITE, OPTIONS, SUBSCRIBE, NOTIFY, and Refer

SIP History-info Header Support

The SIP History-info Header Support feature provides support for the history-info header in SIP INVITE messages only. The SIP gateway generates history information in the INVITE message for all forwarded and transferred calls. The history-info header records the call or dialog history. The receiving application uses the history-info header information to determine how and why the call has reached it.

The receiving application can use the call or dialog history to enhance services, such as calls to voice mail servers or sessions initiated to call centers from a click-to-talk SIP URL on a web page.

To configure the SIP History-info Header Support feature, you need to understand the following concepts:

Feature Design of SIP History-Info Header Support

Cisco implements SIP History-Info Header Support on SIP TDM gateways and SIP-SIP Cisco UBEs by supporting the history-info header, as defined in RFC 4244 , An Extension to the Session Initiation Protocol (SIP) for Request History Information. The history-info header forms part of the SIP INVITE messages that establish media sessions between user agents, and the subsequent responses to the INVITE messages.

Support for history-info headers on a gateway is enabled using the **history-info** command. The system supports multiple history-info headers (maximum of nine) for a single INVITE message. The headers are contained in a comma-separated list.

SIP History-Info Header Support on SIP TDM Gateways

When the TDM gateway sends an INVITE message, it creates the history-info header based on the request URI.

When the gateway receives a redirected PSTN call, it builds the history-info header using the redirect information provided by the PSTN source signaling address, the local host configuration (DNS name), and the host registrar.

To maintain the correct order and to record any redirection of a request, the header includes index information (as a series of dot-delimited digits). The index format is defined in RFC 4244 section 4.3.3.1.3.

If history-info headers are enabled for the SIP stage, the gateway sends both diversion headers and history-info headers in the outbound request. However, the history-info header takes precedence when the gateway maps the header to the ISDN redirect number.

Note SIP-TDM gateways support the latest redirecting-number information from the ISDN leg to the peer SIP leg.They do not send multiple redirecting number-information. The History-Info header is inserted on the peer SIP leg based on the latest redirecting number.

SIP History-Info Header Support on SIP-SIP Cisco Unified Border Element Gateways

When the Cisco UBE gateway receives an inbound INVITE message without a history-Info header, it generates the history-Info header based on the request URI of the outbound INVITE message. If privacy is enabled on the gateway, history is added to the privacy settings.

When Cisco UBE receives an outbound message, it adds the history-Info header to the message based on the request URI. Cisco UBE supports multiple history-Info headers, and history-Info headers received on the incoming SIP call leg are sent to the peer SIP leg. Cisco UBE can support upto a maximum of nine history-Info headers. If Cisco UBE receives a message with nine or more headers, it keeps only the first eight messages and adds a new header to the end of the header list.

When history-Info privacy is configured on the Cisco UBE, it transparently passes all history-Info and privacy headers in the message from one SIP stage to the next.

If history-Info headers are enabled for the SIP stage, the gateway behaves as follows:

- If no history-Info header is present, Cisco UBE converts the diversion headers to history-Info headers and sets the *cause* parameter to 302. Cisco UBE, then sends both the diversion and the history-Info headers to the next SIP stage.
- If no diversion headers are present, Cisco UBE converts all the history-Info headers with the cause parameter is set to 302 to diversion headers. Cisco UBE then sends both the diversion and history-Info headers to the next SIP stage.
- If both diversion headers and history-Info headers are present, no conversion is performed.

If history-Info headers are disabled for the SIP stage, Cisco UBE sends all diversion headers (including any new diversion headers) to the next SIP stage.

SIP Trunk Registration

The Cisco IOS gateway registers all its POTS dial peers to the registrar when the registrar is configured on the Gateway. With the introduction of trunk registration support, registration of a single number would represent the SIP trunk. SIP trunk registration can be associated with multiple dial-peers for routing outbound calls. This registration will represent all the gateway end points for routing calls from or to the endpoints.

IOS-SIP gateway sends the REGISTER request to the configured registrar after resolving the outbound-proxy DNS name.On successful registration, IOS-SIP gateway re-uses the Outbound Proxy IP address, port number, service-route response received for sending subsequent REGISTER/INVITE.

Support for SIP 181 Call is Being Forwarded Message

Support for SIP 181 Call is Being Forwarded message on Cisco IOS SIP TDM gateways and Cisco UBEs. This feature is enabled by default, allowing Cisco IOS SIP TDM gateways and Cisco UBEs to pass SIP 181 messages as is. To disable this feature for all SIP 181 messages or for SIP 181 messages either with or without SDP, see the **block** and **voice-class sip block** commands in the *Cisco IOS Voice Command Reference*.

On the Cisco UBE, this feature also adds the ability to receive SIP 181 messages on one leg and send out SIP 183 messages on the other leg. For details about enabling this feature on a Cisco UBE, see the **map resp-code** and **voice-class sip map resp-code** commands, also in the *Cisco IOS Voice Command Reference*.

Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

Support for resetting the Expires timer upon receipt of SIP 183 messages on Cisco Unified Communications Manager Express (Cisco Unified CME), a Cisco IOS voice gateway, or a Cisco UBE. When the terminating device lacks answer supervision or does not send the required SIP 200 OK message within the timer expiry, you can enable this feature to send periodic SIP 183 messages to reset the Expires timer and preserve the call until final response. This feature can be enabled globally or on a specific dial peer. Additionally, you can configure this feature based on the presence or absence of SDP.

For details about enabling this feature, see the **reset timer expires** and **voice-class sip reset timer expires** commands in the *Cisco IOS Voice Command Reference*.

Support for Stripping off Progress Indication from Incoming ISDN Messages on SIP and H.323 TDM Gateways

You can use the **progress_ind** command to configure replacement behavior on outbound dial peers on a Cisco IOS SIP or H.323 TDM voice gateway or Cisco UBE to ensure proper end-to-end signaling of VoIP calls. You can also use this command to configure removal (stripping) of PIs on outbound dial peers on Cisco IOS voice gateways or Cisco UBEs, such as when configuring a Cisco IOS SIP gateway (or SIP-SIP Cisco UBE) to not generate additional SIP 183 Session In Progress messages.

However, before configuring the **progress_ind** command on an outbound dial peer, you must configure a destination pattern on the dial peer. To configure a destination pattern for an outbound dial peer, use the **destination-pattern** command in dial peer voice configuration mode. Once you have set a destination pattern on the dial peer, you can then use the **progress_ind** command, also in dial peer voice configuration mode, to override and replace or remove the default progress indication (PI) in specific call message types.

For messages that contain multiple PIs, behavior configured using the **progress_ind** command will override only the first PI in the message. Additionally, configuring a replacement PI will not result in an override of the default PI in call Progress messages if the Progress message is sent after a backward cut-through event, such as when an Alert message with a PI of 8 was sent before the Progress message.

For details about enabling this feature, see the **progress_ind** command in the *Cisco IOS Voice Command Reference*.

How to Configure SIP Message Timer and Response Features

Note For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring Internal Cause Code Consistency Between SIP and H.323

Configure Internal Cause Code Consistency Between SIP and H.323

The standard set of cause-code categories that is now generated for internal voice call failures is used by default. To configure internal failures with existing or nonstandard H.323 and SIP cause codes, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **cause-code legacy**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**cause-code legacy** **Example:** `Router(config-voi-srv)# cause-code legacy`	Represents internal failures with existing or nonstandard H.323 or SIP cause codes. The keyword is as follows: • **legacy** --Sets the internal cause code to the former and nonstandard set of values. Used for backward compatibility.
Step 5	**exit** **Example:** `Router(config-voi-srv)# exit`	Exits the current mode.

Configuring SIP - Configurable PSTN Cause Code Mapping

Map PSTN Codes to SIP Status Codes

To configure incoming PSTN cause codes to SIP status codes, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **set pstn-cause** *value* **sip-status** *value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**set pstn-cause** *value* **sip-status** *value* **Example:** `Router(config-sip-ua)# set pstn-cause 111 sip-status 400`	Use this command to map an incoming PSTN cause code to a SIP error status code. Keywords and arguments are as follows: • **pstn-cause** *value* --PSTN cause code. Range: 1 to 127. • **sip-status** *value* --SIP status code that is to correspond with the PSTN cause code. Range: 400 to 699.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Map SIP Status Codes to PSTN Cause Codes

To map incoming SIP status codes to PSTN cause codes, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **set sip-status** *value* **pstn-cause** *value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**set sip-status** *value* **pstn-cause** *value* **Example:** `Router(config-sip-ua)# set sip-status 400 pstn-cause 111`	Maps an incoming PSTN cause code to a SIP error status code. Keywords and arguments are as follows: • **sip-status** *value* --SIP status code that is to correspond with the PSTN cause code. Range: 400 to 699. • **pstn-cause** *value* --PSTN cause code. Range: 1 to 127.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Accept-Language Header Support

You can configure Accept-Language header support in two different configuration modes: voice service configuration mode and dial-peer voice configuration mode. The gateway first checks for languages configured

under the dial-peer voice configuration mode and failing a match will then default to the global voice service configuration. If no languages are configured in either mode, then the header is not added.

Note For the Accept-Language header to be included in the 200 OK response to an OPTIONS request, you must enable this feature in voice service configuration mode.

Perform this task to enable Accept-Language header support and specify languages carried in the Accept-Language header of SIP INVITE requests and OPTIONS responses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **voice service pots**
   -
   - **dial-peer voice** *tag* **pots**
4. **supported-language** *language-code* **language-param** *qvalue*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • **voice service pots** • • **dial-peer voice** *tag* **pots** **Example:** `Router(config)# voice service pots` **Example:** **Example:**	Enters global voice service configuration mode or dial-peer voice configuration mode. **Note** Voice service configuration mode configures the gateway to support the Accept-Language header in both outgoing SIP INVITE messages and OPTIONS responses. Dial-peer voice configuration mode configures it to support the header in INVITE messages only.

	Command or Action	Purpose
	Example: `Router(config)# dial-peer voice 1 pots`	
Step 4	**supported-language** *language-code* **language-param** *qvalue* **Example:** `Router(conf-voi-serv)# supported-language EO language-param .25`	Specifies languages carried in the Accept-Language header in outgoing SIP INVITE messages and OPTIONS responses. Keywords and arguments are as follows: • *language-code* --Any of 139 supported languages designated by a two-letter ISO-639 country code. The note below shows a partial list of supported language codes and languages. To display a complete listing, use the help command supported-language ? • **language-param** *qvalue* --Priority of the language, in descending order according to the assigned parameter value. You can assign a value for each language. Range: 0, a decimal fraction in the range 0.001 to 0.999, and 1. Default: 1 (highest priority).
Step 5	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits the current mode.

What to do next

The following is a partial list of supported language codes and languages. To display a complete listing, use the help command supported-language ?.

AR--Arabic	HE--Hebrew	ES--Spanish
ZH--Chinese	GA--Irish	SW--Swahili
EN--English	IT--Italian	SV--Swedish
EO--Esperanto	JA--Japanese	VI--Vietnamese
DE--German	KO--Korean	YI--Yiddish
EL--Greek	RU--Russian	ZU--Zulu

Configuring SIP Enhanced 180 Provisional Response Handling

This feature allows you to do the following:

- Enable or disable early media cut-through treatment for SIP 180 messages with SDP
- Configure uniform call treatment for 180 messages with or without SDP

Note Early media cut-through for 180 messages with SDP is disabled by default; no configuration tasks are required to disable it. To re-enable the feature or to disable it after it has been re-enabled, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. Do one of the following:
   - **disable-early-media 180**
   -
   -
   - **no disable-early-media 180**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	Do one of the following: • **disable-early-media 180** • • • **no disable-early-media 180** **Example:** `Router(config-sip-ua)# disable-early-media 180` **Example:**	Disables or (by means of **no** form of the command) reenables early media cut-through for 180 messages with SDP.

	Command or Action	Purpose
	Example: **Example:** `Router(config-sip-ua)# no disable-early-media 180`	
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Extensions for Caller Identity and Privacy

Configure Remote Party-ID

This feature is enabled by default; no configuration tasks are required to enable this feature. If the feature is disabled by means of the no remote-party-id command, perform this task to re-enable the feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **remote-party-id**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.

	Command or Action	Purpose
Step 4	**remote-party-id** **Example:** `Router(config-sip-ua)# remote-party-id`	(Optional) Configures remote-party-id translation. The following apply: • If a Remote-Party-ID header is present in the incoming INVITE message, the calling name and number extracted from the Remote-Party-ID header are sent as the calling name and number in the outgoing Setup message. This is the default behavior. Use the **remote-party-id** command to enable this option. • When no Remote-Party-ID header is available, no translation occurs so the calling name and number are extracted from the From header and are sent as the calling name and number in the outgoing Setup message. This treatment also occurs when the feature is disabled.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure SIP-to-PSTN Calling-Info Policy

When Remote-Party-ID support is enabled, the default calling-info treatment is the following:

- The calling name and calling number are bidirectionally translated between the display-name and the user part of the Remote-Party-ID header of the SIP INVITE message and the calling name and calling number of the PSTN Setup message.
- If a PSTN to SIP call is marked as presentation prohibited, the display-name is populated with “anonymous”. Otherwise, the display-name and user part of the From header of the outgoing INVITE are populated with the calling name and calling number.

To override the default calling-info treatment, perform this task to optionally configure SIP to PSTN calling-info policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **calling-info sip-to-pstn** [**unscreened discard**] [**name set** *name*] [**number set** *number*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**calling-info sip-to-pstn** [**unscreened discard**] [**name set** *name*] [**number set** *number*] **Example:** `Router(config-sip-ua)# calling-info sip-to-pstn unscreened discard`	Specifies calling-information treatment for SIP-to-PSTN calls. Keywords and arguments are as follows: • **unscreened discard** --Calling name and number are discarded. If the incoming SIP INVITE message does not contain a screened (;screen=yes) Remote-Party-ID header, then no name or number is sent in the forwarded Setup message. • **name set** *name* --Calling name is unconditionally set to the *name* argument, a configured ASCII string, in the forwarded Setup message. • **number set** *number* --Calling number is unconditional set to the *number* argument, a configured ASCII string, in the forwarded Setup message.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure PSTN-to-SIP Calling-Info Policy

To override the default calling-info treatment, perform this task to optionally configure PSTN to SIP calling-info policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**

4. **calling-info pstn-to-sip** [**unscreened discard**] [**from** [**name set** *name*] [**number set** *number*]] [**remote-party-id** [**name set** *name*] [**number set** *number*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**calling-info pstn-to-sip** [**unscreened discard**] [**from** [**name set** *name*] [**number set** *number*]] [**remote-party-id** [**name set** *name*] [**number set** *number*]] **Example:** `Router(config-sip-ua)# calling-info pstn-to-sip unscreened discard`	Specifies calling-information treatment for PSTN-to-SIP calls. Keywords and arguments are as follows: • **unscreened discard** --Calling name and number are discarded. Unless the screening indicator in the incoming Setup is marked as "user-provided, passed screening" or "network-provided," no calling name or number is sent in the forwarded INVITE message. • **from name set** *name* --Display name of the From header is unconditionally set to the *name* argument, a configured ASCII string, in the forwarded INVITE message. • **from number set** *number* --User part of the From header is unconditionally set to the *number* argument, a configured ASCII string, in the forwarded INVITE message. • **remote-part-id name set** *name* --Display name of the Remote-Party-ID header is unconditionally set to the *name* argument, a configured ASCII string, in the forwarded INVITE message. • **remote-party-id number set** *number* --User part of the Remote-Party-ID header is unconditionally set to the *number* argument, a configured ASCII string, in the forwarded INVITE message.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP INVITE Request with Malformed Via Header

There are no configuration steps for this feature. Use the **show sip-ua statistics** command (see "Verifying SIP Message Components Session Timers and Responses Configuration") to display the Bad Request counter.

Configuring Privacy Headers

You can configure privacy headers according to values defined in RFC 3323 and RFC 3325 as shown in the table below.

Table 32: Privacy Header Values

Header Value	Description
Header	Indicates that the privacy service enforces privacy for all headers in the SIP message which might identify information about the subscriber.
Session	Indicates that the information held in the session description protocol (SDP) is hidden outside the trusted domain.
User	Enforces user-level privacy for the subscriber by removing any user identification from the SIP message.
Critical	Indicates that the message is rejected if the privacy service cannot or will not enforce the specified privacy. **Note** You can only add critical to privacy headers if you also choose a privacy header for user, header, session, or ID.
ID	Indicates the Network-Asserted Identity remains private with respect to SIP entries outside the user-authenticated trusted domain.
PSTN	Indicates information passed in from the PSTN Octet 3a of the CALLING PARTY Information Element enables privacy information on the VoIP side of the call (see the Privacy Header PSTN with UAC Gateway section).
System	Use system settings.
Disable	Disables the privacy.

UAC Gateway Behavior

When you configure the **privacy** command to use one of the header values shown in the table above, then the gateway's outgoing message request contains a privacy header set to the corresponding privacy value. The

following example shows the format of the "From" header, if you configure the **privacy** command, based on RFC 3323:

```
From: "Anonymous" <sip:anonymous@anonymous.invalid>; tag=<tag value>
```

If you configure the **privacy critical**command, the gateway adds a Proxy-Require header with the value set to critical. Thus, in the unlikely event that the user agent sends a request to an intermediary that does not support the described extension, the request will fail.

If you configure the **asserted-id pai**command, the gateway builds a PAI into the common SIP stack. The **asserted-id pai**command has priority over the Remote-Party-ID (RPID) header and removes this header from any outbound message even if the router is configured to use the RPID header.

If you configure the **asserted-id ppi**command, the gateway builds a PPI into the common SIP stack. The **asserted-id ppi**command has priority over the Remote-Party-ID (RPID) header and removes this header from any outbound message even if the router is configured to use the RPID header.

Privacy Header PSTN with UAC Gateway

You can use the **privacy pstn** command to derive information passed in from the PSTN Octet 3a of the CALLING PARTY Information Element to enable privacy information on the VoIP side of the call. The data within the CALLING PARTY field indicates whether or not you want to relay calling information. The CALLING PARTY field also supplies information and details about who supplied the information, and whether or not the information has been verified.

The table below summarizes the relationship between the ISDN Octet 3a values and the SIP-header values that the UAC gateway generates, when you configure the **privacy pstn** command.

Table 33: ISDN Octet 3a-to-SIP Header Mapping for UAC Gateways

ISDN Octet 3a	SIP Headers			
Presentation Status Bit(5)	Number Origin Bits(0,1)	From	Asserted-ID	Privacy
Presentation allowed.	User provided, number not screened.	No change.	PPI.	Not included.
Presentation allowed.	User provided, number passed, and network screened.	No change.	PAI.	Not included.
Presentation allowed.	User provided, number failed, and network screened.	No change.	PPI.	Not included.
Presentation	Network-provided number.	No change.	PAI.	Not included.
Presentation prohibited.	User provided, number not screened (00).	Anonymous.	P-Preferred Identity.	ID.
Presentation prohibited.	User provided, number passed, and network screened (01).	Anonymous.	P-Asserted Identity.	ID.
Presentation prohibited.	Network provided number (11).	Anonymous.	P-Asserted Identity.	ID.

Interaction with Caller ID When Privacy Exists

When you configure the **privacy pstn** command, on the UAC gateway side of the call, after configuring the **substitute name** command under the **clid (voice-service-voip)** command and defining no “Display Name” parameter, then the PAI or PPI substitutes the calling number in the Display field.

The following example show a PAI header when the **substitute name** command is not set:

```
P-Asserted_Identity: <sip:5551212@example.com>
```

If you set the **substitute name** command, the header in the example is modified:

```
P-Asserted_Identity: “5551212” <sip:5551212@example.com>
```

When you configure the **privacy pstn** command, after configuring the **strip pi-restrict all**command under the **clid (voice-service-voip)** command, and if the CALLING INFORMATION Octet 3a indicates that the number is restricted, then the PAI/PPI value is not sent.

On the UAS gateway side of the call, if you configure the **clid network-provided**command, it will override any value you set by using the **privacy**command. If you configure the **clid network-provided**command and a PPI is received, the number in the Octet 3a is set to “Network Provided.” If you do not configure the **clid network-provided**command, the number in the Octet 3a is set to “User Provided.”

If you configure the **calling-info pstn-to-sip unscreened discard**command and the **privacy pstn** command, and if the calling number has a screening indicator of “User-provided, not screened,” or “User-provided, failed screen” the PAI/PPI is not sent.

The table below summarizes the interaction when you configure the **privacy pstn** command.

Table 34: Interactions When Using the privacy pstn Command

Presentation Indication	Screening Indication	calling-info pstn-to-sip Command	Generated Headers
See the SIP: SIP Support for PAI section.	See the SIP: SIP Support for PAI section.	Not set.	If you do not configure the **calling-info pstn-to-sip** command, then see the SIP: SIP Support for PAI section.
Presentation allowed.	User provided, not screened.	Unscreened discard.	From: <sip:example.com>; tag=1 Contact: <sip:example.com>
Presentation allowed.	User provided number passed network screening.	Unscreened discard.	From: <sip:5551212@example.com>; tag=1 Contact: <sip:5551212@example.com:5060> P-Asserted-Identity: <sip:5551212@example.com>
Presentation allowed.	User provided number failed network screening.	Unscreened discard.	From: <sip:example.com>; tag=1 Contact: <sip:example.com>

Presentation Indication	Screening Indication	calling-info pstn-to-sip Command	Generated Headers
Presentation prohibited.	User provided number, not screened.	Unscreened discard.	From: “Anonymous” <sip:anonymous@anonymous.com>; tag=1 Contact: <sip:example.com> Privacy: ID
Presentation prohibited	User provided number, failed network screening.	Unscreened discard	From: “Anonymous” <sip:anonymous@anonymous.com>; tag=1 Contact: <sip:example.com> Privacy: ID
Presentation prohibited	User provided number, passed network screening.	Unscreened discard	From: “Anonymous” <sip:anonymous@anonymous.com>; tag=1 P-Asserted-Identity: Contact: <sip:5551212@example.com:> Privacy: ID
Presentation prohibited	Network-provided number	Unscreened discard	

Configuring SIP Session Timer Support

To configure SIP session timer support including the Min-SE value, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **min-se** *time*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**min-se** *time* **Example:** `Router(conf-serv-ua)# min-se 90`	Changes the minimum-session-expiration header value for all calls that use the SIP session timer support feature. The argument is as follows: • *time* --Time, in seconds. Range: 60 to 86400 (one day). Default: 90 (1.5 minutes).
Step 6	**exit** **Example:** `Router(conf-serv-ua)# exit`	Exits the current mode.

Configuring SIP Cisco IOS Gateway Reason Header and Buffered Calling Name Completion

The SIP: Cisco IOS Gateway Reason Header and Buffered Calling Name Completion feature implements support for Reason headers and buffered calling-name completion. Reason-header support on Cisco IOS gateways is defined by RFC 3326.

Feature benefits include the following:

- Reason-header support facilitates PSTN internetworking by providing a more deterministic method of transporting the actual PSTN disconnect cause code to a remote PSTN gateway.
- Buffered calling-name completion (such as buffered-invite timers) makes the process of receiving ISDN-display information in a subsequent ISDN FACILITY message transparent to the remote SIP endpoint.
- The requirement of an external SIP user-agent server (UAS) to support INFO message responses before the call is active is removed.

Configure Reason-Header Override

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**

4. **reason-header override**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**reason-header override** **Example:** `Router(config-sip-ua)# reason-header override`	Enables Reason-header override support.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure Buffer Calling-Name Completion

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers buffer-invite** *timer*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP-user-agent configuration mode.
Step 4	**timers buffer-invite** *timer* **Example:** `Router(config-sip-ua)# timers buffer-invite 500`	Enables the SIP buffer-invite timer and sets the timer interval. The argument is as follows: • *timer* --Buffer-invite timer value, in ms. Range: 50 to 5000.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP SIP Header URL Support and SUBSCRIBE NOTIFY for External Triggers

Configure SIP Header Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **header-passing**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**header-passing** **Example:** `Router(conf-serv-sip)# header-passing`	Enables or disables SIP header-passing to applications. When the gateway receives SIP INVITE, SUBSCRIBE, and NOTIFY messages, this command enables passing SIP headers associated with these messages to the target application in the gateway.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configure SIP SUBSCRIBE and NOTIFY for External Triggers

To configure SIP subscription options, perform the following steps.

Before you begin

- Enable SIP header passing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **subscription asnl session history** [**duration** *minutes*] [**count** *number*]

6. **subscription maximum originate** *number*
7. **exit**
8. **sip-ua**
9. **retry subscribe** *number*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**subscription asnl session history** [**duration** *minutes*] [**count** *number*] **Example:** `Router(conf-serv-sip)# subscription asnl session history duration 10 count 100`	(Optional) Specifies how long to keep Application SUBSCRIBE/NOTIFY Layer (ASNL) subscription history records and how many records to keep in memory.
Step 6	**subscription maximum originate** *number* **Example:** `Router(conf-serv-sip)# subscription maximum originate 10`	(Optional) Specifies the maximum number of outstanding subscriptions to be originated by the gateway, up to two times the maximum number of dial peers on the platform. Default is the maximum number of dial peers on the platform.
Step 7	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

	Command or Action	Purpose
Step 8	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 9	**retry subscribe** *number* **Example:** `Router(config-sip-ua)# retry subscribe 10`	(Optional) Sets the number of times that a SIP SUBSCRIBE message is resent to the other user agent.
Step 10	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Stack Portability

No configuration tasks are required to configure the SIP stack portability feature. The feature is enabled by default.

Configuring SIP Domain Name Support in SIP Headers

Configure the Hostname in Locally Generated SIP Headers

You can configure the hostname in either of two configuration modes:

Note Dial-peer-specific configuration takes precedence over more general gateway-wide configuration.

Gateway-Wide Configuration Mode

This procedure allows global configuration of the local hostname for use for locally generated URIs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **localhost dns:local-host-name-string**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(config-voip-serv)# sip`	Enters SIP configuration mode.
Step 5	**localhost dns:local-host-name-string** **Example:** `Router(config-serv-sip)# localhost dns:example.com`	Enters a local hostname string.
Step 6	**exit** **Example:** `Router(config-serv-sip\)# exit`	Exits the current mode.

Dial-Peer-Specific Configuration Mode

This procedure allows dial-peer configuration of the local hostname for use for locally generated URIs.

Note This procedure takes precedence over more general gateway-wide configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice tag voip**
4. **voice-class sip localhost [dns]:local host-name-string**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice tag voip** **Example:** `Router# dial-peer voice 100 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 4	**voice-class sip localhost [dns]:local host-name-string** **Example:** `Router(config-dial-peer)# voice-class sip localhost dns:example.com`	Enters a local hostname string.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Monitor the Hostname in Locally Generated SIP Headers

This procedure monitors the gateway-wide or dial-peer-specific configuration.

SUMMARY STEPS

1. **enable**
2. **show call active voice**
3. **show call history voice**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show call active voice** **Example:** `Router# show call active voice`	Displays call information for voice calls in progress.
Step 3	**show call history voice** **Example:** `Router# show call history voice`	Displays the call history table for voice calls.
Step 4	**exit** **Example:** `Router# exit`	Exits the current mode.

Examples

This section provides the following command output:

show call active Command Output: Example

The following example shows active-call command output when the local hostname is enabled.

```
Router# show call active voice



Telephony call-legs:1
SIP call-legs:1
H323 call-legs:0
Call agent controlled call-legs:0
Multicast call-legs:0
Total call-legs:2



 GENERIC:
SetupTime=126640 ms
Index=1
PeerAddress=9001
PeerSubAddress=
PeerId=100
PeerIfIndex=6
LogicalIfIndex=4
ConnectTime=130300 ms
CallDuration=00:00:47 sec
CallState=4
CallOrigin=2
ChargedUnits=0
InfoType=speech
TransmitPackets=2431
TransmitBytes=48620
ReceivePackets=2431
ReceiveBytes=48620
TELE:
```

Examples

```
ConnectionId=[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
IncomingConnectionId=[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
CallID=1
TxDuration=48620 ms


VoiceTxDuration=48620 ms
FaxTxDuration=0 ms
CoderTypeRate=g729r8
NoiseLevel=-61
ACOMLevel=3
OutSignalLevel=-35
InSignalLevel=-30
InfoActivity=2
ERLLevel=3
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=9001
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=9002
GENERIC:
SetupTime=128980 ms
Index=1
PeerAddress=9002
PeerSubAddress=
PeerId=3301
PeerIfIndex=7
LogicalIfIndex=0
ConnectTime=130300 ms
CallDuration=00:00:50 sec
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=speech
TransmitPackets=2587
TransmitBytes=51740
ReceivePackets=2587
ReceiveBytes=51740
VOIP:
ConnectionId[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
IncomingConnectionId[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
CallID=2
RemoteIPAddress=172.18.193.87
RemoteUDPPort=17602
RemoteSignallingIPAddress=172.18.193.87
RemoteSignallingPort=5060
RemoteMediaIPAddress=172.18.193.87
RemoteMediaPort=17602
RoundTripDelay=2 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
```

```
FastConnect=FALSE
AnnexE=FALSE
Separate H245 Connection=FALSE
H245 Tunneling=FALSE
SessionProtocol=sipv2
ProtocolCallId=A240B4DC-115511D9-8005EC82-AB4FD5BE@pip.example.com
SessionTarget=172.18.193.87
OnTimeRvPlayout=48620
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=70 ms
LoWaterPlayoutDelay=69 ms
TxPakNumber=2434
TxSignalPak=0
TxComfortNoisePak=0
TxDuration=48680
TxVoiceDuration=48680
RxPakNumber=2434
RxSignalPak=0
RxDuration=0
TxVoiceDuration=48670
VoiceRxDuration=48620
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
PlayDelayCurrent=69
PlayDelayMin=69
PlayDelayMax=70
PlayDelayClockOffset=43547
PlayDelayJitter=0
PlayErrPredictive=0
PlayErrInterpolative=0
PlayErrSilence=0
PlayErrBufferOverFlow=0
PlayErrRetroactive=0
PlayErrTalkspurt=0
OutSignalLevel=-35
InSignalLevel=-30
LevelTxPowerMean=0
LevelRxPowerMean=-302
LevelBgNoise=0
ERLLevel=3
ACOMLevel=3
ErrRxDrop=0
ErrTxDrop=0
ErrTxControl=0
ErrRxControl=0
ReceiveDelay=69 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
SRTP = off
VAD = enabled
CoderTypeRate=g729r8
CodecBytes=20
Media Setting=flow-around
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=9001
OriginalCallingOctet=0x0
OriginalCalledNumber=9002
OriginalCalledOctet=0x80
```

```
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=9001
TranslatedCallingOctet=0x0
TranslatedCalledNumber=9002
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=9002
GwOutpulsedCalledNumber=9002
GwOutpulsedCalledOctet3=0x80
GwOutpulsedCallingNumber=9001
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
Username=
LocalHostname=pip.example.com  ! LocalHostname field
Telephony call-legs:1
SIP call-legs:1
H323 call-legs:0
Call agent controlled call-legs:0
Multicast call-legs:0
Total call-legs:2
```

show call history Command Output: Example

The following example shows call-history command output when the local hostname is enabled.

```
Router# show call history voice
Telephony call-legs:1
SIP call-legs:1
H323 call-legs:0
Call agent controlled call-legs:0
Total call-legs:2
GENERIC:
SetupTime=128980 ms
Index=1
PeerAddress=9002
PeerSubAddress=
PeerId=3301
PeerIfIndex=7
LogicalIfIndex=0
DisconnectCause=10
DisconnectText=normal call clearing (16)
ConnectTime=130300 ms
DisconnectTime=329120 ms
CallDuration=00:03:18 sec
CallOrigin=1
ReleaseSource=4
ChargedUnits=0
InfoType=speech
TransmitPackets=9981
TransmitBytes=199601
ReceivePackets=9987
ReceiveBytes=199692
VOIP:
ConnectionId[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
IncomingConnectionId[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
CallID=2
RemoteIPAddress=172.18.193.87
RemoteUDPPort=17602
```

```
RemoteSignallingIPAddress=172.18.193.87
RemoteSignallingPort=5060
RemoteMediaIPAddress=172.18.193.87
RemoteMediaPort=17602
SRTP = off
RoundTripDelay=1 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=FALSE
AnnexE=FALSE
Separate H245 Connection=FALSE
H245 Tunneling=FALSE
SessionProtocol=sipv2
ProtocolCallId=A240B4DC-115511D9-8005EC82-AB4FD5BE@pip.example.com
SessionTarget=172.18.193.87
OnTimeRvPlayout=195880
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=70 ms
LoWaterPlayoutDelay=69 ms
ReceiveDelay=69 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
VAD = enabled
CoderTypeRate=g729r8
CodecBytes=20
cvVoIPCallHistoryIcpif=2
MediaSetting=flow-around
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=9001
OriginalCallingOctet=0x0
OriginalCalledNumber=9002
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=9001
TranslatedCallingOctet=0x0
TranslatedCalledNumber=9002
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=9002
GwOutpulsedCalledNumber=9002
GwOutpulsedCalledOctet3=0x80
GwOutpulsedCallingNumber=9001
GwOutpulsedCallingOctet3=0x0
GwOutpulsedCallingOctet3a=0x0
MediaInactiveDetected=no
MediaInactiveTimestamp=
MediaControlReceived=
LocalHostname=pip.example.com ! LocalHostname field
Username=
GENERIC:
SetupTime=126640 ms
Index=2
PeerAddress=9001
PeerSubAddress=
PeerId=100
PeerIfIndex=6
LogicalIfIndex=4
```

```
DisconnectCause=10
DisconnectText=normal call clearing (16)
ConnectTime=130300 ms
DisconnectTime=330080 ms
CallDuration=00:03:19 sec
CallOrigin=2
ReleaseSource=4
ChargedUnits=0
InfoType=speech
TransmitPackets=9987
TransmitBytes=199692
ReceivePackets=9981
ReceiveBytes=199601
TELE:
ConnectionId=[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
IncomingConnectionId=[0xA0DC41CF 0x115511D9 0x8002EC82 0xAB4FD5BE]
CallID=1
TxDuration=195940 ms
VoiceTxDuration=195940 ms
FaxTxDuration=0 ms
CoderTypeRate=g729r8
NoiseLevel=-73
ACOMLevel=4
SessionTarget=
ImgPages=0
CallerName=
CallerIDBlocked=False
OriginalCallingNumber=
OriginalCallingOctet=0x0
OriginalCalledNumber=
OriginalCalledOctet=0x80
OriginalRedirectCalledNumber=
OriginalRedirectCalledOctet=0x0
TranslatedCallingNumber=9001
TranslatedCallingOctet=0x0
TranslatedCalledNumber=
TranslatedCalledOctet=0x80
TranslatedRedirectCalledNumber=
TranslatedRedirectCalledOctet=0x0
GwCollectedCalledNumber=9002
```

Configuring SIP Gateway Support for Session Information

There are no tasks for configuring SIP gateway support for session information.

Configuring SIP Gateway Support for Permit Hostname CLI

To configure a list of hostname to validate against incoming INVITE messages, perform the following steps.

Note Hostname can be a maximum of 30 characters; hostnames longer than 30 characters are truncated.

>

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **sip-ua**
4. **permit hostname dns:** *<domain name>*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router (config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**permit hostname dns:** *<domain name>* **Example:** `Router (config-sip-ua)# permit hostname dns:sip.example.com`	Validates a hostname of initial incoming INVITE messages. The argument is: • *domain name*-- Domain name in DNS format. Domain names can be up to 30 characters in length; those domain names exceeding 30 characters are truncated.
Step 5	**exit** **Example:** `Router (config-sip-ua)# exit`	Exits the current mode.

Configuring Outbound Proxy Support for the SIP Gateway

Configuring an Outbound-Proxy Server Globally on a Gateway

To configure SIP support for an outbound-proxy server globally on a SIP gateway, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service {pots | voatm | vofr | voip}**
4. **sip**

5. **outbound-proxy <** *ipv4:A:B:C:D:* **port/dns:host.domain>**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service {pots | voatm | vofr | voip}** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode and specifies the voice encapsulation.
Step 4	**sip** **Example:** `Router(conf-voi-ser)# sip`	Enters SIP configuration mode.
Step 5	**outbound-proxy <** *ipv4:A:B:C:D:* **port/dns:host.domain>** **Example:** `Router(conf-ser-sip)# outbound-proxy dns:sipproxy.example.com`	Configures an outbound proxy server. This example shows how to configure an outbound-proxy server to a SIP proxy server in the domain example.com.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Configuring an Outbound-Proxy Server on a Dial Peer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **{pots | vofr | voip}**
4. **voice-class sip**
5. **sip**
6. **outbound-proxy** {**ipv4**:*ip-address*[:*port-number*] | *dns*:*host*:*domain*}

7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **{pots | vofr | voip}** **Example:** `Router(conf)# dial-peer voice 111 voip`	Define a particular dial peer, specifies the method of voice encapsulation, and enters dial-peer configuration mode.
Step 4	**voice-class sip** **Example:** `Router(config-dial-peer)# voice service voip`	Enters dial-peer VoIP configuration mode.
Step 5	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 6	**outbound-proxy** {**ipv4**:*ip-address*[:*port-number*] | *dns*:*host*:*domain*} **Example:** `Router(conf-serv-sip)# outbound-proxy ipv4:10.1.1.1`	Configures an outbound proxy server. This example shows how to configure an outbound-proxy server to IP address 10.1.1.1.
Step 7	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Configuring SIP Support for PAI

This section provides procedures for configuring the following supplementary services:

Configuring a Privacy Header

To configure a privacy header in support of RFC 3323, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **privacy** {**pstn** | *privacy-option* [**critical**]}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-ser)# sip`	Enters SIP configuration mode.
Step 5	**privacy** {**pstn** | *privacy-option* [**critical**]} **Example:** `Router(conf-ser-sip)# privacy pstn`	Configures a privacy header set to a value supported by RFC 3323. In this example, the privacy information from the PSTN side of a call is passed on to the VoIP side. The PSTN information is passed in the Octet 3a of the CALLING PARTY Information Element.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Configuring a Privacy Level

To configure a privacy header level for PAI or PPI, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **asserted-id [pai | ppi]**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-ser)# sip`	Enters SIP configuration mode.
Step 5	**asserted-id [pai | ppi]** **Example:** `Router(conf-ser-sip)# asserted-id ppi`	Configures a privacy header. In this example, a P-Preferred-Identity header is configured.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Configuring a Name and Number in the asserted-id Header

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **voice service voip**
4. **sip**
5. **caller-info pstn-to-sip {unscreened discard | {from | remote-party-id | asserted-id] {name set name | number set number}}}**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-ser)# sip`	Enters SIP configuration mode.
Step 5	**caller-info pstn-to-sip {unscreened discard | {from | remote-party-id | asserted-id] {name set name | number set number}}}** **Example:** `Router(conf-ser-sip)# caller-info pstn-to-sip asserted-id name set example`	Configures a name that is populated in the asserted-id field.
Step 6	**exit** **Example:** `Router (conf-ser-sip)# exit`	Exits the current mode.

Configuring SIP History-info Header Support

You can configure SIP History-info Header Support in two different configuration modes: voice service sip configuration (global level) and dial peer voice configuration (dial-peer level) mode. The gateway first checks if support is configured under the dial peer voice configuration mode, and failing a match, then defaults to the voice service configuration. If support is not configured in either mode, then the header is not added.

Configuring SIP History-info Header Support Globally

Perform this task to configure history-info header support at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **history-info**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**history-info** **Example:** `Router(conf-serv-sip)# history-info`	Configures history-info header support globally.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIP History-info Header Support at the Dial-Peer Level

Perform this task to configure history-info header support at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. voice-class sip history-info
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	voice-class sip history-info **Example:** `Router(config-dial-peer)# voice-class sip history-info`	Configures history-info header support for a dial peer.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Call Routing on SIP History-info Header Support Globally

Perform this task to configure call routing on history-info header at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **call route history-info**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**call route history-info** **Example:** `Router(conf-serv-sip)# call-route history-info`	Configures call-route history-info header support globally.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Call Routing on SIP History-info Header Support at the Dial-Peer Level

Perform this task to configure call routing on history-info header support at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. voice-class sip call-route history-info
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	voice-class sip call-route history-info **Example:** `Router(config-dial-peer)# voice-class sip call-route history-info`	Configures call-route history-info header support for a dial peer.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring SIP Trunk Registration

The IOS Gateway registers all its POTS dial peers to the registrar when the registrar is configured on the Gateway. The gateway now supports registering the number configured in the command. With the introduction of trunk registration support, registration of a single number would represent the SIP trunk. SIP trunk registration can be associated with multiple dial peers for routing outbound calls. On successful registration, all subsequent registration refresh and outbound calls will use the same outbound proxy IP address and port used for the registration.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. Do one of the following:
   - **voice service voip**
   - **dial-peer voice**
4. **sip**
5. **outbound-proxy dns:host:domain reuse**
6. Do one of the following:
   - **preloaded-route sip-server service-route**
   - **voice-class sip preloaded-route sip-server service-route**
7. **exit**
8. **exit**
9. **dial-peer voice** *tag* **pots**
10. **no sip-register**
11. **exit**
12. **sip-ua**
13. **credentials number** number **username** username **password** password **realm** realm
14. **registrar** registrar-server-address[:port] **auth-realm** realm
15. **exit**
16. Do one of the following:
    - **voice service voip**
    - **dial-peer voice**
17. **sip**
18. Do one of the following:
    - **associate registered-number** *number*
    - **voice-class sip associate registered-number** number
19. Do one of the following:
    - **call-route p-called-party-id**
    - or
    - **voice-class sip call-route p-called-party-id**
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	Do one of the following: • **voice service voip** • **dial-peer voice** **Example:** `Router(config)# voice service voip` **Example:** `Router(config)# dial-peer voice`	Enters voice service VoIP configuration mode or dial peer configuration mode.
Step 4	**sip** **Example:** `Router(config-voi-serv)# sip`	Enters voice service VoIP SIP configuration mode.
Step 5	**outbound-proxy dns:host:domain reuse** **Example:** `Router(config-serv-sip)# outbound-proxy dns:obp.twc.com reuse`	Defines the outbound proxy information with the reuse option.
Step 6	Do one of the following: • **preloaded-route sip-server service-route** • **voice-class sip preloaded-route sip-server service-route** **Example:** `Router(config-serv-sip)# preloaded-route sip-server service-route` **Example:** `Router(config-dial-peer)# voice-class sip preloaded-route sip-server service-route`	Use this command under global or dial-peer configuration mode to add service route and outbound proxy information after successful trunk registration.
Step 7	**exit** **Example:** `Router(config-serv-sip)# exit`	Exits voice service VOIP SIP configuration mode and returns to the voice service VOIP mode.
Step 8	**exit** **Example:** `Router(config-voi-serv)# exit`	Exits voice service VOIP configuration mode and returns to global configuratiion mode.

	Command or Action	Purpose
Step 9	**dial-peer voice** *tag* **pots** **Example:** `Router(config)# dial-peer voice 100 pots`	Enters dial-peer voice configuration mode. **Note** Voice service configuration mode configures the gateway to support the Accept-Language header in both outgoing SIP INVITE messages and OPTIONS responses. Dial-peer voice configuration mode configures it to support the header in INVITE messages only.
Step 10	**no sip-register** **Example:** `Router(config-dial-peer)# no sip-register`	Use this command under each dial peer to limit the gateway to register only one number.
Step 11	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits dial-peer configuration mode and enters the global configuration mode.
Step 12	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 13	**credentials number** number **username** username **password** password **realm** realm **Example:** `Router(config-sip-ua)# credentials number 1234 username abc password pass realm cisco.com`	Configures a single number registration. Deregistration of the number when the POTS interface goes down will be triggered for the POTS dial peer only when the dial peer is already registered. This is not the case for the POTS interface.
Step 14	**registrar** registrar-server-address[:port] **auth-realm** realm **Example:** `Router(config-sip-ua)# registrar ipv4:209.165.201.2 auth-realm cisco.com`	Associates a protection domain with the registrar using the auth-realm option. Configure the authentication for the number with the same realm.
Step 15	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits SIP user-agent configuration mode and enters the global configuration mode.
Step 16	Do one of the following: • **voice service voip** • **dial-peer voice** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode or dial-peer configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# dial-peer voice`	
Step 17	**sip** **Example:** `Router(config-voi-serv)# sip`	Enters voice service VoIP SIP configuration mode.
Step 18	Do one of the following: • **associate registered-number** *number* • **voice-class sip associate registered-number** number **Example:** `Router(config-voi-serv-sip)# associate registered-number 1234` **Example:** `Router(config-dial-peer)# voice-class sip associate registered-number 1234`	Associates a registered number under a global voice service configuration level or dial peer. This is required for trunk registration to obtain the service route and outbound proxy information used for last registration.
Step 19	Do one of the following: • **call-route p-called-party-id** • or • **voice-class sip call-route p-called-party-id** **Example:** `Router(config-voi-serv-sip)# call-route p-called-party-id` **Example:** `Router(config-dial-peer)# voice-class sip call-route p-called-party-id`	Routes the call based on the p-called party-id header in the incoming INVITE at the global voice service configurationl level or dial peer level.
Step 20	**end** **Example:** `Router(config-serv-sip)# exit`	Exits voice service VoIP SIP configuration model and returns to privileged exec mode.

Configuring Support for SIP 181 Call is Being Forwarded Message

You can configure support for SIP 181 Call is Being Forwarded messages either globally or on a specific dial-peer. Use the **block** command in voice service SIP configuration mode to globally configure Cisco IOS voice gateways and Cisco UBEs to drop specified SIP provisional response messages. To configure settings for an individual dial peer, use the **voice-class sip block** command in dial peer voice configuration mode.

Both globally and at the dial peer level, you can also use the **sdp** keyword to further control when the specified SIP message is dropped based on either the absence or presence of SDP information.

Additionally, you can use commands introduced for this feature to configure a Cisco UBE, either globally or at the dial peer level, to map specific received SIP provisional response messages to a different SIP provisional response message on the outgoing SIP dial peer. To do so, use the **map resp-code** command in voice service SIP configuration mode for global configuration or, to configure a specific dial peer, use the **voice-class sip map resp-code** in dial peer voice configuration mode.

Configuring Support for SIP 181 Call is Being Forwarded Message Globally

Perform this task to configure support for SIP 181 messages at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**block** {**180**	**181**	**183**} [**sdp** {**absent**	**present**}] **Example:**	Configures support of SIP 181 messages globally so that messages are passed as is. The sdp keyword is optional and allows for dropping or passing of SIP 181 messages based on the presence or absence of SDP.

	Command or Action	Purpose
	`Router(conf-serv-sip)# block 181 sdp present`	
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Support for SIP 181 Call is Being Forwarded Message at the Dial-Peer Level

Perform this task to configure support for SIP 181 messages at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. voice-class sip **block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	voice-class sip **block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}] **Example:** `Router(config-dial-peer)# voice-class sip block 181 sdp present`	Configures support of SIP 181 messages on a specific dial peer so that messages are passed as is. The sdp keyword is optional and allows for dropping or passing of SIP 181 messages based on the presence or absence of SDP.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Mapping of SIP Provisional Response Messages Globally

Perform this task to configure mapping of specific received SIP provisional response messages at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **map resp-code 181 to 183**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**map resp-code 181 to 183** **Example:** `Router(conf-serv-sip)# map resp-code 181 to 183`	Enables mapping globally of received SIP messages of a specified message type to a different SIP message type.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Mapping of SIP Provisional Response Messages at the Dial-Peer Level

Perform this task to configure mapping of received SIP provisional response messages at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. voice-class sip **map resp-code 181 to 183**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	voice-class sip **map resp-code 181 to 183** **Example:** `Router(config-dial-peer)# voice-class sip map resp-code 181 to 183`	Enables mapping of received SIP messages of a specified SIP message type on a specific dial peer to a different SIP message type.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

You can configure support for resetting the Expires timer upon receipt or sending of SIP 183 messages to reset the Expires timer and preserve the call until final response. You can enable this feature globally, using the **reset timer expires** command in voice service SIP configuration mode, or on a specific dial-peer using the **voice-class sip reset timer expires** command in dial peer voice configuration mode.

Configuring Reset of Expires Timer Globally

Perform this task to enable resetting of the Expires timer at the global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **reset timer expires 183**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**reset timer expires 183** **Example:**	Enables resetting of the Expires timer on receiving SIP 183 messages globally.

	Command or Action	Purpose
	`Router(conf-serv-sip)# reset timer expires 183`	
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Reset of Expires Timer at the Dial-Peer Level

Perform this task to enable resetting of the Expires timer at the dial-peer level in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip reset timer expires 183**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	**voice-class sip reset timer expires 183** **Example:** `Router(config-dial-peer)# voice-class sip reset timer expires 183`	Enables resetting of the Expires timer on receiving SIP 183 messages on a specific dial peer.
Step 5	**exit** **Example:**	Exits the current mode.

	Command or Action	Purpose
	`Router(config-dial-peer)# exit`	

Configuring Support for Stripping off Progress Indication

Perform this task to configure support for stripping off PI from incoming ISDN messages on a Cisco IOS SIP or H.323 TDM voice gateway or on a Cisco UBE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **pots**
4. **destination-pattern** [+]*string*[**T**]
5. **progress_ind** {{**alert** | **callproc**} {**enable** *pi-number* | **disable** | **strip** [*strip-pi-number*]} | {**connect** | **disconnect** | **progress** | **setup**} {**enable** *pi-number* | **disable**}}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **pots** **Example:** `Router(config)# dial-peer voice 3 pots`	Enters dial peer POTS configuration mode.
Step 4	**destination-pattern** [+]*string*[**T**] **Example:** `Router(config-dial-peer)# destination-pattern 555`	Specifies either the prefix or the full E.164 telephone number to be used for a dial peer.
Step 5	**progress_ind** {{**alert**	**callproc**} {**enable** *pi-number*	**disable**	**strip** [*strip-pi-number*]}	{**connect**	**disconnect**	**progress**	**setup**} {**enable** *pi-number*	**disable**}} **Example:**	Configure an outbound dial peer to override and remove or replace the default PI in specified call message types.

	Command or Action	Purpose
	`Router(config-dial-peer)# progress_ind callproc strip 8`	
Step 6	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Verifying SIP Message Components Session Timers and Responses Configuration

To verify SIP message components, session timers, and responses configuration, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show call active voice**
2. **show call application sessions**
3. **show call history**
4. **show logging**
5. **show running-config**
6. **show sip min-se**
7. **show sip-ua map pstn-sip**
8. **show sip-ua map sip-pstn**
9. **show sip-ua statistics**
10. **show sip-ua status**
11. **show sip-ua timers**
12. **show subscription** {**asnl session** {**active** | **history** [**errors** | **session-id** *session-id* | **url**] | **statistics**} | **sip**} [**summary**]

DETAILED STEPS

Step 1 show call active voice

Use this command to display call information for voice calls in progress.

Note For sample output, see "Monitor the Hostname in Locally Generated SIP Headers".

Step 2 show call application sessions

Use this command to view whether the application is running.

Example:

```
Router# show call application sessions
TCL Sessions
    There are 1 active TCL sessions
```

```
     SID  Name        Called     Calling                App Name               Legs
       1                 50276  50280                  testapp31             4
VXML Sessions
    No running VXML sessions
```

Step 3 **show call history**

Use this command, optionally with the **voice** keyword, to display the call history table for voice calls.

Example:

```
Router# show call history
DisconnectCause=10
DisconnectText=normal call clearing
.
.
.
```

Note For more sample output, see "Monitor the Hostname in Locally Generated SIP Headers".

Step 4 **show logging**

Use this command to display the state of logging (syslog).

The following partial sample output shows that the outgoing gateway is receiving a 180 message with SDP and is configured to ignore the SDP.

Example:

```
Router# show logging
Log Buffer (600000 bytes):
00:12:19:%SYS-5-CONFIG_I:Configured from console by console
00:12:19:%SYS-5-CONFIG_I:Configured from console by console
00:12:20:0x639F6EEC :State change from (STATE_NONE, SUBSTATE_NONE)  to
(STATE_IDLE, SUBSTATE_NONE)
00:12:20:****Adding to UAC table
00:12:20:adding call id 2 to table
00:12:20: Queued event from SIP SPI :SIPSPI_EV_CC_CALL_SETUP
00:12:20:CCSIP-SPI-CONTROL: act_idle_call_setup
00:12:20: act_idle_call_setup:Not using Voice Class Codec
00:12:20:act_idle_call_setup:preferred_codec set[0] type :g711ulaw
bytes:160
00:12:20:sipSPICopyPeerDataToCCB:From CLI:Modem NSE payload = 100,
Passthrough = 0,Modem relay = 0, Gw-Xid = 1
SPRT latency 200, SPRT Retries = 12, Dict Size = 1024
String Len = 32, Compress dir = 3
00:12:20:sipSPICanSetFallbackFlag - Local Fallback is not active
00:12:20:****Deleting from UAC table
00:12:20:****Adding to UAC table
00:12:20: Queued event from SIP SPI :SIPSPI_EV_CREATE_CONNECTION
00:12:20:0x639F6EEC :State change from (STATE_IDLE, SUBSTATE_NONE)  to
(STATE_IDLE, SUBSTATE_CONNECTING)
00:12:20:0x639F6EEC :State change from (STATE_IDLE,
SUBSTATE_CONNECTING)  to (STATE_IDLE, SUBSTATE_CONNECTING)
00:12:20:sipSPIUsetBillingProfile:sipCallId for billing records =
41585FCE-14F011CC-8005AF80-D4AA3153@10.1.1.42
00:12:20:CCSIP-SPI-CONTROL: act_idle_connection_created
00:12:20:CCSIP-SPI-CONTROL: act_idle_connection_created:Connid(1)
created to 10.1.1.15:5060, local_port 57838
00:12:20:CCSIP-SPI-CONTROL: sipSPIOutgoingCallSDP
00:12:20:sipSPISetMediaSrcAddr: media src addr for stream 1 = 10.1.1.42
00:12:20:sipSPIReserveRtpPort:reserved port 18978 for stream 1
00:12:20: convert_codec_bytes_to_ptime:Values :Codec:g711ulaw
```

```
codecbytes :160, ptime:20
00:12:20:sip_generate_sdp_xcaps_list:Modem Relay disabled. X-cap not needed
00:12:20:Received Octet3A=0x00 -> Setting ;screen=no ;privacy=off
00:12:20:sipSPIAddLocalContact
00:12:20: Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE
00:12:20:sip_stats_method
00:12:20:sipSPIProcessRtpSessions
00:12:20:sipSPIAddStream:Adding stream 1 (callid 2) to the VOIP RTP library
00:12:20:sipSPISetMediaSrcAddr: media src addr for stream 1 = 10.1.1.42
00:12:20:sipSPIUpdateRtcpSession:for m-line 1
00:12:20:sipSPIUpdateRtcpSession:rtcp_session info
laddr = 10.1.1.42, lport = 18978, raddr = 0.0.0.0,
rport=0, do_rtcp=FALSE
src_callid = 2, dest_callid = -1
00:12:20:sipSPIUpdateRtcpSession:No rtp session, creating a new one
00:12:20:sipSPIAddStream:In State Idle
00:12:20:act_idle_connection_created:Transaction active. Facilities will be queued.
00:12:20:0x639F6EEC :State change from (STATE_IDLE,
SUBSTATE_CONNECTING)  to (STATE_SENT_INVITE, SUBSTATE_NONE)
00:12:20:Sent:
INVITE sip:222@10.1.1.15:5060 SIP/2.0
Via:SIP/2.0/UDP  10.1.1.42:5060
From:"111" <sip:111@10.1.1.42>;tag=B4DC4-9E1
To:<sip:222@10.1.1.15>
Date:Mon, 01 Mar 1993 00:12:20 GMT
Call-ID:41585FCE-14F011CC-8005AF80-D4AA3153@10.1.1.42
Supported:timer
Min-SE: 1800
Cisco-Guid:1096070726-351277516-2147659648-3567923539
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE,
NOTIFY, INFO
CSeq:101 INVITE
Max-Forwards:6
Remote-Party-ID:<sip:111@10.1.1.42>;party=calling;screen=no;privacy=off
Timestamp:730944740
Contact:<sip:111@10.1.1.42:5060>
Expires:180
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:230
v=0
o=CiscoSystemsSIP-GW-UserAgent 4629 354 IN IP4 10.1.1.42
s=SIP Call
c=IN IP4 10.1.1.42
t=0 0
m=audio 18978 RTP/AVP 0 100
c=IN IP4 10.1.1.42
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
00:12:21:Received:
SIP/2.0 100 Trying
Via:SIP/2.0/UDP  10.1.1.42:5060
From:"111" <sip:111@10.1.1.42>;tag=B4DC4-9E1
To:<sip:222@10.1.1.15>;tag=442AC-22
Date:Wed, 16 Feb 2000 18:19:56 GMT
Call-ID:41585FCE-14F011CC-8005AF80-D4AA3153@10.1.1.42
Timestamp:730944740
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow-Events:telephone-event
Content-Length:0
```

```
00:12:21:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
10.1.1.15:5060
00:12:21:CCSIP-SPI-CONTROL: act_sentinvite_new_message
00:12:21:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:12:21:sip_stats_status_code
00:12:21: Roundtrip delay 420 milliseconds for method INVITE
00:12:21:0x639F6EEC :State change from (STATE_SENT_INVITE,
SUBSTATE_NONE)  to (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
00:12:21:Received:
SIP/2.0 180 Ringing
Via:SIP/2.0/UDP  10.1.1.42:5060
From:"111" <sip:111@10.1.1.42>;tag=B4DC4-9E1
To:<sip:222@10.1.1.15>;tag=442AC-22
Date:Wed, 16 Feb 2000 18:19:56 GMT
Call-ID:41585FCE-14F011CC-8005AF80-D4AA3153@10.1.1.42
Timestamp:730944740
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow-Events:telephone-event
Contact:<sip:222@192.0.2.59:5060>
Record-Route:<sip:222@10.1.1.15:5060;maddr=10.1.1.15>
Content-Length:230
Content-Type:application/sdp
v=0
o=CiscoSystemsSIP-GW-UserAgent 4629 354 IN IP4 10.1.1.42
s=SIP Call
c=IN IP4 10.1.1.42
t=0 0
m=audio 18978 RTP/AVP 0 100
c=IN IP4 10.1.1.42
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
00:12:21:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
10.1.1.15:5060
00:12:21:CCSIP-SPI-CONTROL: act_recdproc_new_message
00:12:21:CCSIP-SPI-CONTROL: act_recdproc_new_message_response
00:12:21:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:12:21:sip_stats_status_code
00:12:21: Roundtrip delay 496 milliseconds for method INVITE
00:12:21:CCSIP-SPI-CONTROL: act_recdproc_new_message_response :Early
media disabled for 180:Ignoring SDP if present
00:12:21:HandleSIP1xxRinging:SDP in 180 will be ignored if present: No
early media cut through
00:12:21:HandleSIP1xxRinging:SDP Body either absent or ignored in 180
RINGING:- would wait for 200 OK to do negotiation.
00:12:21:HandleSIP1xxRinging:MediaNegotiation expected in 200 OK
00:12:21:sipSPIGetGtdBody:No valid GTD body found.
00:12:21:sipSPICreateRawMsg:No GTD passed.
00:12:21:0x639F6EEC :State change from (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_PROCEEDING)  to (STATE_RECD_PROCEEDING,
SUBSTATE_PROCEEDING_ALERTING)
00:12:21:HandleSIP1xxRinging:Transaction Complete. Lock on Facilities
released.
00:12:22:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  10.1.1.42:5060
From:"111" <sip:111@10.1.1.42>;tag=B4DC4-9E1
To:<sip:222@10.1.1.15>;tag=442AC-22
Date:Wed, 16 Feb 2000 18:19:56 GMT
Call-ID:41585FCE-14F011CC-8005AF80-D4AA3153@10.1.1.42
Timestamp:730944740
Server:Cisco-SIPGateway/IOS-12.x
```

```
CSeq:101 INVITE
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE,
NOTIFY, INFO
Allow-Events:telephone-event
Contact:<sip:222@10.1.1.59:5060>
Record-Route:<sip:222@10.1.1.15:5060;maddr=10.1.1.15>
Content-Type:application/sdp
Content-Length:231
v=0
o=CiscoSystemsSIP-GW-UserAgent 9600 4816 IN IP4 10.1.1.59
s=SIP Call
c=IN IP4 10.1.1.59
t=0 0
m=audio 19174 RTP/AVP 0 100
c=IN IP4 10.1.1.59
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
```

Step 5 **show running-config**

Use this command to display the contents of the currently running configuration file or the configuration for a specific interface. Use it to display the current configuration and to verify header passing and subscription configuration.

Note If early media (the default setting) is enabled, this command does not show any information related to the feature.

The following sample output shows that the disable-early-media 180 command was used.

Example:

```
Router# show running-config
.
.
.
dial-peer voice 223 pots
 application session
 destination-pattern 223
 port 1/0/0
!
gateway
!
sip-ua
 disable-early-media 180
```

Step 6 **show sip min-se**

Use this command to show the current value of a minimum-session-expiration header for calls that use SIP.

Example:

```
Router# show sip min-se
SIP UA MIN-SE Value (seconds)
Min-SE: 90
```

Step 7 **show sip-ua map pstn-sip**

Use this command to display the mapping table of PSTN cause codes and their corresponding SIP error status codes or the mapping table of PSTN-to-SIP codes.

Example:

```
Router# show sip-ua map pstn-sip
PSTN-Cause   Configured        Default
             SIP-Status       SIP-Status
1               404             404
2               404             404
3               404             404
4               500             500
.
.
.
110             500             500
111             400             400
126             500             500
127             500             500
```

Step 8 **show sip-ua map sip-pstn**

Use this command to display the mapping table of PSTN cause codes and their corresponding SIP error status codes or the mapping table of SIP-to-PSTN codes.

Example:

```
Router# show sip-ua map sip-pstn
SIP-Status   Configured        Default
             PSTN-Cause      PSTN-Cause
400             127             127
401             57              57
402             21              21
403             57              57
.
.
.
600             17              17
603             21              21
604             1               1
606             58              58
```

Step 9 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics, including the Bad Request counter. Use it to verify configuration of the SIP INVITE Request with Malformed Via Header feature, which increments a counter (shown as *Client Error: Bad Request*) when a malformed Via header is received.

Note To reset counters after you view statistics, use the **clear sip-ua statistics** command.

The following sample output shows response, traffic, and retry SIP statistics, including the Bad Request counter. Use it to verify configuration of the SIP INVITE Request with Malformed Via Header feature, which increments a counter (shown as *Client Error: Bad Request*) when a malformed Via header is received.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound
)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
```

```
      OkPrack 0/0, OkPreconditionMet 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, SeeOther 0,
      UseProxy 0, AlternateService 0
    Client Error:
      BadRequest
 0/0
, 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,
      RequestCancel 0/0, NotAcceptableMedia 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0
Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0
```

Step 10 **show sip-ua status**

Use this command to display status for the SIP user agent.

The following sample output shows status for the SIP user agent after the **disable-early-media 180** command was used.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP :ENABLED
SIP User Agent for TCP :ENABLED
SIP User Agent bind status(signaling):ENABLED 10.0.0.0
SIP User Agent bind status(media):ENABLED 0.0.0.0
SIP early-media for 180 responses with SDP:DISABLED
SIP max-forwards :6
SIP DNS SRV version:2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP:NONE
Check media source packets:DISABLED
Redirection (3xx) message handling:ENABLED
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
```

Step 11 **show sip-ua timers**

Use this command to display all SIP UA information.

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 150000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500, refer 500,
hold 2880 minutes, buffer-invite 500
```

Step 12 **show subscription** {**asnl session** {**active** | **history** [**errors** | **session-id** *session-id* | **url**] | **statistics**} | **sip**} [**summary**]

Use this command to display information about Application SUBSCRIBE/NOTIFY Layer (ASNL)-based and non-ASNL-based SIP subscriptions.

Example:

```
Router# show subscription asnl session history
ASNL Subscription History Records Details:
==========================================
Total history records                           = 1
Total error count                               = 0
Total subscription requests sent                = 1
Total subscription requests received            = 0
Total notification requests sent                = 0
Total notification requests received            = 3
URL: sip:user@10.7.104.88
  Event Name : stress
  Session ID : 8
  Expiration Time : 50 seconds
  Subscription Duration : 10 seconds
  Protocol : ASNL_PROTO_SIP
  Remote IP address : 10.7.104.88
  Port : 5060
  Call ID : 5
  Total Subscriptions Sent : 1
  Total Subscriptions Received: 0
  Total Notifications Sent : 0
  Total Notifications Received : 3
  Last response code                 : ASNL_UNSUBSCRIBE_SUCCESS
  Last error code                        : ASNL_NONE
  First Subscription Time : 10:55:12 UTC Apr 9 2000
  Last Subscription Time : 10:55:12 UTC Apr 9 2000
  First Notify Time : 10:55:12 UTC Apr 9 2000
  Last Notify Time : 10:55:22 UTC Apr 9 2000
Router# show subscription asnl session history summary
ASNL Subscription History Records Summary:
==========================================
Total history records = 2
Total error count = 0
Total subscription requests sent = 2
Total subscription requests received = 0
Total notification requests sent = 0
Total notification requests received = 6
URL                                             Session ID  Call ID
---              ----------  -------
sip:user@10.7.104.88                             9           5
sip:user@10.7.104.88                             8           5
```

The following sample output shows the error type ASNL_SUBSCRIBE_FAILED. This error indicates that the subscription request has failed.

Example:

```
Router# show subscription asnl session history summary
ASNL Subscription History Records Summary:
==========================================
Total history records = 8
Total error count = 6
  Total error type (ASNL_SUBSCRIBE_FAILED) = 6
Total subscription requests sent = 8
Total subscription requests received = 0
Total notification requests sent = 0
Total notification requests received = 6
URL                                                Session ID  Call ID
---              ----------  -------
sip:user@10.7.104.88                                15          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                14          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                13          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                12          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                11          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                10          N/A
    ASNL_SUBSCRIBE_FAILED
sip:user@10.7.104.88                                9           5
sip:user@10.7.104.88                                8           5
Router# show subscription asnl session history error
ASNL Subscription History Error Statistics:
===========================================
Total history records             = 8
Total history records with errors = 6
URL       : sip:user@10.7.104.88
Session ID: 15
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
URL       : sip:user@10.7.104.88
Session ID: 14
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
URL       : sip:user@10.7.104.88
Session ID: 13
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
URL       : sip:user@10.7.104.88
Session ID: 12
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
URL       : sip:user@10.7.104.88
Session ID: 11
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
URL       : sip:user@10.7.104.88
Session ID: 10
```

```
Call ID   : N/A
Event Name: newstress
Error     : ASNL_SUBSCRIBE_FAILED
Router# show subscription asnl session history url
ASNL Subscription History URL Records Details:
==============================================
Total history records = 3
Total history records with errors = 0
Total number of different URLs = 1
Total number of different events = 2
Total subscription requests sent = 3
Total subscription requests received = 0
Total notification requests sent = 0
Total notification requests received = 9
URL: sip:user@10.7.104.88
  Event Name: stress1
  Session ID: 19         Call ID: N/A
  Event Name: stress
  Session ID: 18         Call ID: 5
  Event Name: newstress
  Session ID: 17         Call ID: N/A
  Total error count for this URL                          = 0
  Total events subscribed by this URL                     = 0
  Total subscription requests sent for this URL           = 3
  Total subscription requests received for this URL       = 0
  Total notification requests sent for this URL           = 0
  Total notification requests received for this URL       = 9
Router# show subscription asnl session history url summary
ASNL Subscription History URL Records Summary:
==============================================
Total history records = 3
Total history records with errors = 0
Total number of different URLs = 1
Total number of different events = 2
Total subscription requests sent = 3
Total subscription requests received = 0
Total notification requests sent = 0
Total notification requests received = 9
Router# show subscription asnl session statistics
ASNL Subscription and Notification Statistics:
==============================================
Total subscription requests sent = 3
```

Troubleshooting Tips for SIP Message Timer and Response Features

Note For general troubleshooting tips and a list of important **debug** commands, see the “General Troubleshooting Tips” section of the “Basic SIP Configuration” module.

- Make sure that you can make a voice call.
- Use the **debug asnl events**command to verify that the SIP subscription server is up. For example, the output displays a pending message if the client is unsuccessful in communicating with the server:
- If this is an H.323 gateway, use the **debug cch323**family of commands to enable H.323 debugging capabilities.

- If this is a SIP gateway, use the **debug ccsip**family of commands to enable SIP debugging capabilities. Use the **debug ccsip all**command to view all the SIP messages to trace a call.
- Use the **debug isdn q931**command to display information about call setup and tear down of ISDN network connections (layer 3) between the local router (user side) and the network.
- Use the debug radius command to display information associated with RADIUS.
- Use the **debug voip ccapi protoheaders**command to view messages sent between the originating and terminating gateways. If no headers are being received by the terminating gateway, verify that the **header-passing** command is enabled on the originating gateway.
- Use the **debug voip ivr script**command to display any errors that might occur when the Tcl script is run.

Following is sample output for some of these commands:

Sample Output for the debug asn1 events Command

```
Router# debug asnl events
*Mar  1 02:48:48.831: //-1//ASNL:SUB7:/asnl_subscribe: resp = ASNL_SUBSCRIBE_PENDING[2]
```

Sample Output for the debug ccsip all Command: Originating Gateway

```
Router# debug ccsip all
*Mar  1 01:45:53.783: Sent:
INVITE sip:debbie@example.com:5060 SIP/2.0
Via: SIP/2.0/UDP  10.1.1.109:5060
From: sip:nobody;tag=60F374-1061
To: sip:debbie@example.com
Date: Mon, 01 Mar 1993 01:45:53 GMT
Call-ID: 52F25057-14FD11CC-802B86FA-EE2DDC42@10.1.1.109
Subject: HelloSipTCL
AccountInfo: 123123
Priority: Urgent
testID: AL_FEAT_SIP_URL_O_RV_11
Supported: timer,100rel
Min-SE:  1800
Cisco-Guid: 1145332256-352129484-2150139642-3995982914
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE, NOTIFY, INFO
CSeq: 101 INVITE
Max-Forwards: 6
Remote-Party-ID: <sip:50006@10.1.1.109>;party=calling;screen=no;privacy=off
Timestamp: 730950353
Contact: <sip:50006@10.1.1.109:5060>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 234
v=0
o=CiscoSystemsSIP-GW-UserAgent 2819 5222 IN IP4 10.1.1.109
s=SIP Call
c=IN IP4 10.1.1.109
t=0 0
m=audio 16488 RTP/AVP 0 100
c=IN IP4 10.1.1.109
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
```

```
a=fmtp:100 192-194
a=ptime:20
```

Sample Output for the debug ccsip all Command: Terminating Gateway

```
*Jan 26 00:15:39.250: Received:
INVITE sip:debbie@example.com:5060 SIP/2.0
Via: SIP/2.0/UDP  10.1.1.109:5060
From: sip:nobody;tag=60F374-1061
To: sip:debbie@example.com
Date: Mon, 01 Mar 1993 01:45:53 GMT
Call-ID: 52F25057-14FD11CC-802B86FA-EE2DDC42@10.1.1.109
Subject: HelloSipTCL
AccountInfo: 123123
Priority: Urgent
testID: AL_FEAT_SIP_URL_O_RV_11
Supported: timer,100rel
Min-SE:  1800
Cisco-Guid: 1145332256-352129484-2150139642-3995982914
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, Refer , SUBSCRIBE, NOTIFY, INFO
CSeq: 101 INVITE
Max-Forwards: 6
Remote-Party-ID: <sip:50006@10.1.1.109>;party=calling;screen=no;privacy=off
Timestamp: 730950353
Contact: <sip:50006@10.1.1.109:5060>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 234
v=0
o=CiscoSystemsSIP-GW-UserAgent 2819 5222 IN IP4 10.1.1.109
s=SIP Call
c=IN IP4 10.1.1.109
t=0 0
m=audio 16488 RTP/AVP 0 100
c=IN IP4 10.1.1.109
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
```

Sample Output for the debug ccsip all Command: SIP Trunk Registration

```
Sent:
REGISTER sip:9.13.40.83:5099 SIP/2.0
Via: SIP/2.0/UDP 9.44.48.152:5060;branch=z9hG4bK1E19829
From: <sip:1234@9.13.40.83>;tag=187A1230-B2B
To: <sip:1234@9.13.40.83>
Call-ID: A3EC4165-192D11DF-8037BEFF-D94B81E0
Max-Forwards: 70
CSeq: 2 REGISTER
Contact: <sip:1234@9.44.48.152:5060>
Expires: 180
Supported: path
Authorization: Digest
username="test",realm="cisco.com",uri="sip:9.13.40.83:5099",response="",nonce=""
Content-Length: 0
```

Sample Output for the debug ccsip messages Command

The following shows sample output for one side of a call.

```
Router# debug ccsip messages
SIP Call messages tracing is enabled
Router#
*Mar 6 14:19:14: Sent:
INVITE sip:3660210@192.0.2.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Cisco-Guid: 2881152943-2184249568-0-483551624
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427554
Contact: <sip:3660110@192.0.2.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 192.0.2.230
s=SIP Call
t=0 0
c=IN IP4 192.0.2.230
m=audio 20762 RTP/AVP 0
*Mar 6 14:19:14: Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar 6 14:19:14: Received:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 192.0.2.231
s=SIP Call
t=0 0
c=IN IP4 192.0.2.231
m=audio 20224 RTP/AVP 0
*Mar 6 14:19:16: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Mon, 08 Mar 1993 22:45:12 GMT
```

```
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@192.0.2.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 192.0.2.231
s=SIP Call
t=0 0
c=IN IP4 192.0.2.231
m=audio 20224 RTP/AVP 0
*Mar 6 14:19:16: Sent:
ACK sip:3660210@192.0.2.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 138
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 192.0.2.230
s=SIP Call
t=0 0
c=IN IP4 192.0.2.230
m=audio 20762 RTP/AVP 0
*Mar 6 14:19:19: Received:
BYE sip:3660110@192.0.2.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.0.2.231:53600
From: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@192.0.2.230>
Date: Mon, 08 Mar 1993 22:45:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612717
CSeq: 101 BYE
Content-Length: 0
*Mar 6 14:19:19: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.231:53600
From: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@192.0.2.230>
Date: Sat, 06 Mar 1993 19:19:19 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612717
Content-Length:
CSeq: 101 BYE
```

The following shows sample output for the other side of the call.

```
Router# debug ccsip messages
SIP Call messages tracing is enabled
Router#
*Mar 8 17:45:12: Received:
INVITE sip:3660210@192.0.2.231;user=phone;phone-context=unknown SIP/2.0
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
```

```
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Cisco-Guid: 2881152943-2184249568-0-483551624
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731427554
Contact: <sip:3660110@192.0.2.230:5060;user=phone>
Expires: 180
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 192.0.2.230
s=SIP Call
t=0 0
c=IN IP4 192.0.2.230
m=audio 20762 RTP/AVP 0
*Mar 8 17:45:12: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Length: 0
*Mar 8 17:45:12: Sent:
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 192.0.2.231
s=SIP Call
t=0 0
c=IN IP4 192.0.2.231
m=audio 20224 RTP/AVP 0
*Mar 8 17:45:14: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Mon, 08 Mar 1993 22:45:12 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Timestamp: 731427554
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Contact: <sip:3660210@192.0.2.231:5060;user=phone>
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Length: 138
v=0
o=CiscoSystemsSIP-GW-UserAgent 1193 7927 IN IP4 192.0.2.231
s=SIP Call
t=0 0
c=IN IP4 192.0.2.231
```

```
m=audio 20224 RTP/AVP 0
*Mar 8 17:45:14: Received:
ACK sip:3660210@192.0.2.231:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.0.2.230:55820
From: "3660110" <sip:3660110@192.0.2.230>
To: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
Date: Sat, 06 Mar 1993 19:19:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Max-Forwards: 6
Content-Type: application/sdp
Content-Length: 138
CSeq: 101 ACK
v=0
o=CiscoSystemsSIP-GW-UserAgent 5596 7982 IN IP4 192.0.2.230
s=SIP Call
t=0 0
c=IN IP4 192.0.2.230
m=audio 20762 RTP/AVP 0
*Mar 8 17:45:17: Sent:
BYE sip:3660110@192.0.2.230:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 192.0.2.231:53600
From: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@192.0.2.230>
Date: Mon, 08 Mar 1993 22:45:14 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 6
Timestamp: 731612717
CSeq: 101 BYE
Content-Length: 0
*Mar 8 17:45:17: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.231:53600
From: <sip:3660210@192.0.2.231;user=phone;phone-context=unknown>;tag=27DBC6D8-1357
To: "3660110" <sip:3660110@192.0.2.230>
Date: Sat, 06 Mar 1993 19:19:19 GMT
Call-ID: ABBAE7AF-823100E2-0-1CD274BC@172.18.192.194
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Timestamp: 731612717
Content-Length: 0
CSeq: 101 BYE
```

Sample Output for the debug isdn q931 Command

The following shows sample output of a call-setup procedure for an outgoing call.

```
Router# debug isdn q931
Router# debug isdn q931
TX -> SETUP pd = 8 callref = 0x04
Bearer Capability i = 0x8890
Channel ID i = 0x83
Called Party Number i = 0x80, `415555121202'
RX <- CALL_PROC pd = 8 callref = 0x84
Channel ID i = 0x89
RX <- CONNECT pd = 8 callref = 0x84
TX -> CONNECT_ACK pd = 8 callref = 0x04....
Success rate is 0 percent (0/5)
```

The following shows sample output of a call-setup procedure for an incoming call.

```
Router# debug isdn q931
RX <- SETUP pd = 8 callref = 0x06
```

```
Bearer Capability i = 0x8890
Channel ID i = 0x89
Calling Party Number i = 0x0083, `81012345678902'
TX -> CONNECT pd = 8 callref = 0x86
RX <- CONNECT_ACK pd = 8 callref = 0x06
```

The following shows sample output of a call teardown procedure from the network.

```
Router# debug isdn q931
RX <- DISCONNECT pd = 8 callref = 0x84
Cause i = 0x8790
Looking Shift to Codeset 6
Codeset 6 IE 0x1 1 0x82 `10'
TX -> RELEASE pd = 8 callref = 0x04
Cause i = 0x8090
RX <- RELEASE_COMP pd = 8 callref = 0x84
```

The following shows sample output of a call teardown procedure from the router.

```
Router# debug isdn q931
TX -> DISCONNECT pd = 8 callref = 0x05
Cause i = 0x879081
RX <- RELEASE pd = 8 callref = 0x85
Looking Shift to Codeset 6
Codeset 6 IE 0x1 1 0x82 `10'
TX <- RELEASE_COMP pd = 8 callref = 0x05
```

Sample Output for the debug voip ccapi protoheaders Command: Originating Gateway

```
Router# debug voip ccapi protoheaders
voip ccAPI protocol headers/bodies passing info debugging is on
*Mar  1 01:23:14.711: //-1/xxxxxxxxxxxx/CCAPI/ccGetUriDataFromTDContainer: urlp=642D8EF0,
urlp->original_url=sip:debbie@example.com?Subject=Hello&Priority=Urgent&testID=AL_FEAT_SIP_URL_O_RV_11
*Mar  1 01:23:14.711: //-1/xxxxxxxxxxxx/CCAPI/ccSetupReqDataTDFreeHelper: data=6472C678
*Mar  1 01:23:25.155: //-1/xxxxxxxxxxxx/CCAPI/ccSetupReqDataTDFreeHelper: data=632FFD54
```

Sample Output for the debug voip ccapi protoheaders Command: Terminating Gateway

```
*Jan 25 23:53:00.102: //-1/xxxxxxxxxxxx/CCAPI/ccGetUriDataFromTDContainer: urlp=63CFFCD4,
urlp->original_url=sip:nobody;tag=4C3670-14E3
*Jan 25 23:53:00.102: //-1/xxxxxxxxxxxx/CCAPI/ccGetUriDataFromTDContainer: urlp=652DAF54,
urlp->original_url=sip:debbie@example.com:5060
*Jan 25 23:53:00.110: //-1/xxxxxxxxxxxx/CCAPI/ccGetUriDataFromTDContainer: urlp=63CFFCD4,
urlp->original_url=sip:nobody;tag=4C3670-14E3
*Jan 25 23:53:00.110: //-1/xxxxxxxxxxxx/CCAPI/ccGetUriDataFromTDContainer: urlp=652DAF54,
urlp->original_url=sip:debbie@example.com:5060
*Jan 25 23:53:00.122: //148/256F0CED801A/CCAPI/ccGetAvlistProtoHeader: tag=35,
reqHeader=64417738, reqCount=1, sess_protocol=SIP
```

Sample Output for the debug voip ivr script Command

In the following example, the script fails because the application that is specified in the notificationReceiver field in the script is not configured on the gateway with the **call application voice**command:

```
Router# debug voip ivr script
*Mar  1 02:44:24.927: //73//TCL2:/TclInterpDriver: Tcl_Eval Failed in action=act_Setup
code=1
code=ERROR
*Mar  1 02:44:24.927: IVR TCL script failure
```

```
        Result:
                          notifyRecr is not a configured application. Processing
subscriptionInfo array failed.
*Mar  1 02:44:24.927:   IVR TCL script failure errorInfo:
                         notifyRecr is not a configured application. Processing
subscriptionInfo array failed.
    while executing
"subscription open sip:anglee@sip-server1 subinfo"
    invoked from within
"set subscription_id [subscription open sip:anglee@sip-server1 subinfo]..."
    (procedure "subscribeService" line 49)
    invoked from within
"subscribeService"
    (procedure "act_Setup" line 44)
    invoked from within
"act_Setup"
```

Configuration Examples for SIP Message Timer and Response Features

Internal Cause Code Consistency Between SIP and H.323 Example

This example shows a H.323 and SIP configuration with the **cause-code legacy** command configured. The **cause-code legacy** command sets internal failures with nonstandard H.323 or SIP cause codes. The **cause-code legacy** command is generally used for backward compatibility purposes, as standard cause codes are used by default.

```
Router# show running-config
Building configuration...
Current configuration : 4271 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
enable password %#1$5dQA*@
!
voice-card 3
!
ip subnet-zero
!
ip domain-name example.com
ip name-server 10.100.0.40
!
isdn switch-type primary-net5
!
voice service voip
cause-code legacy
h323
call start slow
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
```

```
mta receive maximum-recipients 0
ccm-manager mgcp
!
controller E1 3/0
pri-group timeslots 1-31
!
controller E1 3/1
pri-group timeslots 1-31
!
interface FastEthernet0/0
ip address 10.102.0.33 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
ip address 10.101.0.33 255.255.255.0
duplex auto
speed auto
h323-gateway voip interface
h323-gateway voip id gatekeeper31 ipaddr 10.101.0.35 1718
h323-gateway voip h323-id gateway31
h323-gateway voip tech-prefix 1#
!
interface Serial3/0:15
no ip address
no logging event link-status
isdn switch-type primary-net5
isdn incoming-voice voice
no cdp enable
!
interface Serial3/1:15
no ip address
no logging event link-status
isdn switch-type primary-net5
isdn incoming-voice voice
no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.102.0.1
ip http server
ip pim bidir-enable
!
call rsvp-sync
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 3/0:15
!
voice-port 3/1:15
!
mgcp ip qos dscp cs5 media
mgcp ip qos dscp cs3 signaling
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 1001096 pots
destination-pattern 1001096
port 1/0/0
!
dial-peer voice 1001097 pots
```

```
destination-pattern 1001097
port 1/0/1
!
dial-peer voice 1003000 pots
destination-pattern 10030..
port 3/1:15
!
dial-peer voice 1003100 pots
destination-pattern 10031..
port 3/1:15
!
dial-peer voice 2000000 voip
destination-pattern 2......
session protocol sipv2
session target ipv4:10.101.0.40
!
dial-peer voice 3000000 voip
destination-pattern 3......
session protocol sipv2
session target sip-server
!
dial-peer voice 4000000 voip
destination-pattern 4......
session target ras
!
gateway
!
sip-ua
sip-server ipv4:10.100.0.40
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
session-timeout 20
password password
login
!
end
```

SIP - Configurable PSTN Cause Code Mapping Example

This examples shows the two commands that change the standard mappings between the SIP and PSTN networks. The **set sip-status** command and **set pstn-cause**command are highlighted in the following configuration.

```
Router# show running-config
Building configuration...
Current configuration : 1564 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3660-1
!
clock timezone GMT 0
voice-card 1
!
ip subnet-zero
```

```
!
ip domain-name example.sip.com
ip name-server 10.10.1.8
!
isdn switch-type primary-5ess
!
voice service voip
 sip
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
controller T1 1/0
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-wink-start
 ds0 busyout 2-24
!
controller T1 1/1
 framing sf
 linecode ami
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.10.1.3 255.255.255.0
 duplex auto
 speed auto
 ip rsvp bandwidth 75000 75000
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/1
ip http server
ip pim bidir-enable
!
call rsvp-sync
!
voice-port 1/0:0
 output attenuation 3
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 3640110 voip
 application session
 incoming called-number 3640110
 destination-pattern 3640110
 rtp payload-type nte 102
```

```
 session protocol sipv2
 session target ipv4:10.10.1.4
 dtmf-relay rtp-nte
 codec g711ulaw
!
dial-peer voice 3660110 pots
 application session
 destination-pattern 3660110
 port 2/0/0
!
sip-ua
 set sip-status 486 pstn-cause 34
 set pstn-cause 17 sip-status 503
 no oli
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
end
```

SIP Accept-Language Header Support Examples

The following provides partial output for SIP Accept-Language Header Support configured in voice service configuration mode and dial-peer configuration mode.

```
Router# show running-config
Building configuration...
Current configuration :2791 bytes
.
.
.
voice service pots
  supported-language yo
  supported-language sd language-param 0.234
  supported-language fr language-param 0.123
.
.
.
end!
Router# show running-config
Building configuration...
Current configuration :2791 bytes
.
.
.
dial-peer voice 1 pots
 application session
 destination-pattern 36601
 port 2/0/0
 supported-language sd
 supported-language zu
 supported-language ln language-param 0.123
.
.
.
end!
```

SIP Extensions for Caller Identity and Privacy Example

In the following example, the PSTN name is set to Company A and the PSTN number is set to 5550111.

```
Router(config-sip-ua)# calling-info sip-to-pstn name set CompanyA
Router(config-sip-ua)# calling-info sip-to-pstn number set 5550111
!
Router# show running-config
Building configuration...
Current configuration :2791 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3640
!
voice-card 2
!
ip subnet-zero
!
no ip domain lookup
ip domain name example.com
ip name-server 172.18.195.113
!
isdn switch-type primary-ni
!
fax interface-type fax-mail
mta receive maximum-recipients 0
ccm-manager mgcp
!
controller T1 2/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2/1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0/0
 ip address 172.18.197.22 255.255.255.0
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial2/0:23
 no ip address
```

```
 no logging event link-status
 isdn switch-type primary-ni
 isdn incoming-voice voice
 isdn outgoing display-ie
 no cdp enable
!
interface Serial2/1:23
 no ip address
 no logging event link-status
 isdn switch-type primary-ni
 isdn incoming-voice voice
 isdn outgoing display-ie
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
no ip http server
ip pim bidir-enable
!
call rsvp-sync
!
voice-port 2/0:23
!
voice-port 2/1:23
!
voice-port 3/0/0
!
voice-port 3/0/1
!
mgcp ip qos dscp cs5 media
mgcp ip qos dscp cs3 signaling
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 1 voip
 incoming called-number 5552222
 destination-pattern 5552222
 session protocol sipv2
 session target ipv4:172.18.197.27
!
dial-peer voice 2 pots
 destination-pattern 5551111
 no digit-strip
 direct-inward-dial
 port 2/0:23
!
gateway
!
sip-ua
calling-info sip-to-pstn name set CompanyA
calling-info sip-to-pstn number set 5550111
!
line con 0
line aux 0
line vty 0 4
 login
!
end!
```

SIP Session Timer Support Example

This example contains partial output showing that the Min-SE value has been changed from its default value. If the default value of 90 seconds remains unchanged, configuration data is not provided.

```
Router# show running-config
.
.
.
!
voice service voip
 sip
  min-se 950
!
.
.
.
```

SIP Cisco IOS Gateway Reason Header and Buffered Calling Name Completion Examples

Reason Header Enabled

The following examples shows output for the **show running-config** command with reason header enabled.

```
Current configuration :4643 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname cartman
!
boot-start-marker
boot-end-marker
!
no logging buffered
enable secret 5 $1$exgC$6yn9bof7/cYMhpNH9DfOp/
enable password password1
!
username 4444
username 232
username username1 password 0 password2
clock timezone EST -5
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
ip domain name example.sip.com
ip name-server 172.18.192.48
```

```
!
ip dhcp pool 1
   host 172.18.193.173 255.255.255.0
   client-identifier 0030.94c2.5d00
   option 150 ip 172.18.193.98
   default-router 172.18.193.98
!
no scripting tcl init
no scripting tcl encdir
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
 sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
fax interface-type fax-mail
!
translation-rule 100
!
interface FastEthernet0/0
 ip address 172.18.193.98 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 ip address 10.1.1.98 255.0.0.0
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 000000090200003094 26F6D0
snmp-server community public RO
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
```

```
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
 station-id number 36601
 caller-id enable
!
voice-port 1/1/1
!
voice-port 2/0/0
 caller-id enable
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
mgcp
mgcp sdp simple
!
dial-peer cor custom
!
dial-peer voice 6 voip
 destination-pattern 36602
 session protocol sipv2
 session target ipv4:10.102.17.80
 session transport tcp
 incoming called-number 36601
 codec g711ulaw
!
dial-peer voice 5 voip
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.197.182
!
dial-peer voice 1 pots
 destination-pattern 36601
 port 2/0/0
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
 dtmf-relay cisco-rtp
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 dtmf-relay rtp-nte
!
dial-peer voice 41 voip
```

```
 application session
 destination-pattern 777
 session protocol sipv2
 session target ipv4:172.18.199.94
 session transport udp
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
 incoming called-number 999
!
dial-peer voice 2 pots
 destination-pattern 361
 port 2/1/1
!
dial-peer voice 55 voip
 destination-pattern 5678
 session protocol sipv2
 session target ipv4:192.0.2.208:5061
!
dial-peer voice 361 voip
 incoming called-number 361
!
dial-peer voice 100 voip
!
dial-peer voice 3 pots
 destination-pattern 36601
 port 2/0/1
!
dial-peer voice 111 pots
!
dial-peer voice 11 pots
 preference 5
 destination-pattern 123
 port 2/0/0
!
dial-peer voice 12 pots
 destination-pattern 456
 port 2/0/0
!
dial-peer voice 14 pots
 destination-pattern 980
 port 2/0/0
!
dial-peer voice 15 pots
 destination-pattern 789
 port 2/0/0
!
gateway
!
sip-ua
 retry invite 2
 retry response 4
 retry bye 2
 retry cancel 1
 timers expires 300000
 sip-server dns:example-srv.sip.com
 reason-header override ! reason header enabled
!
telephony-service
 mwi relay
 mwi expires 600
```

```
 max-conferences 8
!
banner motd ^Chello^C
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
password password1
 transport preferred all
 transport input all
 transport output all
!
end
```

The following shows output for the **show sip-ua status**command with reason header enabled.

```
SIP User Agent Status
SIP User Agent for UDP :ENABLED
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 :70
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:2000 ms
SIP support for ISDN SUSPEND/RESUME:ENABLED
Redirection (3xx) message handling:ENABLED
Reason Header will override Response/Request Codes:ENABLED ! Reader Header Enabled
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
```

Reason Header Disabled

The following shows output for the **show running-config**command with reason header disabled.

```
Current configuration :4619 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
```

```
!
no logging buffered
enable secret 5 $1$exgC$6yn9bof7/cYMhpNH9DfOp/
enable password password1
!
username 4444
username 232
username username1 password 0 password2
clock timezone EST -5
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
ip domain name example.sip.com
ip name-server 172.18.192.48
!
ip dhcp pool 1
   host 172.18.193.173 255.255.255.0
   client-identifier 0030.94c2.5d00
   option 150 ip 172.18.193.98
   default-router 172.18.193.98
!
no scripting tcl init
no scripting tcl encdir
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
 sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
fax interface-type fax-mail
!
translation-rule 100
!
interface FastEthernet0/0
 ip address 172.18.193.98 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 ip address 10.1.1.98 255.0.0.0
 shutdown
 duplex auto
 speed auto
 no cdp enable
```

```
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 00000009020000309426F6D0
snmp-server community public RO
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
 station-id number 36601
 caller-id enable
!
voice-port 1/1/1
!
voice-port 2/0/0
 caller-id enable
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
mgcp
mgcp sdp simple
!
dial-peer cor custom
!
dial-peer voice 6 voip
 destination-pattern 36602
 session protocol sipv2
 session target ipv4:10.102.17.80
 session transport tcp
 incoming called-number 36601
 codec g711ulaw
!
dial-peer voice 5 voip
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.197.182
!
```

```
dial-peer voice 1 pots
 destination-pattern 36601
 port 2/0/0
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
 dtmf-relay cisco-rtp
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 dtmf-relay rtp-nte
!
dial-peer voice 41 voip
 application session
 destination-pattern 777
 session protocol sipv2
 session target ipv4:172.18.199.94
 session transport udp
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
 incoming called-number 999
!
dial-peer voice 2 pots
 destination-pattern 361
 port 2/1/1
!
dial-peer voice 55 voip
 destination-pattern 5678
 session protocol sipv2
 session target ipv4:10.102.17.208:5061
!
dial-peer voice 361 voip
 incoming called-number 361
!
dial-peer voice 100 voip
!
dial-peer voice 3 pots
 destination-pattern 36601
 port 2/0/1
!
dial-peer voice 111 pots
!
dial-peer voice 11 pots
 preference 5
 destination-pattern 123
 port 2/0/0
!
dial-peer voice 12 pots
 destination-pattern 456
 port 2/0/0
!
dial-peer voice 14 pots
 destination-pattern 980
```

```
 port 2/0/0
!
dial-peer voice 15 pots
 destination-pattern 789
 port 2/0/0
!
gateway
!
sip-ua
 retry invite 2
 retry response 4
 retry bye 2
 retry cancel 1
 timers expires 300000
 sip-server dns:example-srv.sip.com
!
telephony-service
 mwi relay
 mwi expires 600
 max-conferences 8
!
banner motd ^Chello^C
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
password password1
 transport preferred all
 transport input all
 transport output all
!
end
```

The following shows output for the **show sip-ua status**command with reason header disabled.

```
SIP User Agent Status
SIP User Agent for UDP :ENABLED
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 :70
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:2000 ms
SIP support for ISDN SUSPEND/RESUME:ENABLED
Redirection (3xx) message handling:ENABLED
Reason Header will override Response/Request Codes:DISABLED ! Reason Header 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
```

Buffer Calling Completion Enabled

```
Current configuration :4646 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
no logging buffered
enable secret 5 $1$exgC$6yn9bof7/cYMhpNH9DfOp/
enable password password1
!
username 4444
username 232
username username1 password 0 password2
clock timezone EST -5
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
ip domain name example.sip.com
ip name-server 172.18.192.48
!
ip dhcp pool 1
   host 172.18.193.173 255.255.255.0
   client-identifier 0030.94c2.5d00
   option 150 ip 172.18.193.98
   default-router 172.18.193.98
!
no scripting tcl init
no scripting tcl encdir
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
 sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
fax interface-type fax-mail
```

```
!
translation-rule 100
!
interface FastEthernet0/0
 ip address 172.18.193.98 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 ip address 10.1.1.98 255.0.0.0
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 000000090200003094 26F6D0
snmp-server community public RO
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
 station-id number 36601
 caller-id enable
!
voice-port 1/1/1
!
voice-port 2/0/0
 caller-id enable
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
mgcp
mgcp sdp simple
!
```

```
dial-peer cor custom
!
dial-peer voice 6 voip
 destination-pattern 36602
 session protocol sipv2
 session target ipv4:10.102.17.80
 session transport tcp
 incoming called-number 36601
 codec g711ulaw
!
dial-peer voice 5 voip
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.197.182
!
dial-peer voice 1 pots
 destination-pattern 36601
 port 2/0/0
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
 dtmf-relay cisco-rtp
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 dtmf-relay rtp-nte
!
dial-peer voice 41 voip
 application session
 destination-pattern 777
 session protocol sipv2
 session target ipv4:172.18.199.94
 session transport udp
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
 incoming called-number 999
!
dial-peer voice 2 pots
 destination-pattern 361
 port 2/1/1
!
dial-peer voice 55 voip
 destination-pattern 5678
 session protocol sipv2
 session target ipv4:10.102.17.208:5061
!
dial-peer voice 361 voip
 incoming called-number 361
!
dial-peer voice 100 voip
!
dial-peer voice 3 pots
```

```
 destination-pattern 36601
 port 2/0/1
!
dial-peer voice 111 pots
!
dial-peer voice 11 pots
 preference 5
 destination-pattern 123
 port 2/0/0
!
dial-peer voice 12 pots
 destination-pattern 456
 port 2/0/0
!
dial-peer voice 14 pots
 destination-pattern 980
 port 2/0/0
!
dial-peer voice 15 pots
 destination-pattern 789
 port 2/0/0
!
gateway
!
sip-ua
 retry invite 2
 retry response 4
 retry bye 2
 retry cancel 1
 timers expires 300000
 timers buffer-invite 5000 ! Buffer Calling Completion enabled
 sip-server dns:example-srv.sip.com
!
!
telephony-service
 mwi relay
 mwi expires 600
 max-conferences 8
!
banner motd ^Chello^C
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
password password1
 transport preferred all
 transport input all
 transport output all
!
end
```

Buffer Calling Completion Disabled

```
Current configuration :4619 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec localtime
```

```
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
no logging buffered
enable secret 5 $1$exgC$6yn9bof7/cYMhpNH9DfOp/
enable password password1
!
username 4444
username 232
username username1 password 0 password2
clock timezone EST -5
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
ip domain name example.sip.com
ip name-server 172.18.192.48
!
ip dhcp pool 1
   host 172.18.193.173 255.255.255.0
   client-identifier 0030.94c2.5d00
   option 150 ip 172.18.193.98
   default-router 172.18.193.98
!
no scripting tcl init
no scripting tcl encdir
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
 sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
fax interface-type fax-mail
!
translation-rule 100
!
interface FastEthernet0/0
 ip address 172.18.193.98 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
```

```
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 ip address 10.1.1.98 255.0.0.0
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 00000009020000309426F6D0
snmp-server community public RO
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
 station-id number 36601
 caller-id enable
!
voice-port 1/1/1
!
voice-port 2/0/0
 caller-id enable
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
mgcp
mgcp sdp simple
!
dial-peer cor custom
!
dial-peer voice 6 voip
 destination-pattern 36602
 session protocol sipv2
 session target ipv4:192.0.2.80
 session transport tcp
 incoming called-number 36601
```

```
 codec g711ulaw
!
dial-peer voice 5 voip
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.197.182
!
dial-peer voice 1 pots
 destination-pattern 36601
 port 2/0/0
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
 dtmf-relay cisco-rtp
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 dtmf-relay rtp-nte
!
dial-peer voice 41 voip
 application session
 destination-pattern 777
 session protocol sipv2
 session target ipv4:172.18.199.94
 session transport udp
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
 incoming called-number 999
!
dial-peer voice 2 pots
 destination-pattern 361
 port 2/1/1
!
dial-peer voice 55 voip
 destination-pattern 5678
 session protocol sipv2
 session target ipv4:10.102.17.208:5061
!
dial-peer voice 361 voip
 incoming called-number 361
!
dial-peer voice 100 voip
!
dial-peer voice 3 pots
 destination-pattern 36601
 port 2/0/1
!
dial-peer voice 111 pots
!
dial-peer voice 11 pots
 preference 5
 destination-pattern 123
```

```
 port 2/0/0
!
dial-peer voice 12 pots
 destination-pattern 456
 port 2/0/0
!
dial-peer voice 14 pots
 destination-pattern 980
 port 2/0/0
!
dial-peer voice 15 pots
 destination-pattern 789
 port 2/0/0
!
gateway
!
sip-ua
 retry invite 2
 retry response 4
 retry bye 2
 retry cancel 1
 timers expires 300000
 sip-server dns:example-srv.sip.com
!
telephony-service
 mwi relay
 mwi expires 600
 max-conferences 8
!
banner motd ^Chello^C
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password password1
 transport preferred all
 transport input all
 transport output all
!
end
```

SIP SIP Header URL Support and SUBSCRIBE NOTIFY for External Triggers Examples

SIP Header Support and Subscription

In the following example, header passing is enabled and a default server IP address is configured. The history log is configured to retain 100 history records, each of which is retained for fifteen minutes after the subscription is removed. SUBSCRIBE messages are configured to retransmit six times.

```
Router# show running-config
Building configuration...
version 12.2
no service pad
```

```
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
service internal
!
hostname Router
!
subscription asnl session history duration 15
subscription asnl session history count 100
logging buffered 1000000 debugging
!
resource-pool disable
!
ip subnet-zero
ip host server.example.com 10.7.104.88
!!
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
voice call carrier capacity active
!
voice service voip
h323
sip
!!
The Cisco IOS VoiceXML features are enabled, and the maximum number of subscriptions to be
 originated by the gateway is configured.
header-passing
subscription maximum originate 200
!
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
!
controller T1 1
 framing sf
 clock source line secondary 1
 linecode ami
!
controller T1 2
 framing sf
 clock source line secondary 2
 linecode ami
!
controller T1 3
 framing sf
 clock source line secondary 3
 linecode ami
!
interface Ethernet0
 ip address 10.7.102.35 255.255.0.0
 ip helper-address 223.255.254.254
 no ip mroute-cache
 no cdp enable
!
interface Serial0
 no ip address
 no ip mroute-cache
 shutdown
```

```
 clockrate 2015232
 no fair-queue
 no cdp enable
!
interface Serial1
 no ip address
 no ip mroute-cache
 shutdown
 clockrate 2015232
 no fair-queue
 no cdp enable
!
interface Serial2
 no ip address
 no ip mroute-cache
 shutdown
 clockrate 2015232
 no fair-queue
 no cdp enable
!
interface Serial3
 no ip address
 no ip mroute-cache
 shutdown
 clockrate 2015232
 no fair-queue
 no cdp enable
!
interface Serial0:23
 no ip address
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 isdn disconnect-cause 1
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 172.19.139.114 255.255.255.0
 no ip mroute-cache
 duplex auto
 speed auto
 no cdp enable
!
ip default-gateway 172.19.139.1
ip classless
ip route 172.71.56.39 255.255.255.255 172.19.139.1
ip route 10.255.254.0 255.255.255.0 10.7.104.1
no ip http server
!
ip pim bidir-enable
!
!
no cdp run
!
call application voice mwi tftp://dirt/ramsubra/cli_mwi.tcl
!
voice-port 0:D
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
```

```
dial-peer cor custom
!
dial-peer voice 1 pots
 application mwi
 destination-pattern 408.......
 incoming called-number 52943
 port 0:D
 prefix 950
!
dial-peer voice 789 voip
 destination-pattern 789
 session target ipv4:10.7.104.88
 codec g711ulaw
!
dial-peer voice 766 voip
 application get_headers_tcl
 session protocol sipv2
 session target ipv4:10.7.102.35
 incoming uri request 766
 codec g711ulaw
!
dial-peer voice 88888 pots
 destination-pattern 767....
 port 0:D
 prefix 9767
!
sip-ua
retry subscribe 6
!
line con 0
 exec-timeout 0 0
 logging synchronous level all
line aux 0
line vty 0 4
!
end
```

SIP Domain Name Support in SIP Headers Examples

Configuration in Gateway-Wide Global Configuration Mode

The following example shows the command output when the local hostname uses the gateway-wide global configuration settings.

```
Router# show running-config
Building configuration...
Current configuration :3512 bytes
!
! Last configuration change at 14:25:20 EDT Tue Aug 31 2004
! NVRAM config last updated at 14:17:44 EDT Tue Aug 31 2004
!
version 12.3
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
! voice service voip
 sip
  localhost dns:example.com
!
!
```

Configuration in Dial-Peer-Specific Dial-Peer Configuration Mode

The following example shows the command output when the local hostname uses the dial-peer configuration settings.

```
Router# show running-config
Building configuration...
Current configuration :3512 bytes
!
! Last configuration change at 14:25:20 EDT Tue Aug 31 2004
! NVRAM config last updated at 14:17:44 EDT Tue Aug 31 2004
!
version 12.3
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
!
!
dial-peer voice 3301 voip
 destination-pattern 9002
 voice-class sip localhost  dns:gw11.example.com
 session protocol sipv2
 session target dns:example.sip.com
 session transport tcp
 dtmf-relay rtp-nte
!
```

SIP Gateway Support for Permit Hostname Example

The following example shows a configured list of hostnames.

```
router>
 enable
router#
 configure terminal
router (config)#
sip-ua
router (config-sip-ua)#
 permit hostname dns:
esample1.sip.com
router (config-sip-ua)#
permit hostname dns:example2.sip.comrouter (config-sip-ua)#
permit hostname dns:example3.sip.comrouter (config-sip-ua)#
permit hostname dns:example4.sip.comrouter (config-sip-ua)#
permit hostname dns:example5.sip.comrouter (config-sip-ua)#
 exit
```

Outbound-Proxy Support for the SIP Gateway Examples

The following example shows how to configure an outbound-proxy server globally on a gateway for the specified IP address:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# outbound-proxy ipv4:10.1.1.1
```

The following example shows how to configure an outbound-proxy server globally on a gateway for the specified domain:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# outbound-proxy dns:sipproxy:example.com
```

The following examples shows how to configure an outbound-proxy server on a dial peer for the specified IP address:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# dial-peer voice 111 voip
gateway(conf-dial-peer)# voice-class sip
gateway(conf-dial-peer)# outbound-proxy ipv4:10.1.1.1
```

The following examples shows how to configure an outbound-proxy server on a dial peer for the specified domain:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# dial-peer voice 111 voip
gateway(conf-dial-peer)# voice-class sip
gateway(conf-dial-peer)# outbound-proxy dns:sipproxy.example.com
```

The following example shows how to disable the global outbound proxy feature for all line-side SIP phones on Cisco Unified CME:

```
gateway> enable
gateway# configure
 terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice register global
gateway(config-register-global)# no outbound-proxy
```

SIP SIP Support for PAI Examples

Configuring a Privacy Header Example

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# privacy
```

Configuring PPI Example

The following example shows how to configure a privacy header level for PPI:

```
gateway> enable
```

```
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# asserted-id ppi
```

Configuring PAI Example

The following example shows how to configure a privacy header level for PAI:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# asserted-id pai
```

SIP History-Info Header Support Examples

The following example shows how to configure history-info header support at the global level:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
gateway(config)# voice service voip
gateway(conf-voi-serv)# sip
gateway(conf-serv-sip)# history-info
```

The following example shows partial output from the **show running-config** command when history-info header support is configured at the global level:

```
gateway# show running-config
Building configuration...
Current configuration : 10198 bytes
.
.
voice service voip
 fax protocol t38 ls-redundancy 0 hs-redundancy 0 fallback none
 sip
  privacy user critical
  history-info
.
.
dial-peer voice 1 voip
 voice-class sip resource priority namespace drsn
 voice-class sip privacy header id critical
!
dial-peer voice 2 voip
 voice-class sip privacy header critical
.
.
end
```

The following example shows how to configure history-info header support at the dial-peer level:

```
gateway> enable
gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
gateway(config)# dial-peer voice 2 voip
gateway(config-dial-peer)# voice-class sip history-info
```

The following example shows partial output from the **show running-config** command when history-info header support is configured at the dial-peer level:

```
gateway# show running-config
Building configuration...
Current configuration : 10183 bytes
.
.
voice service voip
 fax protocol t38 ls-redundancy 0 hs-redundancy 0 fallback none
 sip
  privacy user critical
.
.
dial-peer voice 1 voip
voice-class sip resource priority namespace drsn
voice-class sip privacy header id critical
!
dial-peer voice 2 voip
 voice-class sip privacy header history critical
.
.
end
```

Additional References

The following sections provide references related to the SIP message, timer, and response features.

Related Documents

Related Topic	Document Title
"Achieving SIP RFC Compliance" module	*Cisco IOS SIP Configuration Guide*
"Overview of SIP" module	*Cisco IOS SIP Configuration Guide*
Cisco IOS Tcl IVR and VoiceXML Application Guide	Cisco IOS Release 12.3(14)T and later: http://www.cisco.com/en/US/docs/ios/voice/ivr/configuration/guide/tcl_c.html
	Cisco IOS releases prior to 12.3(14)T: http://www.cisco.com/en/US/docs/ios/voice/ivr/pre12.3_14_t/configuration/guide/ivrapp.pdf
Cisco IOS Voice Command Reference	http://www.cisco.com/en/US/docs/ios/voice/command/reference/vr_book.html

Related Topic	Document Title
Cisco Unified Communications Manager Express Command Reference	http://www.cisco.com/en/US/docs/voice_ip_comm/cucme/command/reference/cme_cr.html
Cisco Unified Communications Manager Express support documentation	http://www.cisco.com/en/US/products/sw/voicesw/ps4625/tsd_products_support_series_home.html
Cisco Unified SIP SRST System Administrator Guide	http://www.cisco.com/en/US/docs/voice_ip_comm/cusrst/admin/sccp_sip_srst/configuration/guide/srst_
Cisco VoiceXML Programmer's Guide	http://www.cisco.com/en/US/docs/ios/voice/vxml/developer/guide/vxmlprg.html
Tcl IVR API Version 2.0 Programming Guide	http://www.cisco.com/en/US/docs/ios/voice/tcl/developer/guide/tclivrv2.html

Standards

Standard	Title
International Organization for Standardization (ISO) specification, ISO 639	Codes for Representation of Names of Languages

MIBs

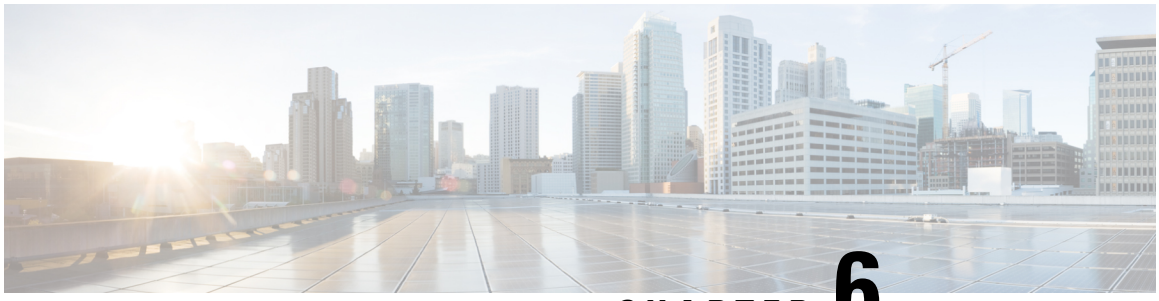
MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2833	RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
RFC 3261	SIP: Session Initiation Protocol
RFC 3262	Reliability of Provisional Responses in the Session Initiation Protocol (SIP)
RFC 3264	An Offer/Answer Model with the Session Description Protocol (SDP)
RFC 3265	Session Initiation Protocol (SIP)-Specific Event Notification
RFC 3312	Integration of Resource Management and Session Initiation Protocol (SIP)
RFC 3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)
RFC 3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
RFC 3326	The Reason Header Field for the Session Initiation Protocol (SIP)
RFC 4028	Session Timers in the Session Initiation Protocol (SIP)
RFC 4244	An Extension to the Session Initiation Protocol (SIP) for Request History Information

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport



CHAPTER 6

Support for Resource Availability Indication Over SIP Trunks

The Support for Monitoring Utilization of Critical Resources on Gateway Router, Cisco UBE and Cisco UCME and Reporting Over SIP Trunks feature implements monitoring of resource utilization and reporting functionality over the Session Initiation Protocol (SIP) trunk on the Cisco IOS gateway, Cisco Unified Border Element (Cisco UBE) and Cisco Unified Communications Manager Express (Cisco UCME). This feature supports monitoring of CPU, memory, Digital Signaling Processor (DSP) and the DS0 port on the Cisco IOS gateway and reporting the status to external devices using SIP OPTION mechanism.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for the Support for Resource Availability Indication Over SIP Trunks Feature

- The number of resource monitoring entities should be limited to a maximum of 50.

- The target device configured on the Cisco IOS gateway and the OPTIONS message obtained from the external polling device must be the same to accept the resource status query. For example, if an IP address is configured as the target on the Cisco IOS gateway, it is expected that the OPTIONS message coming into the Cisco IOS gateway will have the same IP address. The same applies if a Fully Qualified Domain Name (FQDN) is configured against the target device.

Information About the Support for Resource Availability Indication Over SIP Trunks Feature

Overview of the Support for Resource Availablity Indication Over SIP Trunks

The Support for Resource Available Indication Over SIP Trunks feature enables monitoring of CPU, memory, DSP, and DS0 resources on the Cisco IOS gateway and reporting of the status to external device using the gateway resources. The reporting from the gateway takes place either based on a configured time interval or on the configured threshold crossover. The gateway also supports extracting the gateway resource information from the external entity either on periodic or nonperiodic basis.

The monitoring and reporting of resources is achieved by SIP OPTIONS using **x-cisco-rai** headers. The highlights of monitoring and reporting of resources are as follows:

- Resource groups are created by grouping the required resource monitoring entities together and providing each of the resource groups a unique index value.
- The required monitoring and reporting parameters are applied on the resource groups for monitoring the selected resource entities.
- In the periodic reporting, Resource Allocation Indication (RAI) information is reported based on a periodic-interval value configured through a CLI. The periodic reporting is disabled by default.
- In the threshold-based reporting, RAI information is reported when one of the resources reaches an out-of-resource state. Threshold-based reporting is disabled by default.

Note Call treatment is not handled as part of this feature on the gateway. If call treatment needs to be performed along with reporting, then the call threshold feature needs to be configured appropriately with desired threshold values. The gateway supports CPU, memory, and maximum number of calls per interface.

Benefits of the Support for Resource Availablity Indication Over SIP Trunks

- Intelligent routing can be performed by Customer Voice Portal (CVP) if the Cisco IOS gateway passes information with the methodology provided by this feature.
- Call handling is performed more efficiently in Voice over Internet Protocol (VOIP) networks where CVP uses the Cisco IOS gateway for both VoiceXML (VXML) application as well as time-division multiplexing (TDM) calls for Public Switched Telephone Network (PSTN) connectivity.
- By monitoring and reporting the usage status of the gateway resources and generic resources to the external entity, you can take decisions based on the current load and status of various gateway resources.

- Migration from H.323 to SIP is made easier.

Overview on Monitoring and Reporting of Gateway Resources

The gateway uses the **x-cisco-rai** header in the OPTIONS message to report the resource utilization to the routing or monitoring entities.

ABNF Format for Reporting

The Augmented Backus-Naur Form (ABNF) format of reporting is used to report the status of the report utilization to the monitoring entity. The Cisco IOS gateway may not use all the fields. Provision is also made for adding new resource types and new parameters for the resources.

The following is the format of the header used for reporting:

```
x-cisco-rai = ("x-cisco-rai") HCOLON resource-param *(COMMA resource-param)
resource-param = resource-name-param SEMI resource-params *(SEMI resource-params)
resource-name-param = "SYSTEM" / "CPU" / "MEM" / "DS0" / "DSP" / token
resource-params =
resource-status-param/resource-total-param/resource-available-param/resource-used-param/resource-extension
resource-status-param = "almost-out-of-resource" EQUAL ("true" / "false")
resource-total-param="total" EQUAL 1*(DIGIT) ["%" / "MB" / token]
resource-available-param = "available" EQUAL 1*(DIGIT) ["%" / "MB" / token]
resource-used-param = "used" EQUAL 1*3DIGIT "%"
resource-extension=generic-param
```

Semantics of the Header

The rules for constructing the header are as follows:

- SYSTEM is a mandatory parameter that should be the first resource for a given device. Other resource-name-param values are optional parameters.
- The resource-identity-param parameter is a mandatory parameter within the SYSTEM resource-name-param parameter.
- At least one resource parameter (resource-params) should be present. If the resource-available-param parameter is present, the resource-total-param parameter should also be present so that the resource-used-param parameter can be calculated.
- For the SYSTEM resource-name-param parameter, the provision of a resource-status-param parameter is mandatory.

Following are the examples of the reporting format in the SIP OPTIONS message:

Example 1: Cisco IOS Gateway Sending Resource Header when the Resources are Available

```
x-cisco-rai : SYSTEM; almost-out-of-resource=false
x-cisco-rai : CPU; almost-out-of-resource=false;total=100%;available=60%
x-cisco-rai : MEM; almost-out-of-resource=false;total=100%;available=40%
x-cisco-rai : DSO; almost-out-of-resource=false;total=64;available=20
x-cisco-rai : DSP; almost-out-of-resource=false;total=64;available=20
```

Example 2: Cisco IOS Gateway Sending Resource Header when CPU Resources are Not Available

```
x-cisco-rai : SYSTEM; almost-out-of-resource=true
x-cisco-rai : CPU; almost-out-of-resource=true;total=100%;available=20%
x-cisco-rai : MEM; almost-out-of-resource=false;total=100%;available=40%
x-cisco-rai : DSO; almost-out-of-resource=false;total=64;available=20
x-cisco-rai : DSP; almost-out-of-resource=false;total=64;available=20
```

Modes of Reporting

There are two modes of reporting:

Gateway Triggered Reporting to Routing Monitoring Entity

In gateway triggered reporting, the gateway sends the resource utilization information in an OPTION message to the external entity. The gateway includes a supported header with the option tag **x-cisco-rai** to indicate that this OPTION message carries the resource availability information. The trigger to send the OPTION message can be either periodic, or threshold based, or both.

Periodic Reporting

In the periodic reporting mode, reporting is triggered based on a preconfigured timer value. This type of reporting is used to collect information on a resource usage.

Threshold Triggered Reporting

In the threshold triggered reporting mode, reporting is triggered depending on the threshold levels configured for the resources that are being monitored. This mode of reporting is used to inform the external entity of the availability or nonavailability of the gateway to service further call requests from the external entity. Threshold triggered reporting is disabled by default.

Routing Monitoring Entity Triggered Reporting

In routing or monitor triggered reporting, the external entity can extract the resource information from the gateway using the SIP OPTIONS message. The gateway expects the **x-cisco-rai** tag in either the **require:** or **supported:** header fields for an incoming OPTIONS message. The presence of the **x-cisco-rai** tag in any one of the header fields informs the gateway that this option is to retrieve the gateway resource information.

Reporting Mechanism over SIP Trunk

The monitoring and reporting of gateway resources to an external entity is achieved by SIP OPTIONS using **x-cisco-rai** headers. The sample OPTIONS message format is as provided below.

Inbound OPTIONS Message

The following is a sample format of the incoming OPTIONS message and the corresponding response from the Cisco IOS gateway for resource statistics request:

```
Received:OPTIONS sip:9.13.38.162:5060 SIP/2.0Via: SIP/2.0/UDP
9.13.40.83:5060;branch=z9hG4bK-21983-1-0From: <sip:9.13.40.83:5060>;tag=1To: sut
<sip:service@9.13.38.162:5060>Call-ID: 1-21983@9.13.40.83CSeq: 1 OPTIONSContact:
sip:sipp@9.13.40.83:5060Max-Forwards: 70Subject: Performance TestSupported: sec-agree,
```

```
precondition
Require:  x-cisco-rai   Content-Type: application/sdpContent-Length: 0
```

Outbound OPTIONS Message

The following is a sample format of the OPTIONS message that is sent out from the Cisco IOS gateway to the external entity with resource statistics information in it:

```
Sent:OPTIONS sip:9.13.38.163:5060 SIP/2.0
Via: SIP/2.0/UDP 9.13.38.162:5060;branch=z9hG4bKE211D
From: <sip:9.13.38.162>;tag=1005B0-1CB
To: <sip:9.13.38.163>
Date: Wed, 16 Dec 2009 05:53:35 GMT
Call-ID: 2EE072F6-E93E11DE-801BBA85-BFCDE3D6@9.13.38.162
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 70
CSeq: 101 OPTIONS
Contact: <sip:9.13.38.162:5060>
X-cisco-rai: SYSTEM ; almost-out-of-resource=true
X-cisco-rai: CPU ; almost-out-of-resource=false;available=99%;total=100%;used=1%
X-cisco-rai: DS0 ; almost-out-of-resource=false;available=23;total=23;used=0%
X-cisco-rai: DSP ; almost-out-of-resource=false;available=96;total=96;used=0%
X-cisco-rai: MEM ; almost-out-of-resource=true;available=51%;total=100%;used=49%
Supported: x-cisco-rai
Content-Length: 0
```

How to Configure the Support for Resource Availability Indication Over SIP Trunks Feature

Configuring Resource Groups and Resource Monitoring Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class resource-group** *tag*
4. **resource** {**cpu** {**1-min-avg** | **5-sec-avg**} | **ds0** | **dsp** | **mem** {**io-mem** | **proc-mem** | **total-mem**}} [**threshold high** *threshold-value* **low** *threshold-value*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice class resource-group** *tag* **Example:** `Router(config)# voice class resource-group 1`	Enters voice-class configuration mode and assigns a unique value to the resource group.
Step 4	**resource** {**cpu** {**1-min-avg** | **5-sec-avg**} | **ds0** | **dsp** | **mem**{**io-mem** | **proc-mem** | **total-mem**}} [**threshold high** *threshold-value* **low** *threshold-value*] **Example:** `Router(config-class)# resource cpu 1-min-avg mem io-mem threshold high 5 low 1`	Selects the required resources to be monitored and configures parameters for monitoring them.
Step 5	**end** **Example:** `Router(config-class)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **show voice class resource-group** command to display the configuration parameters for monitoring gateway resources.

Configuring SIP RAI Mechanism

This task enables the router to extract the details of the SIP along with the index of the resource group that needs to be monitored.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **rai target** *target-address* **resource-group** *group-index* [**transport** [**tcp** [**tls** [**scheme** {**sip** \| **sips**}]] \| **udp**]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters into SIP UA configuration mode.
Step 4	**rai target** *target-address* **resource-group** *group-index* [**transport** [**tcp** [**tls** [**scheme** {**sip** | **sips**}]] | **udp**]] **Example:** `Router(config-class)# rai target ipv4:10.2.1.1 resource-group 1`	Configures the SIP RAI mechanism.
Step 5	**end** **Example:** `Router(config-class)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **debug rai**command to enable debugging for RAI.

Configuration Examples for the Support for Resource Availability Indication Over SIP Trunks Feature

Example Configuring Resource Groups and Resource Monitoring Parameters

The following example shows how to configure the resource group 1 to monitor CPU, DS0, DSP, and memory resources:

```
voice class resource-group 1
 resource cpu 1-min-avg threshold high 50 low 30
 resource ds0 threshold high 50 low 30
 resource dsp threshold high 50 low 30
 resource memory total-mem threshold high 50 low 30
 periodic-report interval 30
```

Example Configuring SIP RAI Mechanism

```
sip-ua
 rai target ipv4:9.13.40.83 resource-group 1 transport udp
 rai target dns:whitesmoke resource-group 2 transport tcp
 rai target ipv6:[2217:10:10:10:10:10:10:2] resource-group 3  transport tcp
 rai target dns:butterfly resource-group 4 transport tcp tls scheme sips
 rai target ipv4:10.13.40.84 resource-group 5 transport tcp
 rai target ipv4:10.13.40.83 resource-group 1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Voice commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Voice Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

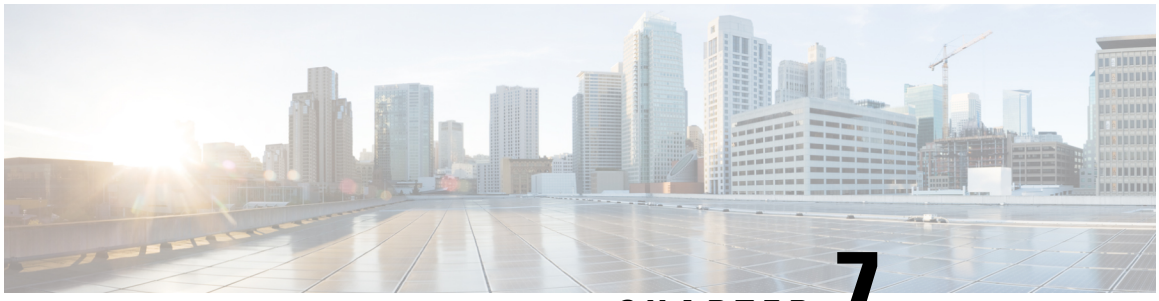
Feature Information for the Support for Resource Availability Indication Over SIP Trunks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for the Support for Resource Availablity Indication Over SIP Trunks Feature

Feature Name	Releases	Feature Information
Support for Monitoring Utilization of Critical Resources on Gateway Router, Cisco UBE and Cisco UCME and Reporting Over SIP Trunks	15.1(2)T	The Support for Monitoring Utilization of Critical Resources on Gateway Router, Cisco UBE and Cisco UCME and Reporting Over SIP Trunks feature implements monitoring of resource utilization and reporting functionality over SIP trunk on Cisco IOS gateway, Cisco UBE and Cisco UCME. The following commands were introduced or modified: **debug rai**, **rai target**, **voice class resource-group**, **show voice class resource-group**.



CHAPTER 7

Configuring Multiple Registrars on SIP Trunks

The Support for Multiple Registrars on SIP Trunks on a Cisco Unified Border Element, on Cisco IOS SIP TDM Gateways, and on Cisco Unified Communications Manager Express feature allows configuration of multiple registrars on Session Initiation Protocol (SIP) trunks, each simultaneously registered using their respective authentication instance. This feature allows a redundant registrar for each of the SIP trunks, which provides SIP trunk redundancy across multiple service providers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Multiple Registrars on SIP Trunks

Before configuring support for multiple registrars on the SIP trunks of Cisco IOS SIP time-division multiplexing (TDM) gateways, Cisco Unified Border Elements (Cisco UBEs), or Cisco Unified Communications Manager Express (Cisco Unified CME), verify the SIP configuration within the VoIP network for the appropriate originating and terminating gateways as described in documentation listed in the "Related Documents" section.

Restrictions for Configuring Multiple Registrars on SIP Trunks

Configuring multiple registrars on SIP trunks has the following restrictions:

- Old and new forms of registrar command are mutually exclusive: the registrar can be configured in either primary/secondary mode or multiple registrar mode--not both.
- Dynamic Host Configuration Protocol (DHCP) support is not available with multiple registrars (available for primary/secondary mode only).
- Only one authentication configuration per username can be configured at any one time.
- A maximum of six registrars can be configured at any given time.
- A maximum of 12 different realms can be configured for each endpoint.
- You cannot restrict the registration of specific endpoints with specific registrars--once a new registrar is configured, all endpoints will begin registering to the new registrar.
- You cannot remove multiple configurations of credentials simultaneously--only one credential can be removed at a time.

Information About Configuring Multiple Primary SIP Trunks

To configure multiple registrars on SIP trunks, you should understand the following concepts:

Purpose of Multiple Registrars on SIP Trunks

You can configure endpoints on Cisco IOS SIP TDM gateways, Cisco UBEs, and Cisco Unified CME to register to a primary and a secondary registrar. You can use this setup to register all endpoints to their assigned registrar using authentication details that can be associated with only one domain (or service provider). The drawback is that if the registrar or service provider domain becomes unavailable, users supported on Cisco IOS SIP TDM gateways, Cisco UBEs, and Cisco Unified CME cannot make or receive calls.

The Support for Multiple Registrars on SIP Trunks on a Cisco Unified Border Element, on Cisco IOS SIP TDM Gateways, and on Cisco Unified Communications Manager Express feature provides support for simultaneous registration with multiple registrars for endpoints on Cisco IOS SIP TDM gateways, Cisco UBEs, and Cisco Unified CME. This feature allows outbound and inbound traffic to be distributed across different service providers while simultaneously providing redundancy for users supported on these devices.

Note In a Cisco UBE multihome environment, all sets of credentials configured under the SIP user agent are sent to all configured registrars, regardless of realm configuration. This means that if a Cisco UBE registers multiple service providers, the credentials for both service providers are sent out to both. While the correct credentials will register, the incorrect sets will fail, possibly resulting in security measures taken by the service provider for failed registration attempts.

Authentication Attributes and Enhancements

The Support for Multiple Registrars on SIP Trunks on a Cisco Unified Border Element, on Cisco IOS SIP TDM Gateways, and on Cisco Unified Communications Manager Express feature is available in Cisco IOS Release 15.0(1)XA and later releases and utilizes the **authentication** command to allow configuration for specific endpoints and includes the following attributes and enhancements:

- You can configure up to 12 instances of the **authentication** command for a given endpoint, supporting 12 different realms for a single service provider domain.
- The authentication behavior can handle challenges from different service providers for SIP REGISTER and SIP INVITE/Other messages.
- You can specify a realm, which acts as a unique key for picking up username and password authentication details for each endpoint.
- The system uses authentication details to rebuild SIP Request messages in response to challenges.
- Enhancements introduced by this feature apply at both the dial peer and SIP user agent (UA) level.

Credentials Attributes and Enhancements

The **credentials** command is used to trigger SIP Register requests wherever registration is required for users who are not part of any POTS dial peers. The Support for Multiple Registrars on SIP Trunks on a Cisco Unified Border Element, on Cisco IOS SIP TDM Gateways, and on Cisco Unified Communications Manager Express feature is available in Cisco IOS Release 15.0(1)XA and later releases and modifies the **credentials** command to allow configuration for specific endpoints and includes the following attributes and enhancements:

- Credentials behavior supports SIP REGISTER and SIP INVITE/Other message challenges.
- You can use the **number** keyword to specify an endpoint, the value of which is used to populate the user portion of the To field in outgoing SIP REGISTER messages.
- The **number** keyword is optional and is used to allow the endpoint number to be different from the username.
- When both number and realm are configured, this pair of values acts as a unique key to pick up the username and password credentials configured for the endpoint.
- A maximum of 12 different realms can be configured for a given endpoint.

Determination of Authentication Details

When a SIP INVITE or SIP REGISTER request is challenged, the username and password details for authentication are determined in the following order:

Note Configuring more than one username is not supported--you must remove any currently configured username before configuring a new username.

1. If the realm specified in the challenge matches the realm in the authentication configuration for a POTS dial peer, the system uses the corresponding username and password.

2. If the realm specified in the challenge doesn't match the configured authentication for the POTS dial peer, then it will check for credentials configured for SIP UA.
3. If the realm specified in the challenge does not match the realm configured for credentials, then it will check for authentication configurations for SIP UA.
4. If the system does not find a matching authentication or credential for the received realm, then the request is terminated.
5. If there is no realm specified for the authentication configuration, then the system uses the username received from the challenge to build the response message.

Use of Preferred Option

When the primary and secondary registrar scenario is configured, the registrar address is used to populate the host portion of the From header in outbound SIP INVITE messages. However, when multiple registrars are configured, the only way to determine which registrar address is used to populate the host portion of the From header is to designate a single service provider domain as the preferred localhost name. The following behaviors are the result of using the **preferred** keyword with the **localhost** command (or **voice-class sip localhost** command for individual dial peers) when configuring multiple registrars:

- If the **localhost** command is enabled with the optional **preferred** keyword included, then the From header of outgoing SIP INVITE messages is populated with the domain name specified by the **localhost** command configuration.
- If the **localhost** command is not enabled or if it is enabled without the optional **preferred** keyword, then the best local interface address is used to populate the From header of outgoing SIP INVITE messages.

How to Configure Multiple Registrars on SIP Trunks

The tasks for configuring multiple registrars on the SIP trunks of Cisco IOS SIP TDM gateways, Cisco UBEs, or Cisco Unified CME are included in the following sections:

Configuring Multiple Registrars on SIP Trunks

To configure multiple registrars on SIP trunks, use the **registrar** and **authentication** commands as described in the following tasks:

Registration of POTS Endpoints to Multiple Registrars on SIP Trunks

Use the **registrar** command in SIP UA configuration mode to register POTS endpoints on a Cisco IOS SIP TDM gateway, Cisco UBE, or Cisco Unified CME to multiple registrars. To do so, you need to configure the command once for each registrar.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registrar** *registrar-index registrar-server-address* **expires** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP UA configuration mode.
Step 4	**registrar** *registrar-index registrar-server-address* **expires** *seconds* **Example:** `Router(config-sip-ua)# registrar 1 dns:example1.com expires 180` **Example:** `Router(config-sip-ua)# registrar 2 dns:example2.com expires 60`	Enables a Cisco IOS voice gateway to register E.164 numbers with external SIP proxies or SIP registrars. The *register-index* range is 1 to 6. (The expires timer specification is optional).

Global Configuration of Authentication for POTS Endpoints with Multiple Registrars on a SIP Trunk

Use the **authentication** command in SIP UA configuration mode to configure authentication of endpoints on a Cisco IOS SIP TDM gateway, Cisco UBE, or Cisco Unified CME to multiple registrars. To do so, you need to configure the command once for each registrar.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **authentication username** *username* **password** [**0** | **7**] *password* [**realm** *realm*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP UA configuration mode.
Step 4	**authentication username** *username* **password** [**0** | **7**] *password* [**realm** *realm*] **Example:** `Router(config-sip-ua)# authentication username MyUsername password MyPassword realm Realm1.example.com` **Example:** `Router(config-sip-ua)# authentication username MyUsername password MyPassword realm Realm2.example.com`	Enables SIP digest authentication globally (encryption type and realm specification are optional).

Dial Peer Configuration of Authentication for POTS

Use the **authentication** command in dial peer voice configuration mode to authenticate endpoints on a Cisco IOS SIP TDM gateway to multiple registrars on SIP trunks. To do so, you need to configure the command once for each registrar.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice 1 pots**
4. **authentication username** *username* **password** [**0** | **7**] *password* [**realm** *realm*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice 1 pots** **Example:** `Router(config)# dial-peer voice 1 pots`	Enters dial peer voice configuration mode.
Step 4	**authentication username** *username* **password** [**0** | **7**] *password* [**realm** *realm*] **Example:** `Router(config-dial-peer)# authentication username MyUser password 7 MyPassword realm Realm1.example.com` **Example:** `Router(config-dial-peer)# authentication username MyUser password 7 MyPassword realm Realm2.example.com`	Enables SIP digest authentication on an individual dial peer (encryption type and realm specification are optional).

Configuration of Credentials for Registering POTS Endpoints with Multiple Registrars on SIP Trunks

Use the **credentials** command in SIP UA configuration mode to configure registration requests sent from a Cisco IOS SIP TDM gateway, Cisco Unified CME, or a Cisco UBE to multiple registrars on a SIP trunk.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **credentials** {**dhcp** | **number** *number* **username** *username*} **password** [**0** | **7**] *password* **realm** *realm*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP UA configuration mode.
Step 4	**credentials** {**dhcp** | **number** *number* **username** *username*} **password** [**0** | **7**] *password* **realm** *realm* **Example:** `Router(config-sip-ua)# credentials number 1111 username MyUsername password MyPassword realm Realm1.example.com` **Example:** `Router(config-sip-ua)# credentials number 1111 username MyUsername password MyPassword realm Realm2.example.com` **Example:** `Router(config-sip-ua)# credentials number 1111 username AnotherUsername password TheirPassword realm Realm1.example.com` **Example:** `Router(config-sip-ua)# credentials number 1111 username AnotherUsername password TheirPassword realm Realm2.example.com`	Configures credentials for endpoints on a Cisco IOS SIP TDM gateway, Cisco Unified CME, or Cisco UBE for responding to authentication challenges. **Note** The encryption type specification is optional but the **dhcp** keyword is not allowed when configuring credentials for multiple registrars on a SIP trunk.

Configuration Examples for Configuring Multiple Registrars on SIP Trunks

Registering POTS Endpoints to Multiple Registrars on SIP Trunks Example

The following example shows how to configure POTS endpoints to register with multiple registrars ("example1.com" and "example2.com") on a SIP trunk simultaneously:

```
Router> enable
Router# configure terminal
Router(config)# sip-ua
Router(config-sip-ua)# registrar 1 dns:example1.com expires 180
Router(config-sip-ua)# registrar 2 dns:example2.com expires 360
Router(config-sip-ua)# authentication username MyUsername password MyPassword1 realm
Realm.example1.com
Router(config-sip-ua)# authentication username MyUsername password MyPassword2 realm
AnotherRealm.example1.com
Router(config-sip-ua)# authentication username MyUsername password MyPassword3 realm
```

```
Realm.example2.com
Router(config-sip-ua)# authentication username MyUsername password MyPassword4 realm
AnotherRealm.example2.com
```

Registering Individual POTS Endpoints on a Cisco IOS SIP

The following example shows how to configure individual POTS endpoints on a Cisco IOS SIP TDM gateway to register with multiple registrars ("Realm1" and "Realm2") on a SIP trunk simultaneously:

```
Router> enable
Router# configure terminal
Router(config)# dial-peer voice 3 pots
Router(config-dial-peer)# authentication username MyUser password MyPassword realm
realm1.example.com
Router(config-dial-peer)# authentication username MyUser password MyPassword2 realm
Realm2.example.com
Router(config-dial-peer)# exit
Router(config)# sip-ua
Router(config-sip-ua)# registrar 1 dns:example1.com expires 180
Router(config-sip-ua)# registrar 2 ipv4:1.1.1.1 expires 360
```

Configuring Credentials for Endpoints to Register with Multiple Registrars on a SIP Trunk Example

The following example shows how to configure credentials for endpoints on a Cisco IOS SIP TDM gateway, Cisco Unified CME, or Cisco UBE to register with multiple registrars ("Example1.com" and "Example2.com") on a SIP trunk simultaneously:

```
Router> enable
Router# configure terminal
Router(config)# sip-ua
Router(config-sip-ua)# credentials number 1111 username MyUsername password MyPassword realm
 Realm.example1.com
Router(config-sip-ua)# credentials number 1111 username MyUsername2 password MyOtherPassword
 realm AnotherRealm.example1.com
Router(config-sip-ua)# credentials number 2222 username TheirUsername password TheirPassword
 realm Realm.example1.com
Router(config-sip-ua)# credentials number 2222 username TheirUsername2 password
TheirOtherPassword realm AnotherRealm.example1.com
Router(config-sip-ua)# registrar 1 dns:example1.com expires 180
Router(config-sip-ua)# registrar 2 dns:example2.com expires 360
```

Additional References

The following sections provide references related to the SIP Connection-Oriented Media, Forking, and MLPP features.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
SIP commands	*Cisco IOS Voice Command Reference*
Tcl IVR and VoiceXML	*Cisco IOS Tcl IVR and VoiceXML Application Guide*
Cisco VoiceXML	*Cisco VoiceXML Programmer's Guide*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Multiple Registrars on SIP Trunks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 36: Feature Information for Configuring Multiple Registrars on SIP Trunks

Feature Name	Releases	Feature Information
Support for Multiple Registrars on SIP Trunks on a Cisco Unified Border Element, on Cisco IOS SIP TDM Gateways, and on Cisco Unified Communications Manager Express feature	15.0(1)XA 15.1(1)T	This feature provides support for multiple registrars on SIP trunks on Cisco IOS SIP TDM gateways, Cisco Unified CME, and Cisco UBEs. This feature allows for a redundant registrar for each SIP trunk and enables registrar redundancy across multiple service providers. This feature includes the following new or modified commands: **credentials**, **localhost**, **registrar**, **voice-class sip localhost**.

Glossary

ISDN --Integrated Services Digital Network.

EFXS --IP phone virtual voice ports.

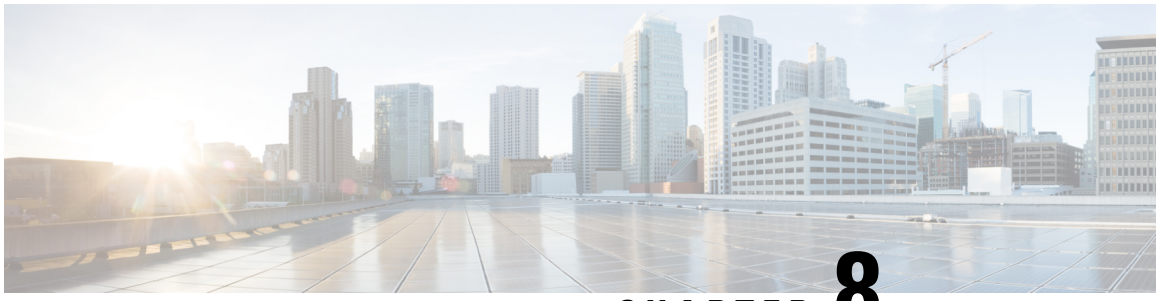
FXS --analog telephone voice ports.

SCCP --Skinny Client Control Protocol.

SDP --Session Description Protocol.

SIP --Session Initiation Protocol.

TDM --time-division multiplexing.



CHAPTER 8

Configuring SIP AAA Features

This chapter describes how to configure the following SIP AAA features:

- Configurable Screening Indicator (handled in this document as a subset of SIP - Enhanced Billing Support for Gateways)
- RADIUS Pre-authentication for Voice Calls
- SIP - Enhanced Billing Support for Gateways
- SIP: Gateway HTTP Authentication Digest

Feature History for Configurable Screening Indicator

(Introduced as part of the SIP Gateway Support of RSVP and TEL URL feature)

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(8)T	This feature was integrated into this release.

Feature History for RADIUS Pre-authentication for Voice Calls

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for SIP - Enhanced Billing Support for SIP Gateways

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Feature History for SIP: Gateway HTTP Authentication Digest

Release	Modification
12.3(8)T	This feature was introduced.

Finding Support Information for Platforms and Cisco Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP AAA

All SIP AAA Features

- Establish a working IP network. For information about configuring IP, see *Cisco IOS IP Command Reference*, Release 12.3
- Configure VoIP. For information about configuring VoIP, see the following:
  - *Cisco IOS Voice Configuration Library*, Release 12.4T
  - *Cisco IOS Voice Command Reference*
  - "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms"
- Ensure that the gateway has voice functionality configured for SIP.

RADIUS Pre-authentication for Voice Calls Feature

- Ensure that you have an application that supports preauthentication.
- Set up preauthentication profiles and have them running on a RADIUS-based PPM server in your network.

- Enable gateway accounting using the gw-accounting command. All call-accounting information must be forwarded to the server that is performing preauthentication. Accounting stop packets must be sent to this server so that call billing is ended when calls are disconnected from the gateway. In addition, authentication and accounting start packets are needed to enable other features, such as virtual private dialup network (VPDN).

Note For information on setting up the preauthentication profiles, see the *Cisco IOS Security Command Reference*. For information on Cisco RPMS, see the Cisco Resource Policy Management System 2.0. For standards supporting RADIUS-based PPM servers, see RFC 2865, Remote Authentication Dial In User Service (RADIUS).

SIP: Gateway HTTP Authentication Digest Feature

- Implement a Cisco IOS SIP gateway that supports SIP.
- Implement a configuration that supports SIP.
- Implement an authentication configuration for the gateway to respond to authentication challenges for requests that it originates.

Restrictions for SIP AAA

All SIP AAA Features

- If Cisco Resource Policy Management System (RPMS) is used as the RADIUS-based PPM server, it must be Version 2.0 or a later release.
- In SIP environments, if you want the Cisco SIP proxy server to generate preauthentication queries, you must run Cisco SPS 2.0 or a later version.

SIP: Gateway HTTP Authentication Digest Via SIP UA Feature

- SIP Register is supported only on platforms with digital trunk type ports.

Information About SIP AAA

AAA features for SIP provide the following benefits:

- RADIUS preauthentication allows wholesalers to accept or reject calls to enforce SLAs before calls are connected, thereby conserving gateway resources.
- Call admission control prevents call connections when resources are unavailable.
- Extended dial plan features enable the call service type to be determined from preauthentication request data, simplifying dial plan entries.
- Universal gateways provide other specific benefits:

- Flexibility in deploying new services and adapting to changes in the business environment
- Cost savings through reduction of total number of ports required to provide different services
- Optimized utilization of access infrastructure by supporting more services during off-peak hours
- Flexibility in access network engineering by leveraging dial infrastructure to handle both dial and voice

To configure AAA features for SIP, you should understand the following concepts:

RADIUS Pre-authentication for Voice Calls

This section explains how to configure the AAA RADIUS communication link between a universal gateway and a RADIUS-based PPM server for RADIUS preauthentication.

Information about an incoming call is relayed through the gateway to the RADIUS-based PPM server in the network before the call is connected. The RADIUS-based PPM server provides port policy management and preauthentication by evaluating the call information against contracted parameter levels in SLAs. If the call falls within SLA limits, the server preauthenticates the call and the universal gateway accepts it. If the server does not authorize the call, the universal gateway sends a disconnect message to the public network switch to reject the call. The available call information includes one or more of the following:

- DNIS number, also referred to as the called number.
- CLID number (calling line identification number), also referred to as the calling number.
- Call type, also referred to as the bearer capability.
- IP address of the originating domain.
- Interzone ClearToken (IZCT) information, which contains the origination gatekeeper zone name for intradomain calls or the origination domain border gatekeeper zone name for interdomain calls. Whenever IZCT information is available, it is used to preauthenticate leg-3 H.323 VoIP calls.

Note To enable IZCT, use the **security izct password** command on the gatekeeper. For multiple gatekeeper zones, use the **lrq forward-queries**command. For information on IZCT configuration, see *Inter-Domain Gatekeeper Security Enhancement*, Release 12.2(4)T.

A timer monitors the preauthentication query in case the RADIUS-based PPM server application is unavailable or slow to respond. If the timer expires before an acceptance or rejection is provided, the universal gateway rejects the call.

The RADIUS Pre-authentication for Voice Calls feature supports the use of RADIUS attributes that are configured in RADIUS preauthentication profiles to specify preauthentication behavior. These attributes can also be used, for instance, to specify whether subsequent authentication should occur and, if so, what authentication method should be used.

The commands in this section are used for both leg 1 calls (calls from a PSTN that enter an incoming, or originating, gateway) and leg 3 calls (calls that exit the IP network to an outgoing, or terminating, gateway). The use of optional commands depends on individual network factors.

Note Before configuring AAA preauthentication, you must make sure that the supporting preauthentication application is running on a RADIUS-based PPM server in your network, such as a Cisco RPMS. You must also set up preauthentication profiles on the RADIUS-based PPM server. For full information on AAA, see the *Cisco IOS Security Configuration Guide*.

The RADIUS Pre-authentication for Voice Calls feature provides the means to evaluate and accept or reject call setup requests for both voice and dial calls received at universal gateways. This process is known as preauthentication. The feature also optionally allows voice calls to bypass this evaluation.

With universal gateways, voice customers and dial customers contend for the same gateway resources. This competition can present problems for IP service wholesalers who lease their IP services to various customers such as Internet service providers (ISPs), Internet telephony service providers (ITSPs), and telephony application service providers (T-ASPs). Wholesalers need a way to implement and enforce with these customers service-level agreements (SLAs) that describe the levels of connectivity, performance, and availability that they guarantee to provide. The RADIUS Pre-authentication for Voice Calls feature allows a wholesaler to determine whether a call is within SLA limits before gateway resources are dedicated to terminating the call.

With RADIUS preauthentication enabled, end customers from over-subscribed service providers are prevented from consuming ports that exceed the number allotted to their service provider in its SLA. If the call is accepted in the preauthentication step, it proceeds to full dial authentication and authorization or to voice dial-peer matching and voice session application authentication and authorization.

RADIUS preauthentication uses a RADIUS-based port-policy management (PPM) server, such as the Cisco Resource Policy Management System (RPMS), to interpret and enforce universal PPM and preauthentication SLAs. RADIUS provides the communication link between the PPM server and universal gateways.

Customer profiles are defined in the PPM server with information from the SLA. Then, when a call is received at the universal gateway, the server determines which specific customer SLA policy to apply to the call on the basis of information associated with the call. For example, calls can be identified as either dial or voice on the basis of the called number (also called the dialed number identification service number or DNIS). Then the PPM server might be set up to allow only a certain number of dial calls. When a new dial call is received, it is rejected if adding it to the count makes the count exceed the number of dial calls stipulated in the SLA.

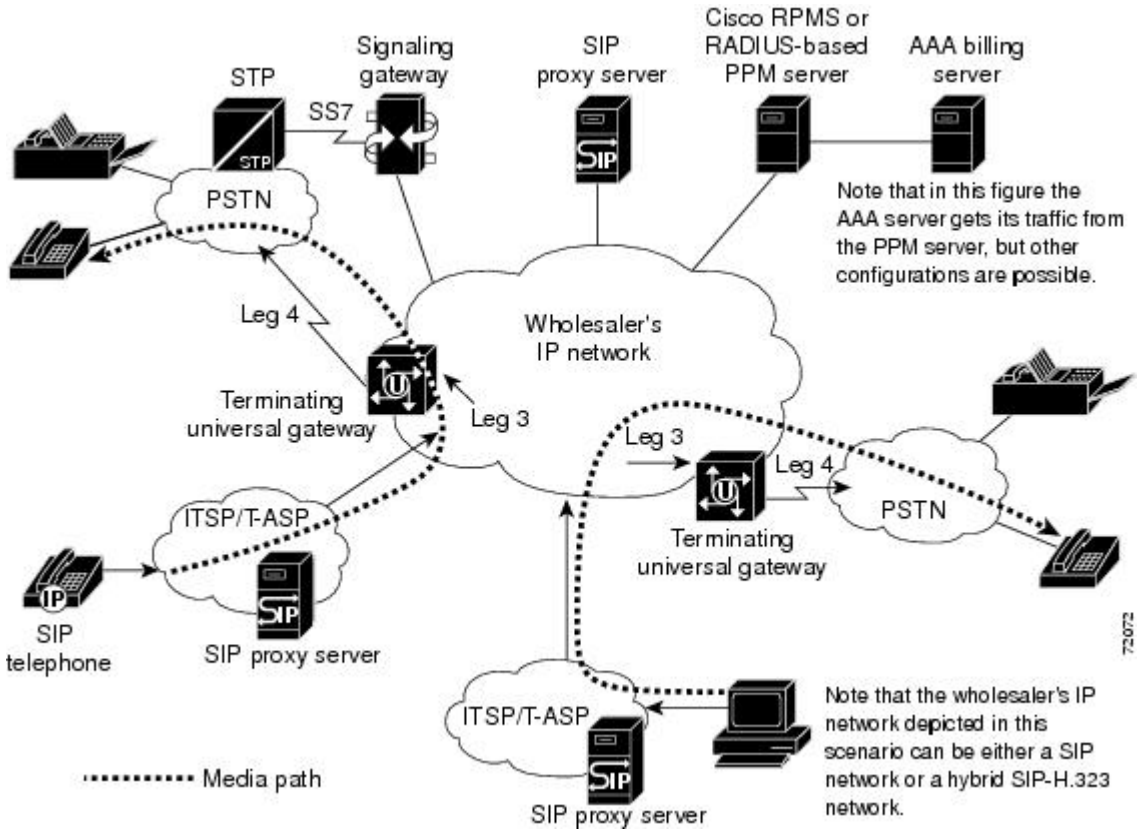
Calls that are accepted by the PPM server continue with their normal call setup sequences after preauthentication. The response from the PPM server is returned to the calling entity--such as an ISDN or SIP call signaling interface--which then proceeds with the regular call flow. Calls that are rejected by the PPM server follow the given call model and apply the error codes or rejection reasons that are specified by the signaling entity.

SIP-Based Voice Termination

In the figure below, a voice call from a SIP telephone or SIP terminal is sent from an ITSP to a wholesaler.The Cisco SIP proxy server (Cisco SPS) chooses the appropriate universal gateway to which the SIP INVITE is forwarded, on the basis of its own routing mechanism. In Step 3, Cisco SPS makes a preauthentication query to the RPMS-based PPM server. Cisco SPS locks out calls that are rejected by the RPMS-based PPM server. In Step 5, the universal gateway makes a preauthentication reservation request to the RPMS-based PPM server, which locks in the resources to handle the call.

Note This scenario requires Cisco SPS 2.0.

Figure 62: SIP-Based Voice Termination



The call flow is as follows:

1. A SIP INVITE is sent from an end user's PC to an ITSP SIP proxy server.
2. The ITSP's SIP proxy server forwards the SIP INVITE to a Cisco SPS at a wholesaler or ISP.
3. Preauthentication--The Cisco SPS sends a preauthentication query to the RADIUS-based PPM server, which locates the appropriate SLA and makes sure that the call is within the SLA limits. If the call is outside the limits, the call is rejected and Cisco SPS responds to the sender with an "Error code 480 - Temporarily not available" message. Cisco SPS interaction with the RADIUS-based PPM server is optional and requires Cisco SPS version 2.0 or a later release. If you are not using Cisco SPS 2.0, the gateway makes the preauthentication query to the RADIUS-based PPM server if it has been configured to do so.
4. Gateway selection--If the preauthentication request is accepted, the Cisco SPS uses its routing logic to determine the appropriate terminating universal gateway to which it should forward the INVITE.
5. Call admission control--If the preauthentication request is accepted, the terminating universal gateway checks its configured call admission control limits. If the call is outside the limits, the call is rejected.
6. Authentication and authorization--The universal gateway reserves a port and sends an authentication, authorization, and accounting (AAA) accounting start packet to the RADIUS-based PPM server.
7. The connection between the caller and the universal gateway is completed (call leg 3).
8. The caller is connected to the PSTN (call leg 4).

9. Accounting stop--After the caller hangs up or is otherwise disconnected, the terminating universal gateway issues an accounting stop packet to the RADIUS-based PPM server. The PPM server uses the accounting stop packet to clear out the count for that call against the SLA.

SIP - Enhanced Billing Support for Gateways

This section describes the SIP - Enhanced Billing Support for Gateways feature. The feature describes the changes to authentication, authorization, and accounting (AAA) records and the Remote Authentication Dial-In User Service (RADIUS) implementations on Cisco SIP gateways. These changes were introduced to provide customers and partners the ability to effectively bill for traffic transported over SIP networks.

Username Attribute

The username attribute is included in all AAA records and is the primary means for the billing system to identify an end user. The password attribute is included in authentication and authorization messages of inbound VoIP call legs.

For most implementations, the SIP gateway populates the username attribute in the SIP INVITE request with the calling number from the FROM: header, and the password attribute with null or with data from an IVR script. If a Proxy-Authorization header exists, it is ignored. The **aaa username** command determines the information with which to populate the username attribute.

Within the Microsoft Passport authentication service that authenticates and identifies users, the passport user ID (PUID) is used. The PUID and a password are passed from a Microsoft network to the Internet telephony service provider (ITSP) network in the Proxy-Authorization header of a SIP INVITE request as a single, base-64 encoded string. For example,

```
Proxy-Authorization: basic MDAwMzAwMDA4MDM5MzJlNjou
```

The **aaa username** command enables parsing of the Proxy-Authorization header; decoding of the PUID and password; and populating of the PUID into the username attribute, and the decoded password into the password attribute. The decoded password is generally a "." because a Microsoft Network (MSN) authenticates users prior to this point. For example,

```
Username = "123456789012345"
Password = "Z\335\304\326KU\037\301\261\326GS\255\242\002\202"
```

The password in the example above is an encrypted "." and is the same for all users.

SIP Call ID

From the Call ID header of the SIP INVITE request, the SIP Call ID is extracted and populated in Cisco vendor-specific attributes *(* VSA) as an attribute value pair *call-id=string* . The value pair can be used to correlate RADIUS records from Cisco SIP gateways with RADIUS records from other SIP network elements for example, proxies.

Note For complete information on this attribute value pair, see the *RADIUS Vendor-Specific Attributes Voice Implementation Guide* .

Session Protocol

Session Protocol is another attribute value pair that indicates whether the call is using SIP or H.323 as the signaling protocol.

Note For complete information on this attribute value pair, see the *RADIUS Vendor-Specific Attributes Voice Implementation Guide* .

Silent Authentication Script

As part of the SIP - Enhanced Billing Support for SIP Gateways feature, a Tool Command Language (Tcl) Interactive Voice Response (IVR) 2.0 Silent Authorization script has been developed. The Silent Authorization script allows users to be authorized without having to separately enter a username or password into the system. The script automatically extracts the passport user ID (PUID) and password from the SIP INVITE request, and then authenticates that information through RADIUS authentication and authorization records. The script is referred to as *silent* since neither the caller or called party hears any prompts.

Note You can upgrade to the latest script version through the CCO Software Center. You can download the app_passport_silent.2.0.0.0.tcl script from http://www.cisco.com/cgi-bin/tablebuild.pl/tclware . You must be a registered CCO user to log in and access these files.

- For information regarding Tcl IVR API 2.0, see the *Tcl IVR API Version 2.0 Programmer's Guide.*

Developers using the Tcl Silent Authorization script may be interested in joining the Cisco Developer Support Program. This program provides you with a consistent level of support that you can depend on while leveraging Cisco interfaces in your development projects. It also provides an easy process to open, update, and track issues through Cisco.com. The Cisco web-site is a key communication vehicle for using the Cisco Online Case tracking tool. A signed Developer Support Agreement is required to participate in this program. For more details, and access to this agreement, please visit us at http://www.cisco.com/en/US/products/svcs/ps3034/ps5408/ps5418/serv_home.html or contact developer-support@cisco.com.

Configurable Screening Indicator

Screening Indicator (SI) is a signaling-related information element found in octet 3a of the ISDN SETUP message that can be used as an authorization mechanism for incoming calls. The Tcl IVR 2.0 command set allows SIP terminating gateways to assign a specific value to the screening indicator through the use of Tcl scripts.

The screening indicator can contain four possible values:

- User provided, not screened
- User provided, verified and passed
- User provided, verified and failed
- Network provided

Note In all scenarios, gateway accounting must be enabled, and all call-accounting information must be forwarded to the server that is performing preauthentication. Accounting stop packets must be sent to this server so that call billing is ended when calls are disconnected from the gateway. In addition, authentication and accounting start packets are needed to enable other features, such as virtual private dial-up network (VPDN).

- For information on using Tcl IVR scripts to set and retrieve screening indicators, see the *Tcl IVR API Version 2.0 Programmer's Guide*

SIP Gateway HTTP Authentication Digest

The SIP: Gateway HTTP Authentication Digest feature implements authentication using the digest access on the client side of a common SIP stack. The gateway responds to authentication challenges from an authenticating server, proxy server, or user-agent server (UAS). This feature also maintains parity between the Cisco gateways, proxy servers, and SIP phones that already support authentication.

Feature benefits include the following:

- A SIP gateway is able to respond to authentication challenges from authenticating proxy servers or user-agent servers (UASs). The authentication method supported is digest authentication. Although digest authentication is not the best method, it provides a basic level of security.

Note The UAS challenges with a 401 response and the proxy server with a 407 response. It tries to find authentication credentials appropriate to the realm issuing the challenge and response. A gateway can handle authentication challenges from both the proxy server and UAS.

- Registration of the destination patterns on POTS dial peers extends to all PSTN interfaces.

Note The proxy server previously performed authentication only with the SIP phones.

The SIP Survivable Remote Site Telephony (SRST) feature in an earlier release added support to register E.164 numbers for foreign exchange stations (FXSs) (analog telephone voice ports) and extended foreign exchange stations (IP phone virtual voice ports) to an external SIP registrar. This feature extends that functionality for the gateway to register numbers configured on PSTN trunks such as PRI pipes.

Digest Access Authentication

SIP provides a stateless challenge-response mechanism for authentication based on digest access. A UAS or proxy server receiving a request challenges the initiator of the request to provide its identity. The user-agent client (UAC) generates a response by performing a message digest 5 (MD5) checksum on the challenge and its password. The response is passed back to the challenger in a subsequent request.

There are two modes of authentication:

- Proxy-server authentication
- UAS authentication

This feature also supports multiple proxy authentication on the gateway. The gateway can respond to up to five different authentication challenges in the signaling path between gateway as UAC and a UAS.

UAC-to-UAS Authentication

When the UAS receives a request without credentials from a UAC, it challenges the originator to provide credentials by rejecting the request with a “401 Unauthorized” response that includes a WWW-Authenticate header. The header field value consists of arguments applicable to digest scheme, as follows:

- realm--A string to be displayed to users so they know which username and password to use.
- nonce--A server-specified data string that should be uniquely generated each time a 401 response is made.

In addition, the header field may contain the following optional arguments:

- opaque--A string of data, specified by the server, that should be returned by the client unchanged in the Authentication header of subsequent requests with URIs in the same protection space.
- stale--A flag, indicating that the previous request from the client was rejected because the nonce value was stale.
- algorithm--A string indicating a pair of algorithms used to produce the digest and a checksum.
- qop-options--A string of one or more tokens indicating the “quality of protection” values supported by the server.
- auth-param--Directive that allows for future extensions.

The UAC reoriginates the request with proper credentials in the Authorization header field. The Authorization header field value consists of authentication information and arguments:

- username--User’s name in specified realm. This value is taken from the configuration, either at the dial-peer or the global level.
- digest-uri--Same as request uri of the request.
- realm, nonce-- From WWW-Authenticate header.

Message digest 5 (MD5) is computed as follows:

```
MD5(concat(MD5(A1),(unquoted)nonce-value":"nc-value":"
(unquoted)cnonce-value ":"(unquoted)qop-value":"MD5(A2)))
```

where A1 = (unquoted) username-value ":" (unquoted) realm-value ":" password

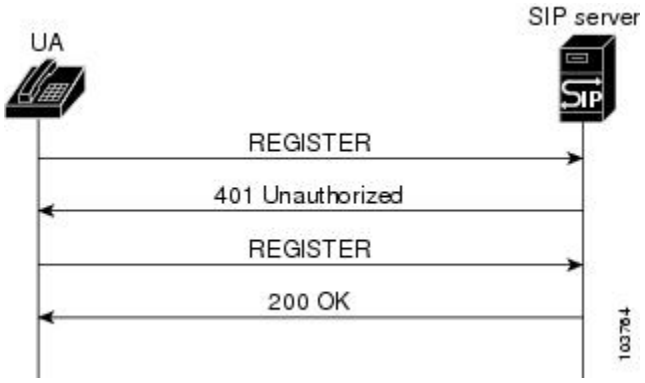
```
A2 = Method ":" request-uri if qop is "auth"
& A2 = Method ":" request-uri ":" MD5(entity-body) if qop is "auth-int".
```

- The nc-value is the hexadecimal count of the number of requests (including the current request that the client has sent with the nonce value in this request.
- The cnonce-value is an opaque string provided by client for mutual authentication between client and server.
- The qop-value is quality of protection directive, “auth” or “auth-int”.

UAC-to-UAS Call Flow with Register Message

In this call flow (see the figure below), the UA sends a Register message request without the Authorization header and receives a 401 status code message response challenge from the SIP server. The UA then resends the request including the proper credentials in the Authorization header.

Figure 63: UA-to-UAS Call Flow with Register Message



The UA sends a Register message request to the SIP server with the CSeq initialized to 1:

```
REGISTER sip:172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK200B
From: "36602" <sip:36602@172.18.193.120>;tag=98AS-87RT
To: <sip:36602@172.18.193.187>
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
User-Agent: Cisco-SIPGateway/IOS-12.x
CSeq: 1 REGISTER
Contact: <sip:36602@172.18.193.120:5060>;user=phone
Expires: 60
Content-Length: 0
```

The SIP server responds with a 401 Unauthorized challenge response to the UA:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK200B
From: "36602" <sip:36602@172.18.193.120>;tag=98AS-87RT
To: <sip:36602@172.18.193.187>;tag=3046583040568302
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="example.com", qop="auth",
nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

The UA resends a Register message request to the SIP server that includes the authorization and increments the CSeq:

```
REGISTER sip:172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK1DEA
From: "36602" <sip:36602@172.18.193.120>;tag=98AS-89FD
To: <sip:36602@172.18.193.187>
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
User-Agent: Cisco-SIPGateway/IOS-12.x
Authorization: Digest username="36602", realm="example.com",
nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="", uri="sip:172.18.193.187",
response="dfe56131d1958046689d83306477ecc"
CSeq: 2 REGISTER
```

```
Contact: <sip:36602@172.18.193.120:5060>;user=phone
Expires: 60
Content-Length: 0
```

The SIP server responds with a 200 OK message response to the UA:

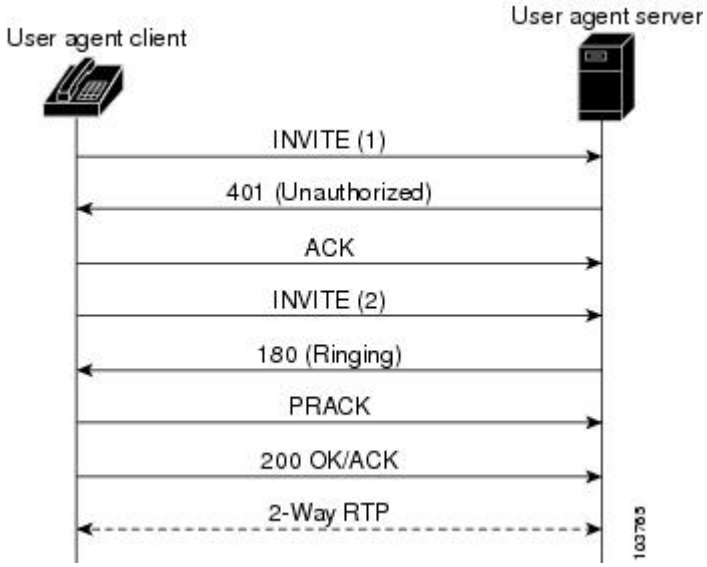
```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK1DEA
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
From: "36602" <sip:36602@172.18.193.120>;tag=98AS-89FD
To: <sip:36602@172.18.193.187>;tag=1q92461294
CSeq: 2 REGISTER
Contact: <sip:36602@172.18.193.120:5060>;expires="Wed, 02 Jul 2003 18:18:26 GMT"
Expires: 60
Content-Length: 0
```

Note A SIP server can challenge any request except ACK and CANCEL request messages, because an ACK message request does not take any response and a CANCEL message request cannot be resubmitted. The UA uses the same credentials in an ACK message request as in an INVITE message request.

UAC-to-UAS Call Flow with INVITE Message

In this call flow (see the figure below), the UAC sends an INVITE message request to a UAS without proper credentials and is challenged with a 401 Unauthorized message response. A new INVITE message request is then sent, containing the correct credentials. Finally, the call is completed.

Figure 64: UAC-to-UAS Call Flow with INVITE Message



The UAS challenges the UAC to provide user credentials by issuing a 401 Unauthorized message response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK45TGN
From: "36602" <sip:36602@172.18.193.120>;tag=98AS-87RT
To: <sip:36602@172.18.193.187>;tag=3046583040568302
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
```

```
CSeq: 101 INVITE
WWW-Authenticate: Digest realm="example.com", qop="auth",
nonce="ea9c8e8809345gf1cec4341ae6cgh5a359", opaque=""
Content-Length: 0
```

The UAC resubmits the request with proper credentials in the Authorization header:

```
INVITE sip:36601@172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK8DF8H
From: "36602"<sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
CSeq: 102 INVITE
Authorization: Digest username="36602", realm="example.com",
nonce="ea9c8e8809345gf1cec4341ae6cgh5a359", opaque="", uri="sip:36601@172.18.193.187",
response="42ce3cef44b22f50c02350g6071bc8"
.
.
.
```

The UAC uses the same credentials in subsequent requests in that dialog:

```
PRACK sip:36601@172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK8YH5790
From: "36602"<sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>;tag=AG09-92315
Call-ID: A9EEC728-495E11D6-8003AD63-F55A9C4
CSeq: 103 PRACK
Authorization: Digest username="36602", realm="example.com",
nonce="ea9c8e8809345gf1cec4341ae6cgh5a359", opaque="", uri="sip:36601@172.18.193.187",
response="42ce3cef44b22f50c02350g6071bc9"
Content-Length: 0
```

Proxy-Server-to-UA Authentication

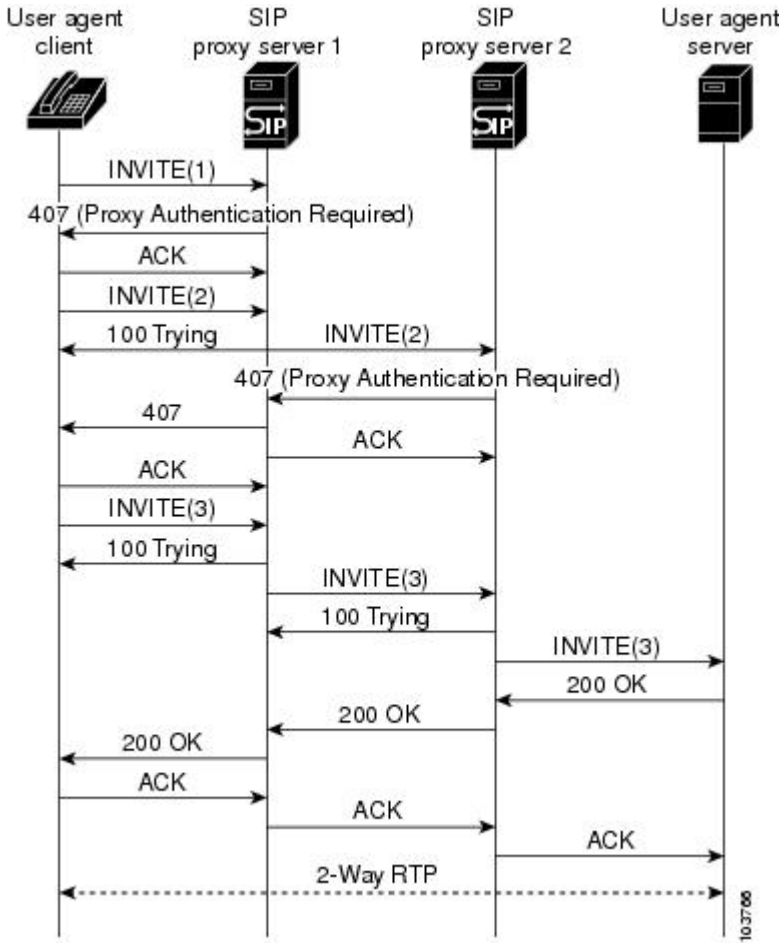
When a UA submits a request to a proxy server without proper credentials, the proxy server authenticates the originator by rejecting the request with a 407 message response (Proxy Authentication Required) and includes a Proxy-Authenticate header field value applicable to the proxy server for the requested resource. The UAC follows the same procedure mentioned in "UAC-to-UAS_Authentication" to get proper credentials for the realm and resubmits the request with the credentials in the Proxy-Authorization header.

Note realm--A string to be displayed to users so they know which username and password to use.

Proxy Server to UA Authentication Call Flow

In this call flow the UAC completes a call to user a UAS by using two proxy servers (PS 1 or PS 2, (see the figure below). The UAC has valid credentials in both domains. Because the initial INVITE message request does not contain the Authorization credentials proxy server 1 requires, a 407 Proxy Authorization message response containing the challenge information is sent. A new INVITE message request containing the correct credentials is then sent and the call proceeds after proxy server 2 challenges and receives valid credentials.

Figure 65: Proxy-Server-to-UA Call Flow



Proxy server 1 challenges the UAC for authentication:

```
SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK207H
From: <sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>;tag=929523858000835
Call-ID: D61E40D3-496A11D6-80070030-9426ED30@172.18.193.120
CSeq: 101 INVITE
Proxy-Authenticate: Digest realm="proxy1.example.com", qop="auth”,
nonce="wf84f1cczx41ae6cbeaea9ce88d359", opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

The UAC responds by resending the INVITE message request with authentication credentials. The same Call-ID is used, so the CSeq is increased.

```
INVITE sip:36601@172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bKEE1
From: <sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>
Call-ID: D61E40D3-496A11D6-80070030-9426ED30@172.18.193.120
CSeq: 102 INVITE
Proxy-Authorization: Digest username="36602", realm="proxy1.example.com",
nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359", opaque="", uri="sip:36601@172.18.193.187",
```

```
response="42ce3cef44b22f50c6a6071bc8"
Contact: <sip:172.18.193.120:5060>
.
.
.
```

The proxy server 2 challenges the UAC INVITE message request for authentication which is the 407 authentication message response that is forwarded to the UAC by proxy server 1.

```
SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bKEE1
From: <sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>;tag=083250982545745
Call-ID: D61E40D3-496A11D6-80070030-9426ED30@172.18.193.120
Proxy-Authenticate: Digest realm="proxy2.example.com", qop="auth",
nonce="c1e22c41ae6cbe5ae983a9c8e88d359", opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

The UAC responds by resending the INVITE message request with authentication credentials for proxy server 1 and proxy server 2.

```
INVITE sip:36601@172.18.193.187:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK8GY
From: <sip:36602@172.18.193.120>;tag=50EB48-383
To: <sip:36601@172.18.193.187>
Call-ID: D61E40D3-496A11D6-80070030-9426ED30@172.18.193.120
CSeq: 103 INVITE
Proxy-Authorization: Digest username="36602", realm="proxy1.example.com",
nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359", opaque="", uri="sip:36601@172.18.193.187",
response="42ce3cef44b22f50c6a6071bc8"
Proxy-Authorization: Digest username="36602", realm="proxy2.example.com",
nonce="c1e22c41ae6cbe5ae983a9c8e88d359", opaque="", uri="sip:36601@172.18.193.187",
response="f44ab22f150c6a56071bce8"
.
.
.
```

Extending SIP Register Support on Gateway

The SIP: Gateway HTTP Authentication Digest feature enhances functionality for Cisco IOS SIP gateway to Register all addresses specified by destination patterns in operational POTS dial-peers for all ports. This provides customer flexibility to register and authenticate users behind a private branch exchange (PBX) connected to the gateway through a PRI interface. There is no change in the way the gateway with foreign-exchange-station (FXS) ports registers individual E.164 addresses.

This feature leverages dial peers to create granularity for registration and authentication. However, the dial peers can be created with wildcards (for example: .919T , where terminator [T] makes the gateway wait until the full dial-string is received.) and a range of numbers (for example: .919392... , where ... indicates numbers in the range 0000 to 9999). Such destination patterns are registered with a single character wildcard in the user portion of To and Contact headers. The table below shows how the various types of gateway dial plans map to its registration. You need to modify the proxy/registrar behavior to correctly route calls for wildcard patterns or destination pattern with a range. Proxy server or registrars that do not match a wildcard patterns or destination pattern with a range should be ignored for that specific request.

Table 37: SIP Cisco IOS Gateway Dial Peer Mapping to Register

Cisco IOS SIP GW Configuration	Corresponding Register
dial-peer voice 919 pots destination-pattern 919..... port 0:D	REGISTER sip:proxy.example.com SIP/2.0 To: <sip:919.....@172.18.193.120> From: <sip:172.18.192.120>;tag=ABCD Contact: <sip:919.....@172.18.193.120>;user=phone
dial-peer voice 555 pots destination-pattern 555T port 0:D	REGISTER sip:proxy.example.com SIP/2.0 To: <sip:555*@172.18.193.120> From: <sip:172.18.192.120>;tag=ABCD Contact: <sip:555*@172.18.193.120>;user=phone
dial-peer voice 5550100 pots destination-pattern 5550100 port 0:D	REGISTER sip:proxy.example.com SIP/2.0 To: <sip:5550100@172.18.193.120> From: <sip:5550100@172.18.192.120>;tag=ABCD Contact: <sip:5550100@172.18.193.120>;user=phone

How to Configure SIP AAA Features

Configuring RADIUS Pre-authentication for Voice Calls

Configure a RADIUS Group Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *groupname*
5. **server** *ip-address* [**auth-port** *port*] [**acct-port** *port*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting access-control model.
Step 4	**aaa group server radius** *groupname* **Example:** `Router(config-sg-radius)# aaa group server radius radgroup1`	(Optional) Groups different RADIUS server hosts into distinct lists and distinct methods. The argument is as follows: • groupname--Character string used to name the group of servers.
Step 5	**server** *ip-address* [**auth-port** *port*] [**acct-port** *port*] **Example:** `Router(config-sg-radius)# server 172.20.0.1 auth-port 1000 acct-port 1001`	(Required if the aaa group server command is used) Configures the IP address of the RADIUS server for the group server. Keywords and arguments are as follows: • ip-address--IP address of the RADIUS server host. • auth-post port-number--UDP destination port for authentication requests. The host is not used for authentication if this value is set to 0. Default: 1645. • acct-port port-number--UDP destination port for accounting requests. The host is not used for accounting services if this value is set to 0. Default: 1646.
Step 6	**exit** **Example:** `Router(config-sg-radius)# exit`	Exits the current mode.

Configure Access and Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authentication login h323 group** *groupname*
4. **aaa authentication ppp default group** *groupname*
5. **aaa authorization exec** *list-name* **group** *groupname*
6. **aaa authorization network default group** {**radius** | **rpms**} **if-authenticated**
7. **aaa authorization reverse-access default local**

8. **aaa accounting suppress null-user-name**
9. **aaa accounting send stop-record authentication failure**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa authentication login h323 group** *groupname* **Example:** `Router(config)# aaa authentication billson h323 group 123`	Sets authentication, authorization, and accounting at login. Keywords and arguments are as follows: • h323--Use H.323 for authentication. • group groupname--Use a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius or aaa group server tacacs+ command.
Step 4	**aaa authentication ppp default group** *groupname* **Example:** `Router(config)# aaa authentication ppp default group 123`	(Required for PPP dial-in methods that are to be used with preauthentication) Specifies one or more authentication, authorization, and accounting authentication methods for use on serial interfaces running PPP. Keywords and arguments are as follows: • default--Use the listed authentication methods that follow this argument as the default list of methods when a user logs in. • group groupname--Use a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius or aaa group server tacacs+ command.
Step 5	**aaa authorization exec** *list-name* **group** *groupname* **Example:** `Router(config)# aaa authorization exec billson group 123`	(Optional) Sets parameters that restrict user access to a network. Keywords and arguments are as follows: • exec--Run authorization to determine if the user is allowed to run an EXEC shell. This facility might return user profile information such as autocommand information. • list-name--Character string used to name the list of authorization methods.

	Command or Action	Purpose
		• group groupname--Use a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius or aaa group server tacacs+ command.
Step 6	**aaa authorization network default group** {**radius** | **rpms**} **if-authenticated** **Example:** `Router(config)# aaa authorization network default group radius if-authenticated`	(Optional) Sets parameters that restrict user access to a network. Keywords are as follows: • network--Run authorization for all network-related service requests, including Serial Line Internet Protocol, Point-to-Point Protocol, PPP network Control Programs, and Apple Talk Remote Access. • default--Use the listed authorization methods that follow this argument as the default list of methods for authorization. • group radius--Use a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius or aaa group server tacacs+ command. • group rpms--Use a subset of RADIUS or TACACS+ servers for authentication as defined by the aaa group server radius or aaa group server tacacs+ command. • if-authenticated--Allow the user to access the requested function if the user is authenticated.
Step 7	**aaa authorization reverse-access default local** **Example:** `Router(config)# aaa authorization reverse-access default local`	(Optional) Configures a network access server to request authorization information from a security server before allowing a user to establish a reverse Telnet session. Keywords are as follows: • default--Use the listed authorization methods that follow this argument as the default list of methods for authorization. • local--Use the local database for authorization.
Step 8	**aaa accounting suppress null-user-name** **Example:** `Router(config)# aaa accounting suppress null-username`	(Optional) Prevents the Cisco IOS software from sending accounting records for users whose username string is NULL.
Step 9	**aaa accounting send stop-record authentication failure** **Example:** `Router(config)# aaa accounting send stop-record authentication failure`	(Required if using Cisco RPMS) Generates account "stop" records for users who fail to authenticate at login or during session negotiation.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configure Accounting

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting delay-start**
4. **aaa accounting update** [**periodic** *number*]
5. **aaa accounting exec default start-stop group** *groupname*
6. **aaa accounting exec** *list-name* **start-stop group** *groupname*
7. **aaa accounting network default start-stop group** *groupname*
8. **aaa accounting connection h323 start-stop group** *groupname*
9. **aaa accounting system default start-stop group** *groupname*
10. **aaa accounting resource default start-stop-failure group** *groupname*
11. **gw-accounting aaa**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa accounting delay-start** **Example:** `Router(config)# aaa accounting delay-start`	(Optional) Delays generation of accounting "start" records until the user IP address is established. For a complete explanation of the aaa accounting command, see the *Cisco IOS Security Command Reference*
Step 4	**aaa accounting update** [**periodic** *number*] **Example:** `Router(config)# aaa accounting update periodic 30`	(Optional) Enables periodic interim accounting records to be sent to the accounting server. Keyword and argument are as follows:

	Command or Action	Purpose
		• periodic number--An interim accounting record is sent to the accounting server periodically, as defined by the argument number (in minutes).
Step 5	**aaa accounting exec default start-stop group** *groupname* **Example:** `Router(config)# aaa accounting exec default start-stop group joe`	(Optional) Enables authentication, authorization, and accounting of requested services for billing or security purposes when you use RADIUS or TACACS+ and want to run a shell session. Keywords and arguments are as follows: • **exec** --Run accounting for EXEC shell session. This keyword might return profile information such as what is generated by the autocommand command. • **default** --Use the listed accounting methods that follow this argument as the default list of methods for accounting services. • **start-stop** --Send a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting server. • **group** *groupname* --Use a subset of RADIUS or TACACS+ servers for accounting as defined by the server group groupname.
Step 6	**aaa accounting exec** *list-name* **start-stop group** *groupname* **Example:** `Router(config)# aaa accounting exec joe start-stop group tacacs+`	(Optional) Enables authentication, authorization, and accounting of requested services for billing or security purposes when you use RADIUS or TACACS+ and want to specify method names. Keywords and arguments are as follows: • **exec** --Run accounting for EXEC shell session. This keyword might return profile information such as what is generated by the autocommand command. • *list-name* --Character string used to name the list of at least one of the accounting methods. • **start-stop** --Send a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting server. • **group** *groupname* --Use a subset of RADIUS or TACACS+ servers for accounting as defined by the server group groupname.

	Command or Action	Purpose
Step 7	**aaa accounting network default start-stop group** *groupname* **Example:** `Router(config)# aaa accounting network default start-stop group tacacs+`	(Required for PPP dial-in methods that are to be used for preauthentication) Enables authentication, authorization, and accounting of requested network services for billing and security purposes when you use RADIUS or TACACS+. Keywords and arguments are as follows: • **network** --Run accounting for all network-related service requests, including Serial Line Internet Protocol, Point-to-Point Protocol, PPP Network Control Protocols, and Apple Talk Remote Access Protocol. • **default** --Use the listed accounting methods that follow this argument as the default list of methods for accounting services. • **start-stop** --Send a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting service. • **group** *groupname* --At least one of the following: • **group radius**--Use the list of all RADIUS servers for authentication as defined by the **aaa group server radius** command. • **group-tacacs+**--Use the list of all TACACS+ servers for authentication as defined by the **aaa group server tacacs+** command. • **group** *groupname*--Use a subset of RADIUS or TACACS+ servers for accounting as defined by the server group groupname.
Step 8	**aaa accounting connection h323 start-stop group** *groupname* **Example:** `Router(config)# aaa accounting connection h323 start-stop group tacacs+`	(Required for voice call accounting) Enables accounting, accounting of requested services for billing and security purposes when you use RADIUS or TACACS+ and want connection information. Keywords and arguments are as follows: • connection--Provide information about all outbound connections made from the network access server, such as Telnet, LAT, TN3270, PAD, and rlogin. • h323--Character string used to name the list of at least one of the accounting methods. Uses h323 for accounting. • start-stop--Send a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record

	Command or Action	Purpose
		is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting service. • group groupname--At least one of the following: • group radius--Use the list of all RADIUS servers for authentication as defined by the aaa group server radius command. • group-tacacs+--Use the list of all TACACS+ servers for authentication as defined by the aaa group server tacacs+ command. • group groupname--Use a subset of RADIUS or TACACS+ servers for accounting as defined by the server group groupname.
Step 9	**aaa accounting system default start-stop group** *groupname* **Example:** `Router(config)# aaa accounting system default start-stop group tacacs+`	(Optional) Enables accounting, accounting of requested services for billing and security purposes you when use RADIUS or TACACS+ and want system-level event accounting. Keywords and arguments are as follows: • system--Perform accounting for all system-level events not associated with users, such as reloads. • default--Use the listed accounting methods that follow this argument as the default list of methods for accounting services. • start-stop--Send a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting service. • group groupname--At least one of the following: • group radius--Use the list of all RADIUS servers for authentication as defined by the aaa group server radius command. • group-tacacs+--Use the list of all TACACS+ servers for authentication as defined by the aaa group server tacacs+ command. • group groupname--Use a subset of RADIUS or TACACS+ servers for accounting as defined by the server group groupname.
Step 10	**aaa accounting resource default start-stop-failure group** *groupname* **Example:**	(Optional) Enables full resource accounting, which will generate both a "start" record at call setup and a "stop" record at call termination. Keywords and arguments are as follows:

	Command or Action	Purpose
	`Router(config)# aaa accounting resource default start-stop-failure group tacacs+`	• default--Use the listed accounting methods that follow this argument as the default list of methods for accounting services. • group groupname--Server group to be used for accounting services.Valid values are as follows: • string--Character string used to name a server group. • radius--Use list of all RADIUS hosts. • tacacs+--Use list of all TACACS+ hosts.
Step 11	**gw-accounting aaa** **Example:** `Router(config)# gw-accounting aaa`	Enables VoIP gateway-specific accounting and define the accounting method.
Step 12	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configure Preauthentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa preauth**
4. **group** {**radius** | *groupname*}
5. **clid** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
6. **ctype** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
7. **dnis** [**if-avail** | **required**] [**accept-stop**] [**password** *string*]
8. **dnis bypass** {*dnis-groupname*}
9. **filter voice**
10. **timeout leg3** *time*
11. **service-type call-check**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**aaa preauth** **Example:** `Router(config)# aaa preauth`	Enters AAA preauthentication configuration mode.
Step 4	**group** {**radius** | *groupname*} **Example:** `Router(config-preauth)# group radius`	Specifies the authentication, authorization, and accounting RADIUS server group to use for preauthentication. Keywords and arguments are as follows: • radius--Use a RADIUS server for authentication. • groupname--Name of the server group to use for authentication.
Step 5	**clid** [**if-avail** | **required**] [**accept-stop**] [**password** *string*] **Example:** `Router(config-preauth)# clid required`	(Optional) Preauthenticates calls based on the Calling Line Identification (CLID) number. Keywords and arguments are as follows: • if-avail--If the switch provides the data, RADIUS must be reachable and must accept the string in order for preauthentication to pass. If the switch does not provide the data, preauthentication passes. • required--The switch must provide the associated data, that RADIUS must be reachable, and that RADIUS must accept the string in order for preauthentication to pass. If these three conditions are not met, preauthentication fails. • accept-stop--Prevents subsequent preauthentication elements such as ctype or dnis from being tried once preauthentication has succeeded for a call element. • password string--Defines the password for the preauthentication element.
Step 6	**ctype** [**if-avail** | **required**] [**accept-stop**] [**password** *string*] **Example:** `Router(config-preauth)# ctype required`	(Optional) Preauthenticates calls on the basis of the call type. Keywords and arguments are as described above.

	Command or Action	Purpose
Step 7	**dnis** [**if-avail** | **required**] [**accept-stop**] [**password** *string*] **Example:** `Router(config-preauth)# dnis required`	(Optional) Preauthenticates calls on the basis of the Dialed Number Identification Server (DNIS) number. Keywords and arguments are as described above.
Step 8	**dnis bypass** {*dnis-groupname*} **Example:** `Router(config-preauth)# dnis bypass abc123`	(Optional) Specifies a group of DNIS numbers that will be bypassed for preauthentication. The argument is as follows: • *dnis-groupname* --Name of the defined DNIS group.
Step 9	**filter voice** **Example:** `Router(config-preauth)# filter voice`	(Optional) Specifies that voice calls bypass authentication, authorization, and account preauthentication.
Step 10	**timeout leg3** *time* **Example:** `Router(config-preauth)# timeout leg3 100`	(Optional) Sets the timeout value for a leg 3 AAA preauthentication request. The argument is as follows: • *time* --Timeout value for leg3 preauthentication, in ms. Range: 100 to 1000. Default: 100.
Step 11	**service-type call-check** **Example:** `Router(config-preauth)# service-type call-check`	(Optional) Identifies preauthentication requests to the AAA server.
Step 12	**exit** **Example:** `Router(config-preauth)# exit`	Exits the current mode.

Configure RADIUS Communications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}]
4. **radius-server retransmit** *retries*
5. **radius-server attribute 6 support-multiple**
6. **radius-server attribute 44 include-in-access-req**
7. **radius-server attribute nas-port format c**
8. **radius-server key** {**0** *string* | **7** *string* | *string*}
9. **radius-server vsa send accounting**
10. **radius-server vsa send authentication**

11. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*] [**alias** {*hostname* | *ip-address*}] **Example:** `radius-server host jimname`	Specifies a RADIUS server host. Keywords and arguments are as follows: • *hostname* --DNS name of the RADIUS server host. • *ip-address* --IP address of the RADIUS server host. • **auth-port** *port-number* --UDP destination port for authentication requests; the host is not used for authentication if set to 0. Default: 1645. • **acct-port** *port-number* --UDP destination port for accounting requests; the host is not used for accounting if set to 0. Default: 1646. • **timeout** --Time interval (in seconds) that the router waits for the RADIUS server to reply before retransmitting. This setting overrides the global value of the radius-server timeout command. Range: 1 to 1000. If no timeout value is specified, the global value is used. • **retransmit** *retries* --Number of times that a RADIUS request is resent to a server, if that server is not responding or responding slowly. This setting overrides the global setting of the radius-server retransmit command. Range: 1 to 100. If no retransmit value is specified, the global value is used. • **key** *string* --Authentication and encryption key used between the router and the RADIUS daemon running on this RADIUS server. This key overrides the global setting of the radius-server key command. If no key string is specified, the global value is used. The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius-server host command syntax.

	Command or Action	Purpose
		This is because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key. • alias--Allow up to eight aliases per line for any given RADIUS server.
Step 4	**radius-server retransmit** *retries* **Example:** `Router(config)# radius-server retransmit 1`	(Optional) Specifies the number of times the Cisco IOS software searches the list of RADIUS server hosts before giving up. The argument is as follows: • *retries* --Maximum number of retransmission attempts. Default: 3.
Step 5	**radius-server attribute 6 support-multiple** **Example:** `Router(config)# radius-server attribute 6 support-multiple`	(Optional) Sets an option for RADIUS Attribute 6 (Service-Type) values in a RADIUS profile. The keyword is as follows: • **support-multiple** --Support multiple service-type values in each RADIUS profile.
Step 6	**radius-server attribute 44 include-in-access-req** **Example:** `Router(config)# radius-server attribute 44 include-in-access-req`	Sends RADIUS attribute 44 (Accounting Session ID) in access request packets before user authentication (including requests for preauthentication). **Note** For information on RADIUS attributes, see the *Cisco IOS Security Command Reference*.
Step 7	**radius-server attribute nas-port format c** **Example:** `Router(config)# radius-server attribute nas-port format c`	(Required if using Cisco RPMS) Selects the NAS-Port format used for RADIUS accounting features.
Step 8	**radius-server key** {**0** *string* | **7** *string* | *string*} **Example:** `Router(config)# radius-server key ncmmekweisnaowkakskiiw`	(Optional) Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. Keywords and arguments are as follows: • 0 string--An unencrypted (cleartext) shared key as specified by string. • 7 string--A hidden shared key as specified by string. • string--The unencrypted (cleartext) shared key.
Step 9	**radius-server vsa send accounting** **Example:** `Router(config)# radius-server vsa send accounting`	(Optional) Configures the network access server to recognize and use vendor-specific attributes.

	Command or Action	Purpose
Step 10	**radius-server vsa send authentication** **Example:** `Router(config)# radius-server vsa send authentication`	(Optional) Configures the network access server to recognize and use vendor-specific attributes.
Step 11	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configuring SIP - Enhanced Billing Support for Gateways

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **aaa username** {**calling-name** | **proxy-auth**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**aaa username** {**calling-name**	**proxy-auth**} **Example:** `Router(config-sip-ua)# aaa username calling-name`	Determines the information to populate the username attribute for AAA billing records. Keywords are as follows: • **calling-number** --Use the FROM: header in the SIP INVITE (default value). This keyword is used in most implementations and is the default.

	Command or Action	Purpose
		• **proxy-auth** --Parse the Proxy-Authorization header. Decode the Microsoft Passport user ID (PUID) and password, and then populate the PUID into the username attribute and a "." into the password attribute. The username attribute is used for billing and the "." is used for the password, because the user has already been authenticated.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Gateway HTTP Authentication Digest

Configure SIP Gateway HTTP Authentication Digest Via Dial-Peer

Note This configuration sets up the feature as defined under the POTS dial peer.

- This feature is configured at the POTS dial peer and SIP user agent, with configuration at the dial peer taking precedence over that at the SIP user agent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **pots**
4. **authentication username** *username* **password** *password* [**realm** *realm*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **pots** **Example:** `Router(config)# dial-peer voice 100 pots`	Enters dial-peer configuration mode for the specified POTS dial peer.
Step 4	**authentication username** *username* **password** *password* [**realm** *realm*] **Example:** `Router(config-sip-ua)# authentication username user1 password password1 realm example.com`	Enters SIP digest authentication mode. Keywords and arguments are as follows: • username username--A string representing username of the user authenticating. • password password--A string representing password for authentication. • realm realm--A string representing the applicable credential.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configure SIP Gateway HTTP Authentication Digest Via SIP UA

Note You can configure this feature for a dial peer or globally, for all POTS dial peers, in SIP user-agent configuration mode. If authentication is configured in SIP user-agent configuration mode and on individual dial peers, the individual dial-peer configuration takes precedence.

Note
- SIP Register is supported only on platforms with digital trunk type ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**]
5. **authentication username** *username* **password** *password* [**realm** *realm*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**registrar** {**dns:** *address* | **ipv4:** *destination-address*} **expires** *seconds* [**tcp**] [**secondary**] **Example:** `Router(config-sip-ua)# registrar ipv4:10.1.1.6 expires 60`	Registers E.164 numbers on behalf of analog telephone voice ports (FXS) and IP phone virtual voice ports (EFXS) with an external SIP proxy or SIP registrar server. Keywords and arguments are as follows: • **dns:** *address* --Domain name server that resolves the name of the dial peer to receive calls. • **ipv4:** *destination address* --IP address of the dial peer to receive calls. • **expires** *seconds* --Default registration time, in seconds. • **tcp** --Transport layer protocol is TCP. UDP is the default. • **secondary** --Registration is with a secondary SIP proxy or registrar for redundancy purposes. **Note** When registrar is provisioned, the gateway sends out register with 1.. .
Step 5	**authentication username** *username* **password** *password* [**realm** *realm*] **Example:** `Router(config-sip-ua)# authentication username user1 password password1 realm example.com`	Enters SIP digest authentication mode. Keywords and arguments are as follows: • **username** *username* --A string representing username of the user authenticating. • **password** *password* --A string representing password for authentication. • **realm** *realm* --A string representing the applicable credential.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Verifying AAA Features for SIP

To verify AAA-feature configuration, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show call active voice**
2. **show radius statistics**
3. **show rpms-proc counters**
4. **show running-config**
5. **show sip-ua register status**

DETAILED STEPS

Step 1 show call active voice

Use this command to display call information for active voice calls. You can thus verify the username attribute.

The following sample output shows that the proxy-auth parameter is selected.

Example:

```
Router# show call active voice
Total call-legs: 2
   GENERIC:
  SetupTime=1551144 ms
    .
    . (snip)
    .
  ReceiveBytes=63006
  VOIP:
  ConnectionId[0x220A95B7 0x6B3611D5 0x801DBD53 0x8F65BA34]
    .
    . (snip)
    .
  CallerName=
  CallerIDBlocked=False
  Username=1234567890123456            <-- PUID from Proxy-Auth header
```

The following sample output shows that the calling-number parameter is selected.

Example:

```
Router# show call active voice
Total call-legs: 2
   GENERIC:
```

```
SetupTime=1587000 ms
  .
  . (snip)
  .
ReceiveBytes=22762
VOIP:
ConnectionId[0xF7C22E07 0x6B3611D5 0x8022BD53 0x8F65BA34]
  .
  . (snip)
  .
CallerName=
CallerIDBlocked=False
Username=1234                          <-- calling-number
```

Step 2 **show radius statistics**

Use this command to display RADIUS statistics for accounting and authentication packets.

Step 3 **show rpms-proc counters**

Use this command to display the number of leg 3 preauthentication requests, successes, and rejects.

Note Use the **clear rpms-proc counters** command to reset the counters that record the statistics that the **show rpms-proc counters** command displays.

Step 4 **show running-config**

Use this command to display the current configuration.

Step 5 **show sip-ua register status**

Use this command to verify SIP user-agent register status.

Example:

```
Router# show sip-ua register status
Line  peer    expires(sec)  registered
4001  20001   596           no
4002  20002   596           no
5100  1       596           no
9998  2       596           no
where:
line=phone number to register
peer=registration destination number
expires (sec)=amount of time, in seconds, until registration expires
registered=registration status
```

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the “General Troubleshooting Tips” section.

- Make sure that you can make a voice call.

- If the gateway does not respond to the authentication challenge, make sure that the user credentials for the appropriate domain have been configured.
- For the gateway to register destination patterns on the POTS dial peer, make sure that a registrar has been configured.
- Use the **debug aaa authentication** command to display high-level diagnostics related to AAA logins.
- Use the **debug cch323 preauth** command to enable debug tracing on the H.323 SPI for preauthentication.
- Use the **debug ccsip**family of commands to enable SIP debugging capabilities. In particular, use the following:
  - Use the **debug ccsip all** and **debug ccsip events**commands to display output specific to the SIP - Enhanced Billing Support for Gateways feature.
  - Use the **debug ccsip preauth** command to enable debug tracing on the SIP service provider interface (SPI) for preauthentication.
- Use the **debug radius** command to enable debug tracing of RADIUS attributes.
- Use the **debug rpms-proc preauth** command to enable debug tracing on the RPMS process for H.323 calls, SIP calls, or both H.323 and SIP calls.

Following is sample output for some of these commands:

Sample Output for the debug ccsip Command

```
Router# debug ccsip messages
*Oct 11 21:40:26.175://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
INVITE sip:5550123@172.18.193.187:5060 SIP/2.0 ! Invite request message (command sequence
101)
Via:SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK6ED
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>
Date:Fri, 11 Oct 2002 21:40:26 GMT
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
Supported:100rel,timer
Min-SE: 1800
Cisco-Guid:3787171507-3700953558-2147913662-199702180
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq:101 INVITE
Max-Forwards:70
Remote-Party-ID:"36602" <sip:36602@172.18.193.120>;party=calling;screen=no;privacy=off
Timestamp:1034372426
Contact:<sip:36602@172.18.193.120:5060>
Expires:180
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:244
v=0
o=CiscoSystemsSIP-GW-UserAgent 6603 1568 IN IP4 172.18.193.120
s=SIP Call
c=IN IP4 172.18.193.120
t=0 0
m=audio 17978 RTP/AVP 18 19
c=IN IP4 172.18.193.120
a=rtpmap:18 G729/8000
```

```
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
a=ptime:20
*Oct 11 21:40:26.179://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying ! 100 Trying response message (command sequence 101)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK6ED
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>
CSeq:101 INVITE
Content-Length:0
*Oct 11 21:40:26.179://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 407 Proxy Authentication Required ! 407 proxy authentication required response
message (command sequence 101)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK6ED
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=214b-70c4
CSeq:101 INVITE
Proxy-Authenticate:DIGEST realm="example.com", nonce="405729fe", qop="auth", algorithm=MD5
Content-Length:0
*Oct 11 21:40:26.183://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:5550123@172.18.193.187:5060 SIP/2.0 ! ACK request message (command sequence 101)
Via:SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK6ED
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=214b-70c4
Date:Fri, 11 Oct 2002 21:40:26 GMT
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
Max-Forwards:70
CSeq:101 ACK
Content-Length:0
*Oct 11 21:40:26.183://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
INVITE sip:5550123@172.18.193.187:5060 SIP/2.0 ! Invite message request (command sequence
102)
Via:SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK8BA
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>
Date:Fri, 11 Oct 2002 21:40:26 GMT
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
Supported:100rel,timer
Min-SE: 1800
Cisco-Guid:3787171507-3700953558-2147913662-199702180
User-Agent:Cisco-SIPGateway/IOS-12.x
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq:102 INVITE
Max-Forwards:70
Remote-Party-ID:"36602" <sip:36602@172.18.193.120>;party=calling;screen=no;privacy=off
Timestamp:1034372426
Contact:<sip:36602@172.18.193.120:5060>
Expires:180
Allow-Events:telephone-event
Proxy-Authorization:Digest
username="36602",realm="example.com",uri="sip:172.18.193.187",response="404ee07cc7c808cd04c97260efef5",nonce="405729fe",cnonce="AD4C1",qop=auth,algorithm=MD5,nc=00000001
Content-Type:application/sdp
Content-Length:244
v=0
o=CiscoSystemsSIP-GW-UserAgent 6603 1568 IN IP4 172.18.193.120
s=SIP Call
c=IN IP4 172.18.193.120
```

```
t=0 0
m=audio 17978 RTP/AVP 18 19
c=IN IP4 172.18.193.120
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
a=ptime:20
*Oct 11 21:40:26.187://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying ! 100 Trying response message (command sequence 102)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK8BA
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>
CSeq:102 INVITE
Content-Length:0
*Oct 11 21:40:26.439://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 180 Ringing ! 180 Ringing response message (command sequence 102)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK8BA
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
CSeq:102 INVITE
Server:CSCO/4
Contact:<sip:5550123@172.18.197.182:5060>
Record-Route:<sip:5550123@172.18.193.187:5060;maddr=172.18.193.187>
Content-Length:0
*Oct 11 21:40:28.795://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK ! 200 OK response message (command sequence 102)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK8BA
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
CSeq:102 INVITE
Server:CSCO/4
Contact:<sip:5550123@172.18.197.182:5060>
Record-Route:<sip:5550123@172.18.193.187:5060;maddr=172.18.193.187>
Content-Type:application/sdp
Content-Length:146
v=0
o=Cisco-SIPUA 21297 9644 IN IP4 172.18.197.182
s=SIP Call
c=IN IP4 172.18.197.182
t=0 0
m=audio 28290 RTP/AVP 18
a=rtpmap:18 G729/8000
*Oct 11 21:40:28.799://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:5550123@172.18.193.187:5060;maddr=172.18.193.187 SIP/2.0 ! ACK request message
(command sequence 102)
Via:SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK20A5
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
Date:Fri, 11 Oct 2002 21:40:26 GMT
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
Route:<sip:5550123@172.18.197.182:5060>
Max-Forwards:70
CSeq:102 ACK
Proxy-Authorization:Digest
username="3602",realm="sample.com",uri="sip:172.18.193.187",response="c86e13d76426b6f362c4f56834",cnonce="40529fe",nonce="949DEB",qop=auth,algorithm=MD5,nc=00000002
Content-Length:0
*Oct 11 21:40:32.891://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
```

```
Sent:
BYE sip:5550123@172.18.193.187:5060;maddr=172.18.193.187 SIP/2.0 ! BYE request message
(command sequence 103)
Via:SIP/2.0/UDP  172.18.193.120:5060;branch=z9hG4bK6AF
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
Date:Fri, 11 Oct 2002 21:40:26 GMT
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
User-Agent:Cisco-SIPGateway/IOS-12.x
Max-Forwards:70
Route:<sip:5550123@172.18.197.182:5060>
Timestamp:1034372432
CSeq:103 BYE
Reason:Q.850;cause=16
Proxy-Authorization:Digest
username="3602",realm="sample.com",uri="sip:172.18.193.187",response="9a6617d5982aeaf83cd49d82d12dd",cnonce="40529fe",cnonce="22EBE32",qop=auth,algorithm=MD5,nc=00000003
Content-Length:0
*Oct 11 21:40:32.895://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying ! 100 Trying response message (command sequence 103)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK6AF
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
CSeq:103 BYE
Content-Length:0
*Oct 11 21:40:32.963://-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK ! 200 OK response message (command sequence 103)
Via:SIP/2.0/UDP 172.18.193.120:5060;received=172.18.193.120;branch=z9hG4bK6AF
From:"36602" <sip:36602@172.18.193.120>;tag=3E948-4C5
To:<sip:5550123@172.18.193.187>;tag=003094c2e56a035d4326b6a1-292418c6
Call-ID:E35DBEB1-DC9811D6-80098FBE-BE736A4@172.18.193.120
CSeq:103 BYE
Server:CSCO/4
Content-Length:0
```

Sample Output of the debug ccsip events Command

The example shows how the Proxy-Authorization header is broken down into a decoded username and password.

```
Router# debug ccsip events
CCSIP SPI: SIP Call Events tracing is enabled
21:03:21: sippmh_parse_proxy_auth: Challenge is 'Basic'.
21:03:21: sippmh_parse_proxy_auth: Base64 user-pass string is 'MTIzNDU2Nzg5MDEyMzQ1Njou'.
21:03:21: sip_process_proxy_auth: Decoded user-pass string is '1234567890123456:.'.
21:03:21: sip_process_proxy_auth: Username is '1234567890123456'.
21:03:21: sip_process_proxy_auth: Pass is '.'.
21:03:21: sipSPIAddBillingInfoToCcb: sipCallId for billing records =
10872472-173611CC-81E9C73D-F836C2B6@172.18.192.19421:03:21: ****Adding to UAS Request table
```

Sample Output for the debug radius Command

```
Router# debug radius
Radius protocol debugging is on
Radius protocol brief debugging is off
Radius packet hex dump debugging is off
Radius packet protocol debugging is on
Radius packet retransmission debugging is off
Radius server fail-over debugging is off
```

```
Jan 23 14:30:25.421:RADIUS/ENCODE(00071EBF):acct_session_id:742769
Jan 23 14:30:25.421:RADIUS(00071EBF):sending
Jan 23 14:30:25.421:RADIUS:Send to unknown id 25 192.168.41.57:1812, Access-Request, len
179
Jan 23 14:30:25.421:RADIUS: authenticator 88 94 AC 32 89 84 73 6D - 71 00 50 6C D0 F8 FD
11
Jan 23 14:30:25.421:RADIUS: User-Name           [1]   9   "2210001"
Jan 23 14:30:25.421:RADIUS: User-Password       [2]   18  *
Jan 23 14:30:25.421:RADIUS: Vendor, Cisco       [26]  32
Jan 23 14:30:25.421:RADIUS: Cisco AVpair        [1]   26  "resource-service=reserve"
Jan 23 14:30:25.421:RADIUS: Service-Type        [6]   6   Call Check    [10]
Jan 23 14:30:25.421:RADIUS: Vendor, Cisco       [26]  19
Jan 23 14:30:25.421:RADIUS:  cisco-nas-port     [2]   13  "Serial6/0:0"
Jan 23 14:30:25.425:RADIUS: NAS-Port            [5]   6   6144
Jan 23 14:30:25.425:RADIUS: Vendor, Cisco       [26]  29
Jan 23 14:30:25.425:RADIUS:  Cisco AVpair       [1]   23  "interface=Serial6/0:0"
Jan 23 14:30:25.425:RADIUS: Called-Station-Id   [30]  9   "2210001"
Jan 23 14:30:25.425:RADIUS: Calling-Station-Id  [31]  9   "1110001"
Jan 23 14:30:25.425:RADIUS: NAS-Port-Type       [61]  6   Async [0]
Jan 23 14:30:25.425:RADIUS: NAS-IP-Address      [4]   6   192.168.81.101
Jan 23 14:30:25.425:RADIUS: Acct-Session-Id     [44]  10  "000B5571"
Jan 23 14:30:25.429:RADIUS:Received from id 25 192.168.41.57:1812, Access-Accept, len 20
Jan 23 14:30:25.429:RADIUS: authenticator 2C 16 63 18 36 56 18 B2 - 76 EB A5 EF 11 45 BE
F4
Jan 23 14:30:25.429:RADIUS:Received from id 71EBF
Jan 23 14:30:25.429:RADIUS/DECODE:parse response short packet; IGNORE
Jan 23 14:30:25.433:RADIUS/ENCODE(00071EBF):Unsupported AAA attribute start_time
Jan 23 14:30:25.433:RADIUS/ENCODE(00071EBF):Unsupported AAA attribute timezone
Jan 23 14:30:25.433:RADIUS/ENCODE:format unknown; PASS
Jan 23 14:30:25.433:RADIUS(00071EBF):sending
Jan 23 14:30:25.433:RADIUS:Send to unknown id 26 192.168.41.57:1813, Accounting-Request,
len 443
Jan 23 14:30:25.433:RADIUS: authenticator DA 1B 03 83 20 90 11 39 - F3 4F 70 F0 F5 8C CC
75
Jan 23 14:30:25.433:RADIUS: Acct-Session-Id     [44]  10  "000B5571"
Jan 23 14:30:25.433:RADIUS: Vendor, Cisco       [26]  56
Jan 23 14:30:25.433:RADIUS:  h323-setup-time    [25]  50 "h323-setup-time=14:30:25.429 GMT
 Wed Jan 23 2002"
Jan 23 14:30:25.433:RADIUS: Vendor, Cisco       [26]  26
Jan 23 14:30:25.433:RADIUS:  h323-gw-id         [33]  20  "h323-gw-id=OrigGW."
Jan 23 14:30:25.433:RADIUS: Vendor, Cisco       [26]  56
Jan 23 14:30:25.433:RADIUS:  Conf-Id            [24]  50  "h323-conf-id=931C146B 0F4411D6
AB5591F0 CBF3D765"
Jan 23 14:30:25.433:RADIUS: Vendor, Cisco       [26]  31
Jan 23 14:30:25.437:RADIUS:  h323-call-origin   [26]  25  "h323-call-origin=answer"
Jan 23 14:30:25.437:RADIUS: Vendor, Cisco       [26]  32
Jan 23 14:30:25.437:RADIUS:  h323-call-type     [27]  26  "h323-call-type=Telephony"
Jan 23 14:30:25.437:RADIUS: Vendor, Cisco       [26]  65
Jan 23 14:30:25.437:RADIUS:  Cisco AVpair       [1]   59 "h323-incoming-conf-id=931C146B
0F4411D6 AB5591F0 CBF3D765"
Jan 23 14:30:25.437:RADIUS: Vendor, Cisco       [26]  30
Jan 23 14:30:25.437:RADIUS:  Cisco AVpair       [1]   24  "subscriber=RegularLine"
Jan 23 14:30:25.437:RADIUS: User-Name           [1]   9   "1110001"
Jan 23 14:30:25.437:RADIUS: Acct-Status-Type    [40]  6   Start             [1]
Jan 23 14:30:25.437:RADIUS: Vendor, Cisco       [26]  19
Jan 23 14:30:25.437:RADIUS:  cisco-nas-port     [2]   13  "Serial6/0:0"
Jan 23 14:30:25.437:RADIUS: NAS-Port            [5]   6   0
Jan 23 14:30:25.437:RADIUS: Vendor, Cisco       [26]  29
Jan 23 14:30:25.437:RADIUS:  Cisco AVpair       [1]   23  "interface=Serial6/0:0"
Jan 23 14:30:25.437:RADIUS: Called-Station-Id   [30]  9   "2210001"
Jan 23 14:30:25.437:RADIUS: Calling-Station-Id  [31]  9   "1110001"
Jan 23 14:30:25.437:RADIUS: NAS-Port-Type       [61]  6   Async [0]
Jan 23 14:30:25.437:RADIUS: Service-Type        [6]   6   Login [1]
Jan 23 14:30:25.437:RADIUS: NAS-IP-Address      [4]   6   192.168.81.101
```

```
Jan 23 14:30:25.437:RADIUS: Event-Timestamp     [55]  6   1011796225
Jan 23 14:30:25.437:RADIUS: Delay-Time          [41]  6   0
Jan 23 14:30:25.441:RADIUS/ENCODE(00071EC0):Unsupported AAA attribute start_time
Jan 23 14:30:25.441:RADIUS/ENCODE(00071EC0):Unsupported AAA attribute timezone
Jan 23 14:30:25.441:RADIUS(00071EC0):sending
Jan 23 14:30:25.441:RADIUS:Send to unknown id 27 192.168.41.57:1813, Accounting-Request,
len 411
Jan 23 14:30:25.441:RADIUS: authenticator 15 83 23 D8 0B B2 3A C2 - 1D 8C EF B4 18 0F 1C
65
Jan 23 14:30:25.441:RADIUS: Acct-Session-Id     [44]  10  "000B5572"
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  56
Jan 23 14:30:25.441:RADIUS:  h323-setup-time    [25]  50 "h323-setup-time=14:30:25.441 GMT
 Wed Jan 23 2002"
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  26
Jan 23 14:30:25.441:RADIUS:  h323-gw-id         [33]  20  "h323-gw-id=OrigGW."
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  56
Jan 23 14:30:25.441:RADIUS:  Conf-Id            [24]  50  "h323-conf-id=931C146B 0F4411D6
AB5591F0 CBF3D765"
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  34
Jan 23 14:30:25.441:RADIUS:  h323-call-origin   [26]  28  "h323-call-origin=originate"
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  27
Jan 23 14:30:25.441:RADIUS:  h323-call-type     [27]  21  "h323-call-type=VoIP"
Jan 23 14:30:25.441:RADIUS: Vendor, Cisco       [26]  65
```

Configuration Examples for SIP AAA Features

SIP - Enhanced Billing Support for Gateways Examples

The following configuration example highlights the minimal configuration options that are necessary to carry out the full feature. After you configure the **aaa username** command described in this document, the gateway uses the information received in the SIP Authorization header and makes it available to AAA and Tcl IVR services. Typically, if you expect to use the full functionality of this feature, AAA and Tcl IVR have been configured previously.

```
Router# show running-config
Building configuration...
Current configuration : 4017 bytes
!
version 12.3
no service single-slot-reload-enable
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
hostname 3640-1
!
logging rate-limit console 10 except errors
! Need the following aaa line
aaa new-model
!
! Need the following four aaa lines
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
enable password lab
!
memory-size iomem 15
```

```
clock timezone GMT 0
voice-card 2
!
ip subnet-zero!
ip domain-name example.sip.com
ip name-server 172.18.192.154
ip name-server 10.10.1.5
!
no ip dhcp-client network-discovery
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
voice service voip
sip
rel1xx disable
!
fax interface-type fax-mail
mta receive maximum-recipients 0
call-history-mib retain-timer 500
!
controller E1 1/0
!
controller E1 1/1
!
controller T1 2/0
framing esf
linecode b8zs
pri-group timeslots 1-24
!
controller T1 2/1
framing sf
linecode ami
!
! Need the following three lines
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
interface Ethernet0/0
ip address 10.10.1.4 255.255.255.0
half-duplex
ip rsvp bandwidth 7500 7500
!
interface Ethernet0/1
no ip address
shutdown
half-duplex
!
interface Ethernet0/2
no ip address
shutdown
half-duplex
!
interface Ethernet0/3
no ip address
shutdown
half-duplex
!
interface FastEthernet1/0
ip address 172.18.192.197 255.255.255.0
duplex auto
speed auto
ip rsvp bandwidth 75000 75000
!
```

```
interface Serial2/0:23
no ip address
no logging event link-status
isdn switch-type primary-5ess
isdn incoming-voice modem
isdn T306 200000
isdn T310 200000
no cdp enable
!
ip classless
ip route 10.0.0.0 255.0.0.0 172.18.192.1
ip route 172.18.0.0 255.255.0.0 172.18.192.1
no ip http server
!
ip radius source-interface FastEthernet1/0
logging source-interface FastEthernet1/0
!
! Need the following radius-server lines for accounting/authentication
radius-server host 172.18.192.154 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
call rsvp-sync
!
! Need the following call application lines in order to enable
! tcl scripting feature.
call application voice voice_billing tftp://172.18.207.15/app_passport_silent.2.0.0.0.tcl
!
voice-port 2/0:23
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!
mgcp profile default
dial-peer cor custom
!
dial-peer voice 3640110 pots
destination-pattern 3640110
port 3/0/0
!
dial-peer voice 3640120 pots
destination-pattern 3640120
port 3/0/1
!
dial-peer voice 3660110 voip
destination-pattern 3660110
session protocol sipv2
session target ipv4:172.18.192.194
codec g711ulaw
!
dial-peer voice 3660120 voip
destination-pattern 3660120
session protocol sipv2
session target ipv4:172.18.192.194
codec g711ulaw
!
dial-peer voice 222 pots
huntstop
```

```
application session
destination-pattern 222
no digit-strip
direct-inward-dial
port 2/0:23
!
! Need to add the application line below to enable the tcl script
dial-peer voice 999 voip
application voice_billing
destination-pattern ...
session protocol sipv2
session target ipv4:10.10.1.2:5061
codec g711ulaw
!
! Need to add the aaa line below in order to enable proxy-authorization
! header processing
sip-ua
aaa username proxy-auth
!
line con 0
exec-timeout 0 0
length 0
line aux 0
line vty 0 4
!
!end
```

SIP Gateway HTTP Authentication Digest Examples

SIP: Gateway HTTP Authentication Digest Feature Disabled

```
Router# show running-config
Building configuration...
Current configuration :4903 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$Fyay$DfmV/uLXX.X94CoaRy569.
enable password lab
!
voice-card 3
!
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
```

```
ip cef
ip domain name example.sip.com
ip name-server 172.18.192.48
!
ip dhcp pool 1
host 172.18.193.173 255.255.255.0
client-identifier 0030.94c2.5d00
  option 150 ip 172.18.193.120
  default-router 172.18.193.120
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
voice class codec 2
 codec preference 1 g711ulaw
 codec preference 2 g729r8
 codec preference 5 g726r16
 codec preference 6 g726r24
!
fax interface-type fax-mail
!
translation-rule 100
!
interface FastEthernet0/0
 ip address 172.18.193.120 255.255.255.0
 ip mtu 900
 duplex auto
 speed auto
 no cdp enable
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 000000090200003094 26F6D0
snmp-server community public RO
```

```
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
voice-port 2/0/0
 station-id name 36602
 station-id number 36602
!
voice-port 2/0/1
!
mgcp
mgcp sdp simple
!
dial-peer cor custom
!
dial-peer voice 1 pots
 application session
 destination-pattern 36602
 port 2/0/0
!
dial-peer voice 5 voip
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.193.187
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 req-qos controlled-load
 acc-qos controlled-load
!
dial-peer voice 41 voip
 application session
 destination-pattern 333
 session protocol sipv2
 session target ipv4:10.102.17.80
 dtmf-relay rtp-nte
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
```

```
 incoming called-number 888
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
!
dial-peer voice 88 voip
 preference 1
 destination-pattern 888
 session protocol sipv2
 session target ipv4:172.18.193.187
!
dial-peer voice 123 voip
 destination-pattern 222
 session protocol sipv2
 session target ipv4:10.102.17.80
!
dial-peer voice 6 voip
 destination-pattern 36601
 session protocol sipv2
 session target ipv4:172.18.193.98
 session transport udp
 incoming called-number 36602
!
gateway
 timer receive-rtp 1200
!
sip-ua
 retry invite 1
 retry bye 2
 timers expires 60000
!
rtr responder
!
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password lab
 transport preferred all
 transport input all
 transport output all
!
end
```

SIP: Gateway HTTP Authentication Digest Feature Enabled

```
Router# show running-config
Building configuration...
Current configuration :5087 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$Fyay$DfmV/uLXX.X94CoaRy569.
enable password lab
!
voice-card 3
!
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
ip subnet-zero
ip tcp path-mtu-discovery
!
ip cef
ip domain name example.sip.com
ip name-server 172.18.192.48
!
ip dhcp pool 1
  host 172.18.193.173 255.255.255.0
  client-identifier 0030.94c2.5d00
  option 150 ip 172.18.193.120
  default-router 172.18.193.120
!
voice call carrier capacity active
!
voice service pots
!
voice service voip
 sip
  rel1xx disable
!
voice class codec 1
 codec preference 1 g729r8
 codec preference 2 g711ulaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
voice class codec 2
 codec preference 1 g711ulaw
 codec preference 2 g729r8
 codec preference 5 g726r16
 codec preference 6 g726r24
!
fax interface-type fax-mail
!
translation-rule 100
!
interface FastEthernet0/0
ip address 172.18.193.120 255.255.255.0
 ip mtu 900
 duplex auto
 speed auto
 no cdp enable
```

```
 ip rsvp bandwidth 75000 75000
!
interface FastEthernet0/1
 no ip address
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
!
ip radius source-interface FastEthernet0/0
logging source-interface FastEthernet0/0
dialer-list 1 protocol ip permit
snmp-server engineID local 000000090200003094 26F6D0
snmp-server community public RO
snmp-server community private RW
snmp-server packetsize 4096
snmp-server enable traps tty
!
tftp-server flash:XMLDefault.cnf.xml
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
voice-port 2/0/0
 station-id name 36602
 station-id number 36602
!
voice-port 2/0/1
!
mgcp
mgcp sdp simple
!
dial-peer cor custom
!
dial-peer voice 1 pots
 application session
 destination-pattern 36602
 port 2/0/0
 authentication username user1 password password1 realm example1.com ! authentication
xample 1
 authentication username user2 password password2 realm example2.com ! authentication
xample 2
!
dial-peer voice 5 voip
```

```
 application session
 destination-pattern 5550123
 session protocol sipv2
 session target ipv4:172.18.193.187
!
dial-peer voice 81 voip
 application session
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 req-qos controlled-load
 acc-qos controlled-load
!
dial-peer voice 41 voip
 application session
 destination-pattern 333
 session protocol sipv2
 session target ipv4:10.102.17.80
 dtmf-relay rtp-nte
!
dial-peer voice 7 voip
 application session
 destination-pattern 999
 session protocol sipv2
 session target ipv4:172.18.193.98
 incoming called-number 888
!
dial-peer voice 38 voip
 application session
 destination-pattern 3100802
 voice-class codec 1
 session protocol sipv2
 session target ipv4:172.18.193.99
!
dial-peer voice 88 voip
 preference 1
 destination-pattern 888
 session protocol sipv2
 session target ipv4:172.18.193.187
!
dial-peer voice 123 voip
 destination-pattern 222
 session protocol sipv2
 session target ipv4:10.102.17.80
!
dial-peer voice 6 voip
 destination-pattern 36601
 session protocol sipv2
 session target ipv4:172.18.193.98
 session transport udp
 incoming called-number 36602
!
gateway
 timer receive-rtp 1200
!
sip-ua
 authentication username user3 password password3 ! authentication example 3
 retry invite 1
 retry bye 2
 timers expires 60000
 registrar ipv4:172.18.193.187 expires 100 ! registrar example
!
rtr responder
!
```

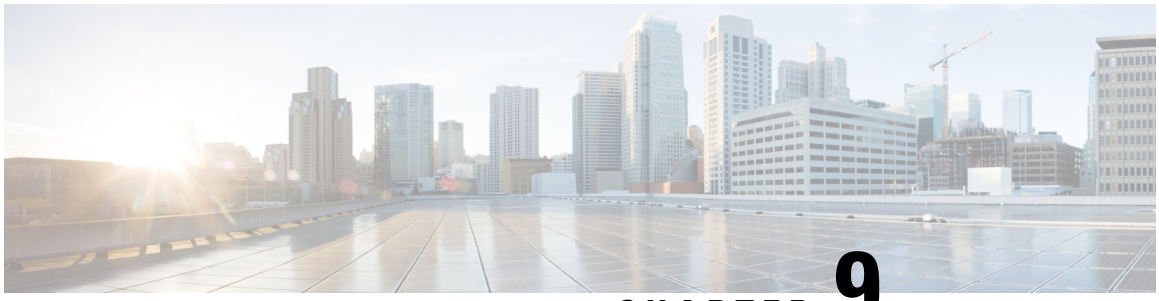
```
line con 0
 exec-timeout 0 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password lab
 transport preferred all
 transport input all
 transport output all
!
end
```

Additional References

General SIP References

References Mentioned in This Chapter (Listed Alphabetically)

- *Cisco IOS Security Command Reference*
- *Cisco IOS Security Configuration Guide,*
- *Cisco IOS SIP Configuration Guide*
- *Cisco IOS Tcl IVR and VoiceXML Application Guide*
- *Cisco Resource Policy Management System 2.0* at http://www.cisco.com/en/US/products/sw/netmgtsw/ps2074/tsd_products_support_eol_series_home.html
- *Cisco Tcl IVR API Programmer's Guide*
- *Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms*
- *Inter-Domain Gatekeeper Security Enhancement*, Cisco IOS Release 12.2(4)T
- *RADIUS Vendor-Specific Attributes Voice Implementation Guide* at http://noc.hsdn.org/files/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm



CHAPTER 9

SIP Binding

The SIP Binding feature enables you to configure a source IP address for signaling packets and media packets.

Feature Information for SIP Binding

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for SIP Binding

Feature Name	Releases	Feature Information
SIP Gateway Support for the bind Command	Cisco IOS 12.2(2)XB, 12.2(2)XB2, 12.2(8)T, 12.2(11)T, and 12.3(4)T Cisco IOS XE 3.1.0S	The SIP Gateway Support for the **bind** Command feature allows you to configure the source IP address of signaling packets and media packets. In 12.2(2)XB, this feature was introduced. In 12.3(4)T, this feature was expanded to provide the flexibility to specify different source interfaces for signaling and media, and allow network administrators a finer granularity of control on the network interfaces used for voice traffic. The following commands were introduced or modified: **bind**, **show dial-peer voice**, **show ip sockets**, **show sip-ua connections**, and **show sip-ua status**.

Feature Name	Releases	Feature Information
Support for Ability to Configure Source IP Address for Signaling and Media per SIP Trunk	15.1(2)T	This feature allows you to configure a separate source IP address per SIP trunk. This source IP address is embedded in all SIP signaling and media packets that traverse the SIP trunk. This feature enables service providers for better profiling and billing policies. It also enables greater security for enterprises by the use of distinct IP addresses within and outside the enterprise domain. The following command was introduced or modified: **voice-class sip bind**.

Information About SIP Binding

When you configure SIP on a router, the ports on all its interfaces are open by default. This makes the router vulnerable to malicious attackers who can execute toll fraud across the gateway if the router has a public IP address and a public switched telephone network (PSTN) connection. To eliminate the threat, you should bind an interface to an IP address so that only those ports are open to the outside world. In addition, you should protect any public or untrusted interface by configuring a firewall or an access control list (ACL) to prevent unwanted traffic from traversing the router.

Benefits of SIP Binding

- SIP signaling and media paths can advertise the same source IP address on the gateway for certain applications, even if the paths used different addresses to reach the source. This eliminates confusion for firewall applications that may have taken action on source address packets before the use of binding.
- Firewalls filter messages based on variables such as the message source, the target address, and available ports. Normally a firewall opens only certain addresses or port combination to the outside world and those addresses can change dynamically. Because VoIP technology requires the use of more than one address or port combination, the **bind** command adds flexibility by assigning a gateway to a specific interface (and therefore the associated address) for the signaling or media application.
- You can obtain a predefined and separate interface for both signaling and media traffic. After a **bind** command is in effect, the interface it limits is bound solely to that purpose. Administrators can therefore dictate the use of one network to transport the signaling and another network to transport the media. The benefits of administrator control are:
  - Administrators know the traffic that runs on specific networks, thereby making debugging easier.
  - Administrators know the capacity of the network and the target traffic, thereby making engineering and planning easier.
  - Traffic is controlled, allowing Qualtiy of Service (QoS) to be monitored.
- The **bind media** command relaxes the constraints imposed by the **bind control** and **bind all** commands, which cannot be set during an active call. The **bind media** command works with active calls.

Source Address

In early releases of Cisco IOS software with SIP functionality, the source address of a packet going out of the gateway was never deterministic. That is, the session protocols and VoIP layers always depended on the IP layer to give the *best local address* . The best local address was then used as the source address (the address showing where the SIP request came from) for signaling and media packets. Using this non-deterministic address occasionally caused confusion for firewall applications, because a firewall could not be configured with an exact address and would take action on several different source address packets.

However, the **bind** command enables you to configure the source IP address of signaling and media packets to a specific interface's IP address. Thus, the address that goes out on the packet is bound to the IP address of the interface specified with the **bind** command. Packets that are not destined to the bound address are discarded.

When you do not want to specify a bind address or if the interface is down, the IP layer still provides the best local address.

The Support Ability to Configure Source IP Address for Signaling and Media per SIP Trunk feature extends the global bind functionality to support the SIP signaling Transport Layer Socket (TLS) with UDP and TCP. The source address at the dial peer is the source address in all the signaling and media packets between the gateway and the remote SIP entity for calls using the dial-peer. Multiple SIP listen sockets with specific source address handle the incoming SIP traffic from each selected SIP entity. The order of preference for retrieving the SIP signalling and media source address for inbound and outbound calls is as follows:

- Bind configuration at dial peer level
- Bind configuration at global level
- Best local IP address to reach the destination

The table below describes the state of the system when the **bind** command is applied in the global or dial peer level:

Table 39: State of the System for the bind Address

Bind State	System Status
No global bind	The best local address is used in all outbound SIP messages. Only one SIP listen socket with a wildcard source address.
Global bind	Global bind address used in all outbound SIP messages. Only one SIP listen socket with global bind address.
No global bind Dial peer bind	Dial peer bind address is used in outbound SIP messages of this dial peer. The remaining SIP messages use the best local address. One SIP listen socket with a wildcard source address. Additional SIP listen socket for each different dial peer bind listening on the specific dial peer bind address.

Bind State	System Status
Global bind Dial peer bind	Dial peer bind address is used in outbound SIP messages of this dial peer. The remaining SIP messages use the global bind address. One SIP listen socket with global bind address. Additional SIP listen socket for each different dial peer bind command listening on the specific dial peer bind address.

The **bind** command performs different functions based on the state of the interface (see the table below).

Table 40: State of the Interface for the bind Command

Interface State	Result Using Bind Command
Shut down With or without active calls	TCP, TLS, and User Datagram Protocol (UDP) socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened to listen to any IP address. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
No shut down No active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened and bound to the IP address set by the **bind** command. The sockets accept packets destined for the bound address only. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.
No shut down Active calls	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any IP address. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.

Interface State	Result Using Bind Command
Bound-interface IP address is removed.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any address, because the IP address has been removed. This happens even when SIP was never bound to an IP address. A message stating that the IP address has been deleted from the SIP bound interface is printed. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
The physical cable is pulled on the bound port or the interface layer is down.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened and bound to listen to any address. When the pulled cable is replaced, the result is as documented for **no shutdown** interfaces. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
A bind interface is shut down or its IP address is changed or the physical cable is pulled while SIP calls are active.	The call becomes a one-way call with media flowing in only one direction. It flows from the gateway where the change or shutdown took place, to the gateway where no change occurred. Thus, the gateway with the status change no longer receives media. The call is then disconnected, but the disconnected message is not understood by the gateway with the status change, and the call is still assumed to be active. If the bind interface is shutdown, the dial peer bind socket listeners of the interface are closed. If the IP address of the interface is changed, the socket listeners representing the bind command is opened with the available IP address of the interface and the configuration turns active for all subsequent SIP messages.
Note If there are active calls, the **bind** command does not take effect if it is issued for the first time or if another **bind** command is in effect. A message reminds you that there are active calls and that the change cannot take effect.	

The **bind** command applied at the dial peer level can be modified only in the following situations:

- Dial peer bind is disabled in the supported IOS configuration options.
- Dial peer bind is removed when the bound interface is removed.
- Dial peer bind is removed when the dial peer is removed.

Voice Media Stream Processing

The SIP Gateway Support Enhancements to the bind Command feature extends the capabilities of the **bind** command by supporting a deterministic network interface for the voice media stream. Before the voice media stream addition, the **bind** command supported a deterministic network interface for control (signaling) traffic or all traffic. With the SIP Gateway Support Enhancements to the bind Command feature, a finer granularity of control is achieved on the network interfaces used for voice traffic.

If multiple **bind** commands are issued in sequence—that is, if one **bind** command is configured and then another **bind** command is configured—a set interaction happens between the commands. The table below describes the expected command behavior.

Table 41: Interaction Between Previously Set and New bind Commands

Interface State	bind Command	Result Using bind Command
Without active calls	**bind all**	Generated **bind control** and **bind media** commands to override existing **bind control** and **bind media** commands.
	bind control	Overrides existing **bind control** command.
	bind media	Overrides existing **bind media** command.
With active calls	**bind all** or **bind control**	Blocks the command, and the following messages are displayed: • 00:16:39: There are active calls • 00:16:39: configure_sip_bind_command: The bind command change will not take effect
	bind media	Succeeds and overrides any existing **bind media** command.

The **bind all** and **bind control** commands perform different functions based on the state of the interface.

Note The **bind all** command only applies to global level, whereas the **bind control** and **bind media** command apply to global and dial peer. The table below applies to **bind media** only if the media interface is the same as the **bind control** interface. If the two interfaces are different, media behavior is independent of the interface state.

Table 42: bind all and bind control Functions, Based on Interface State

Interface State	Result Using bind all or bind control Commands
Shut down With or without active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened to listen to any IP address. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
Not shut down Without active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened and bound to the IP address set by the **bind** command. The sockets accept packets destined for the bound address only. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.
Not shut down With active calls	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any IP address. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.
Bound interface's IP address is removed.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any address because the IP address has been removed. A message is printed that states the IP address has been deleted from the bound SIP interface. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
The physical cable is pulled on the bound port, or the interface layer goes down.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened and bound to listen to any address. When the pulled cable is replaced, the result is as documented for interfaces that are not shut down. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.

Interface State	Result Using bind all or bind control Commands
A bind interface is shut down, or its IP address is changed, or the physical cable is pulled while SIP calls are active.	The call becomes a one-way call with media flowing in only one direction. The media flows from the gateway where the change or shutdown took place to the gateway where no change occurred. Thus, the gateway with the status change no longer receives media. The call is then disconnected, but the disconnected message is not understood by the gateway with the status change, and the call is still assumed to be active. If the bind interface is shutdown, the dial peer bind socket listeners of the interface are closed. If the IP address of the interface is changed, the socket listeners representing the bind command is opened with the available IP address of the interface and the configuration turns active for all subsequent SIP messages.

Configuring SIP Binding

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-addressmask* [**secondary**]
5. **exit**
6. Use one of the following commands to configure SIP binding:
   - **bind** {**control** | **all**} **source-interface** *interface-id* [**ipv6-address** *ipv6-address*] in SIP configuration mode.
   - **bind media** {**source-address ipv4** *ipv4-address* | **source-interface** *interface-id* [**ipv6-address** *ipv6-address*]} in SIP configuration mode.
   - **voice-class sip bind control source interface** *interface-id* [**ipv6-address** *ipv6-address*] in dial-peer configuration mode.
   - **voice-class sip bind media** {**source-address ipv4** *ipv4-address* | **source-interface** *interface-id* [**ipv6-address** *ipv6-address*]} in dial-peer configuration mode.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet0/0`	Configures an interface type and enters the interface configuration mode. • *type number* —Type of interface to be configured and the port, connector, or interface card number.
Step 4	**ip address** *ip-addressmask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.200.33 255.255.255.0`	Configures a primary or secondary IP address for an interface. **Note** Secondary IP address on an interface with SIP binding is not supported for CUBE.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the current mode.
Step 6	Use one of the following commands to configure SIP binding: • **bind** {**control** | **all**} **source-interface** *interface-id* [**ipv6-address** *ipv6-address*] in SIP configuration mode. • **bind media** {**source-address ipv4** *ipv4-address* | **source-interface** *interface-id* [**ipv6-address** *ipv6-address*]} in SIP configuration mode. • **voice-class sip bind control source interface** *interface-id* [**ipv6-address** *ipv6-address*] in dial-peer configuration mode. • **voice-class sip bind media** {**source-address ipv4** *ipv4-address* | **source-interface** *interface-id* [**ipv6-address** *ipv6-address*]} in dial-peer configuration mode. **Example:** SIP binding in SIP configuration mode: `Device(config)# voice service voip` `Device(conf-voi-serv)# sip` `Device(conf-serv-sip)# bind control source-interface FastEthernet0/0` `Device(conf-serv-sip)# exit` `Device(config)# voice service voip` `Device(conf-voi-serv)# sip` `Device(conf-serv-sip)# bind media source-address ipv4 172.18.192.204` `Device(conf-serv-sip)# exit` **Example:**	Sets a source interface for signaling and media packets. The binding applies to the specified interfaces only. SIP must be configured globally or at a dial peer level. • **control** —Binds signaling packets. • **media** —Binds media packets. • **all** —Binds signaling and media packets. • **source-address**—Binds media packets directly to an IP address. • **ipv4** *ipv4-address*—Configures the IPv4 address. • **source interface** *interface-id* —Type of interface and its ID. • **ipv6-address** *ipv6-address* —Configures the IPv6 address. Ensure that the IPv6 address has been applied to an interface.

	Command or Action	Purpose
	SIP binding in dial-peer configuration mode: `Device(config)# dial-peer voice 100 voip` `Device(config-dial-peer)# session protocol sipv2` `Device(config-dial-peer)# voice-class sip bind control source-interface fastethernet0/0` `Device(config-dial-peer)# exit` `Device(config)# dial-peer voice 100 voip` `Device(config-dial-peer)# session protocol sipv2` `Device(config-dial-peer)# voice-class sip bind media source-address ipv4 172.18.192.204` `Device(config-dial-peer)# exit`	
Step 7	**end**	Exits to privileged EXEC mode.

Verifying SIP Binding

SUMMARY STEPS

1. **show ip sockets**
2. **show sip-ua status**
3. **show sip-ua connections** {**tcp** [**tls**] | **udp**} {**brief** | **detail**}
4. **show dial-peer voice**
5. **show running-config**

DETAILED STEPS

Step 1 **show ip sockets**

Use this command to display IP socket information and indicate whether the bind address of the receiving gateway is set.

The following sample output indicates that the bind address of the receiving gateway is set:

Example:

```
Device# show ip sockets

Proto Remote Port Local Port In Out Stat TTY OutputIF
17 0.0.0.0 0--any-- 2517 0 0 9 0
17 --listen-- 172.18.192.204 1698 0 0 1 0
17 0.0.0.0 0 172.18.192.204 67 0 0 489 0
17 0.0.0.0 0 172.18.192.204 5060 0 0 A1 0
```

Example:

Step 2 **show sip-ua status**

Use this command to display SIP user-agent status and indicate whether bind is enabled.

The following sample output indicates that signaling is disabled and media on 172.18.192.204 is enabled:

Example:

```
Device# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent for TLS over TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): ENABLED 172.18.192.204
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 70
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: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
Reason Header will override Response/Request Codes: DISABLED
Out-of-dialog Refer: DISABLED
Presence support is DISABLED
protocol mode is ipv4
SDP application configuration:
 Version line (v=) required
Owner line (o=) required
 Timespec line (t=) required
 Media supported: audio video image
 Network types supported: IN
 Address types supported: IP4 IP6
 Transport types supported: RTP/AVP udptl
```

Step 3 **show sip-ua connections** {**tcp** [**tls**] | **udp**} {**brief** | **detail**}

Use this command to display the connection details for the UDP transport protocol. The command output looks identical for TCP and TLS.

Example:

```
Device# show sip-ua connections udp detail

Total active connections      : 0
No. of send failures          : 0
No. of remote closures        : 0
No. of conn. failures         : 0
No. of inactive conn. ageouts : 10
---------Printing Detailed Connection Report---------
Note:
 ** Tuples with no matching socket entry
    - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port>'
      to overcome this error condition
 ++ Tuples with mismatched address/port entry
    - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port> id <connid>'
      to overcome this error condition
 No Active Connections Found
-------------- SIP Transport Layer Listen Sockets ---------------
  Conn-Id               Local-Address
 ===========    ============================
   2            [9.42.28.29]:5060
```

Step 4 **show dial-peer voice**

Use this command, for each dial peer configured, to verify that the dial-peer configuration is correct. The following is sample output from this command for a VoIP dial peer:

Example:

```
Device# show dial-peer voice 101

VoiceOverIpPeer1234
        peer type = voice, system default peer = FALSE, information type = voice,
        description = `',
        tag = 1234, destination-pattern = `',
        voice reg type = 0, corresponding tag = 0,
        allow watch = FALSE
        answer-address = `', preference=0,
        CLID Restriction = None
        CLID Network Number = `'
        CLID Second Number sent
        CLID Override RDNIS = disabled,
        rtp-ssrc mux = system
        source carrier-id = `', target carrier-id = `',
        source trunk-group-label = `',  target trunk-group-label = `',
        numbering Type = `unknown'
        group = 1234, Admin state is up, Operation state is down,
        incoming called-number = `', connections/maximum = 0/unlimited,
        DTMF Relay = disabled,
        modem transport = system,
        URI classes:
            Incoming (Request) =
            Incoming (Via) =
            Incoming (To) =
            Incoming (From) =
            Destination =
        huntstop = disabled,
        in bound application associated: 'DEFAULT'
        out bound application associated: ''
        dnis-map =
        permission :both
        incoming COR list:maximum capability
        outgoing COR list:minimum requirement
        outgoing LPCOR:
        Translation profile (Incoming):
        Translation profile (Outgoing):
        incoming call blocking:
        translation-profile = `'
        disconnect-cause = `no-service'
        advertise 0x40 capacity_update_timer 25 addrFamily 4 oldAddrFamily 4
        mailbox selection policy: none
        type = voip, session-target = `',
        technology prefix:
        settle-call = disabled
        ip media DSCP = ef, ip media rsvp-pass DSCP = ef
        ip media rsvp-fail DSCP = ef, ip signaling DSCP = af31,
        ip video rsvp-none DSCP = af41,ip video rsvp-pass DSCP = af41
        ip video rsvp-fail DSCP = af41,
        ip defending Priority = 0, ip preemption priority = 0
        ip policy locator voice:
        ip policy locator video:
        UDP checksum = disabled,
        session-protocol = sipv2, session-transport = system,
        req-qos = best-effort, acc-qos = best-effort,
        req-qos video = best-effort, acc-qos video = best-effort,
        req-qos audio def bandwidth = 64, req-qos audio max bandwidth = 0,
        req-qos video def bandwidth = 384, req-qos video max bandwidth = 0,
```

```
RTP dynamic payload type values: NTE = 101
Cisco: NSE=100, fax=96, fax-ack=97, dtmf=121, fax-relay=122
       CAS=123, TTY=119, ClearChan=125, PCM switch over u-law=0,
       A-law=8, GSMAMR-NB=117 iLBC=116, AAC-ld=114, iSAC=124
       lmr_tone=0, nte_tone=0
       h263+=118, h264=119
       G726r16 using static payload
       G726r24 using static payload
RTP comfort noise payload type = 19
fax rate = voice,   payload size =  20 bytes
fax protocol = system
fax-relay ecm enable
Fax Relay ans enabled
Fax Relay SG3-to-G3 Enabled (by system configuration)
fax NSF = 0xAD0051 (default)
codec = g729r8,   payload size =  20 bytes,
video codec = None
voice class codec = `'
voice class sip session refresh system
voice class sip rsvp-fail-policy voice post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy voice post-alert optional keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert optional keep-alive interval 30
text relay = disabled
Media Setting = forking (disabled) flow-through (global)
Expect factor = 10, Icpif = 20,
Playout Mode is set to adaptive,
Initial 60 ms, Max 1000 ms
Playout-delay Minimum mode is set to default, value 40 ms
Fax nominal 300 ms
Max Redirects = 1, signaling-type = cas,
VAD = enabled, Poor QOV Trap = disabled,
Source Interface = NONE
voice class sip url = system,
voice class sip tel-config url = system,
voice class sip rel1xx = system,
voice class sip anat = system,
voice class sip outbound-proxy = "system",
voice class sip associate registered-number =
                system,
voice class sip asserted-id system,
voice class sip privacy system
voice class sip e911 = system,
voice class sip history-info = system,
voice class sip reset timer expires 183 = system,
voice class sip pass-thru headers = system,
voice class sip pass-thru content unsupp = system,
voice class sip pass-thru content sdp = system,
voice class sip copy-list = system,
voice class sip g729 annexb-all = system,
voice class sip early-offer forced = system,
voice class sip negotiate cisco = system,
voice class sip block 180 = system,
voice class sip block 183 = system,
voice class sip block 181 = system,
voice class sip preloaded-route = system,
voice class sip random-contact = system,
voice class sip random-request-uri validate = system,
voice class sip call-route p-called-party-id = system,
voice class sip call-route history-info = system,
voice class sip privacy-policy send-always = system,
voice class sip privacy-policy passthru = system,
voice class sip privacy-policy strip history-info = system,
voice class sip privacy-policy strip diversion = system,
```

```
voice class sip map resp-code 181 = system,
voice class sip bind control = enabled, 9.42.28.29,
voice class sip bind media = enabled, 9.42.28.29,
voice class sip bandwidth audio = system,
voice class sip bandwidth video = system,
voice class sip encap clear-channel = system,
voice class sip error-code-override options-keepalive failure = system,
voice class sip calltype-video = false
voice class sip registration passthrough = System
voice class sip authenticate redirecting-number  = system,
redirect ip2ip = disabled
local peer = false
probe disabled,
Secure RTP: system (use the global setting)
voice class perm tag = `'
Time elapsed since last clearing of voice call statistics never
Connect Time = 0, Charged Units = 0,
Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
Accepted Calls = 0, Refused Calls = 0,
Last Disconnect Cause is "",
Last Disconnect Text is "",
Last Setup Time = 0.
Last Disconnect Time = 0.
```

Note If the bind address is not configured at the dial-peer, the output of the **show dial-peer voice** command remains the same except for the values of the **voice class sip bind control** and **voice class sip bind media**, which display "system," indicating that the bind is configured at the global level.

Step 5 **show running-config**

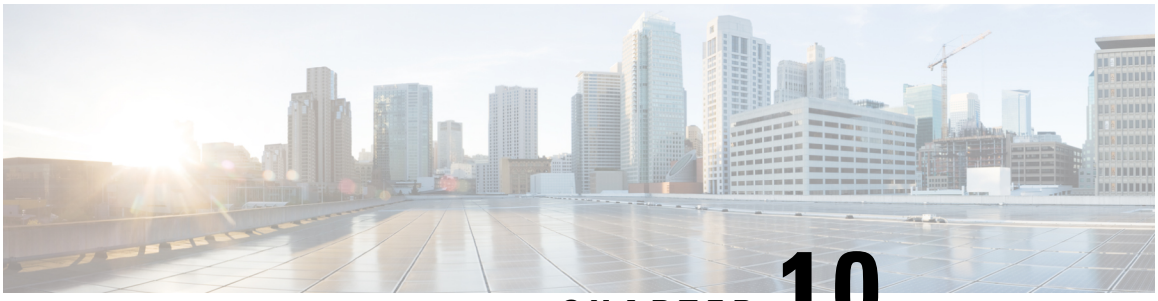
Although the bind all command is an accepted configuration, it does not appear in **show running-config** command output. Because the **bind all** command is equivalent to issuing the commands **bind control** and **bind media**, those are the commands that appear in the **show running-config** command output.

Example:

The following sample output shows that bind is enabled on router 172.18.192.204:

```
Building configuration...
Current configuration : 2791 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
ip subnet-zero
ip ftp source-interface Ethernet0
!
voice service voip
 sip
  bind control source-interface FastEthernet0
!
interface FastEthernet0
 ip address 172.18.192.204 255.255.255.0
 duplex auto
 speed auto
 fair-queue 64 256 1000
 ip rsvp bandwidth 75000 100
```

```
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 5550111
port 1/1/1
!
dial-peer voice 29 voip
application session
destination-pattern 5550133
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
line con 0
line aux 0
line vty 0 4
login
!
end
```



CHAPTER 10

Configuring SIP Connection-Oriented Media Forking and MLPP Features

This chapter describes how to configure the following media-support features for SIP:

- SIP: Connection-Oriented Media (Comedia) Enhancements for SIP
- SIP: Multilevel Precedence and Priority Support
- SIP Support for Media Forking

Feature History for SIP: Connection-Oriented Media (Comedia) Enhancements for SIP

Release	Modification
12.2(13)T	The feature was introduced.

History for the SIP: Multilevel Precedence and Priority Support Feature

Release	Modification
12.4(2)T	This feature was introduced.

Feature History for SIP Support for Media Forking

Release	Modification
12.2(15)T	The feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Connection-Oriented Media Forking and MLPP

SIP: Connection-Oriented Media Enhancements for SIP Feature

- Configure NAT. For information about configuring NAT, see "Configuring Network Address Translation: Getting Started."

SIP Support for Media Forking Feature

- Configure the gateway **receive-rtcp timer** (using the **timer receive-rtcp** command on the gateway) if a SIP media activity timer is desired. The timer monitors and disconnects calls if no Real-Time Control Protocol (RTCP) packets are received within a configurable time period.
- If using a Cisco 2600 series, Cisco 3600 series, or Cisco 37xx, ensure that the voice card is configured for high-complexity mode of operation for full media-forking functionality.
- If using a Cisco 7200 series, ensure that the voice card is configured for high-complexity mode of operation.
- If using a Cisco AS5300, ensure that the proper version of VCWare is loaded onto your router. For media forking, the voice feature card is the DSPM Voice (C542 or C549).

Restrictions for SIP Connection-Oriented Media Forking and MLPP

SIP: Connection-Oriented Media Enhancements for SIP Feature

- The feature does not support the a=direction:both attribute of the Session Description Protocol (SDP) message, as defined in the Internet Engineering Task Force (IETF) draft, http://tools.ietf.org/html/draft-ietf-mmusic-sdp-comedia-00
- There is likewise no corresponding command-line-interface (CLI) command. If the SIP gateway receives an SDP message specifying a=direction:both, the endpoint is treated by the gateway as active and is considered to be inside the NAT.

Note Proxy parallel forking is not supported with this feature unless all endpoints reply with 180 message response without SDP, because this feature does not handle media coming from multiple endpoints simultaneously.

SIP Support for Media Forking Feature

- The following capabilities are not supported:
  - The capability to use IP multicast.
  - The capability to create streams with different codecs.
  - The capability to use media forking functionality over Transmission Control Protocol (TCP). User Datagram Protocol (UDP) only is supported.
  - The capability to make fax calls when multiple streams are used.
  - The capability to make modem calls when multiple streams ar e used.
  - IP-to-IP hairpinning, because there is no telephony call leg to be associated with a call.
  - When using no voice activity detection (VAD), you can use 10 percent of the capacity of the router to make media forked calls. If no VAD is configured on the Cisco 7200 series, a maximum of 15 channels can be used. For example, on a Cisco 2691, two T1s are supported. Ten percent of two T1s is 4.8 calls, so 4.8 media forking calls can be performed when no VAD is configured. For a Cisco AS5300, four T1s are supported that give a total of 96 calls. Ten percent of 96 is 9.6 calls, so 9.6 media forking calls can be performed when no VAD is configured.
- The following restrictions apply to codec usage:
  - The codecs implemented are G.711, G.729, and G.726 on all supported platforms. No other codecs are supported.
  - For dynamic payload type codecs (G.726), the payload type must be the same for all streams in the call. This ensures that the codec is mapped to the same payload type on all streams of a re-Invite message.
  - The codecs on all the streams must be the same and must be one of the supported codecs . If a re-Invite message is received and multiple codecs are listed in the m-lines of the forked-media streams, the gateway attempts to find the codec in the list that matches the first stream. If a matching codec is not found, the stream is rejected by setting the port number to 0 in the response.

Note For information on codecs, see "Map Payload Types to Dynamic Payload Codecs".

- The following restrictions apply to forking functionality and voice feature cards:
  - With the Cisco 2600 series routers, Cisco 3600 series routers, or Cisco 37xx routers, forking is partially supported for NM-HDV configured for medium- complexity mode of operation. A maximum of two streams is supported, and the only combinations supported are the following: voice-only on both streams and voice plus dual-tone multifrequency (DTMF)-relay on both streams. For full functionality, configure the NM-HDV for high-complexity mode of operation.
  - With the Cisco 7200 series routers, the Enhanced High-Capacity Digital Voice port adapter (PA-VXC-2TE1+ T1/E1) must be configured for high-complexity mode of operation.
  - With the Cisco AS5300 universal access server, DSPM Voice (C542 and C549) must be configured.
- The following restrictions apply to DTMF relay:

- DTMF-relay is supported as described in RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals . Cisco proprietary DTMF is not supported.
- When DTMF-relay is configured, only the Real-Time Transport Protocol Named Telephony Event (RTP-NTE) payload format can be used in a forked call. RTP-NTE is described in RFC 2833 and prevents the generation of spurious digits at the receiving gateway.
- Streams that include only DTMF-relay packets can be used only in a two-stream call and must be established as the second stream.
- All forked voice-only and voice plus DTMF-relay streams must use the same codec.

- The following restrictions apply to media streams:
  - Multiple Domain Name System (DNS) media queries are not supported on all media streams. DNS query is done on the fully qualified domain name (FDQN) of the initial stream only.
  - Media streams are created and deleted only through re-Invite messages. They are not created through the CLI.
  - A maximum of three VoIP media streams are supported because three streams can be concurrently sent to the digital signal processor (DSP).
  - Calls that have a single stream (that is, a conventional two-party call) cannot be set up as DTMF only.
  - The first stream cannot be deleted while other streams are active, nor can it be put on hold.
  - The stream type of an active stream cannot be modified. For example, you cannot change a voice-only stream to voice plus DTMF only.

Information About SIP Connection-Oriented Media Forking and MLPP

To configure connection-oriented media and forking features for SIP, you should understand the following concepts:

SIP Connection-Oriented Media Enhancements for SIP

The SIP: Connection-Oriented Media (Comedia) Enhancements for SIP feature allows a Cisco gateway to check the media source of incoming Realtime Transport Protocol (RTP) packets, and allows the endpoint to advertise its presence inside or outside of Network Address Translation (NAT). Using the feature enables symmetric NAT traversal by supporting the capability to modify and update an existing RTP session remote address and port.

Feature benefits include the following:

- Ability to check the media source address and port of incoming RTP packets, thereby enabling the remote address and port of the existing session to be updated
- Enhanced interoperability in networks where NAT devices are unaware of SIP or SDP signaling
- Ability to advertise endpoint presence inside or outside NAT
- Ability to specify the connection role of the endpoint

Symmetric NAT Traversal

The Connection-Oriented Media (Comedia) Enhancements for SIP feature provides the following feature to symmetric NAT traversal:

- Allows the Cisco gateway to check the media source of incoming (RTP) packets.
- Allows the endpoint to advertise its presence inside or outside of NAT.

NAT, which maps the source IP address of a packet from one IP address to a different IP address, has varying functionality and configurations. NAT can help conserve IP version 4 (IPv4) addresses, or it can be used for security purposes to hide the IP address and LAN structure behind the NAT. VoIP endpoints may both be outside NAT, both inside, or one inside and the other outside.

In symmetric NAT, all requests from an internal IP address and port to a specific destination IP address and port are mapped to the same external IP address and port. The feature provides additional capabilities for symmetric NAT traversal.

Prior to the implementation of connection-oriented media enhancements, NAT traversal presented challenges for both SIP, which signals the protocol messages that set up a call, and for RTP, the media stream that transports the audio portion of a VoIP call. An endpoint with connections to clients behind NATs and on the open Internet had no way of knowing when to trust the addressing information it received in the SDP portion of SIP messages, or whether to wait until it received a packet directly from the client before opening a channel back to the source IP:port of that packet. Once a VoIP session was established, the endpoint was, in some scenarios, sending packets to an unreachable address. This scenario typically occurred in NAT networks that were SIP-unaware.

In addition to the challenges posed by NAT traversal in SIP, NAT traversal in RTP requires that a client must know what type of NAT it sits behind, and that it must also obtain the public address for an RTP stream. Any RTP connection between endpoints outside and inside NAT must be established as a point-to-point connection. The external endpoint must wait until it receives a packet from the client so that it knows where to reply. The connection-oriented protocol used to describe this type of session is known as Connection-Oriented Media (Comedia), as defined in the IETF draft, draft-ietf-mmusic-sdp-comedia-04.txt, *Connection-Oriented Media Transport in SDP* .

Cisco IOS VoiceXML features implement one of many possible SIP solutions to address problems with different NAT types and traversals. With Cisco IOS VoiceXM, the gateway can open an RTP session with the remote end and then update or modify the existing RTP session remote address and port (raddr:rport) with the source address and port of the actual media packet received after passing through NAT. The feature allows you to configure the gateway to modify the RTP session remote address and port by implementing support for the SDP direction (a=direction:<role>) attribute defined in, draft-ietf-mmusic-sdp-comedia-04.txt, *Connection-Oriented Media Transport in SDP* . Valid values for the attribute are as follows:

- active--the endpoint initiates a connection to the port number on the m= line of the session description from the other endpoint.
- passive--the endpoint accepts a connection to the port number on the m= line of the session description sent from itself to the other endpoint.
- both--the endpoint both accepts an incoming connection and initiates an outgoing connection to the port number on the m= line of the session description from the other endpoint.

The feature introduces CLI commands to configure the following SIP user-agent settings for symmetric NAT:

- The **nat symmetric check-media-src** command enables checking the incoming packet for media source address. This capability allows the gateway to check the source address and update the media session with the remote media address and port.
- The **nat symmetric role** command specifies the function of the endpoint in the connection setup procedure. Set the **role** keyword to one of the following:
  - **active**--The endpoint initiates a connection to the port number on the m= line of the session description from the other endpoint.
  - **passive**--The endpoint accepts a connection to the port number on the m= line of the session description sent from itself to the other endpoint.

Note The Cisco Comedia implementation does not support a=direction:both. If the Cisco gateway receives a=direction:both in the SDP message, the endpoint is considered active.

Sample SDP Message

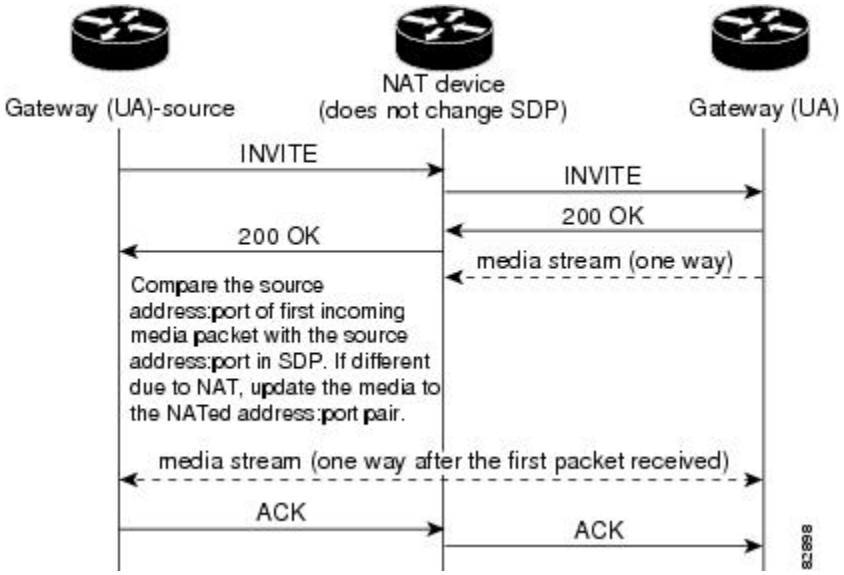
The following example shows a sample SDP message that describes a session with the direction:<role> attribute set to passive:

```
v=o
o=CiscoSystemsSIP-GW-UserAgent 5732 7442 IN IP4 10.15.66.43
s=SIP Call
c=IN IP4 10.15.66.43
t=0 0
m=audio 17306 RTP/AVP 0 100
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
a=direction:passive
```

Symmetric NAT Call Flows

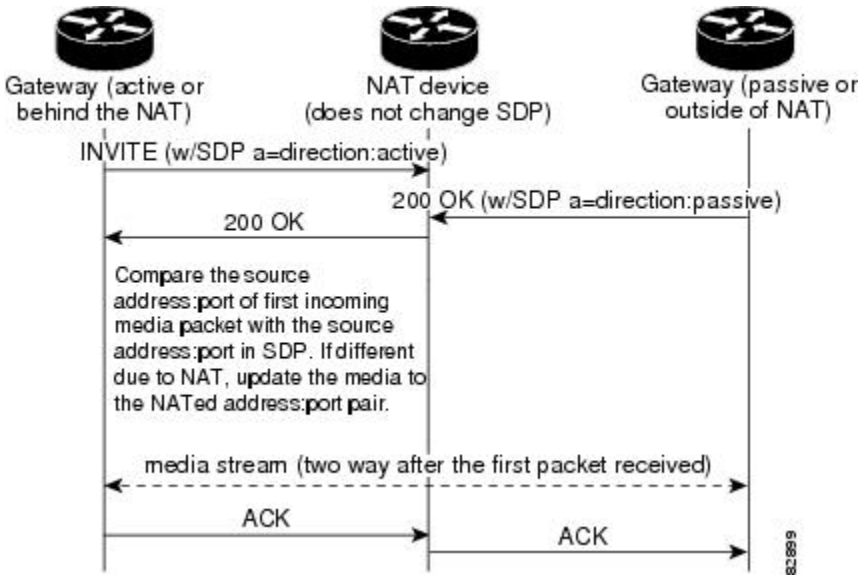
The following call flow diagrams describe call setup during symmetric NAT traversal scenarios. The figure below shows a NAT device that is unaware of SIP or SDP signaling. The SIP endpoints are not communicating the connection-oriented media direction role in the SDP message. The originating gateway is configured, using the **nat symmetric check-media-src** command to detect the media source and update the VoIP RTP session to the network address translated address:port pair.

Figure 66: SIP Endpoints Not Communicating the SDP direction:<role> Attribute



The figure below shows a NAT device that is unaware of SIP or SDP signaling, but the SIP endpoints are communicating the connection-oriented media direction role in the SDP message. The originating gateway is configured as a passive entity in the network using the **nat symmetric role** command. When the passive entity receives a direction role of active, it updates the VoIP RTP session to the network address translated address:port pair.

Figure 67: SIP Endpoints Communicating the SDP direction:<role>



Note Configuring the originating gateway for passive or active setting can differ from the NAT device setup. The SIP user agent communicates the CLI configured direction role in the SDP body. Checking for media packets is automatically enabled only if the gateway receives a direction role of active or both.

When renegotiating media mid call, such as when the call is moving from standard to T38 fax relay, the UDP ports used are often renegotiated on third-party endpoints. This new port is not recognized by the symmetric NAT feature.

SIP Multilevel Precedence and Priority Support

The SIP: Multilevel Precedence and Priority Support feature enables Cisco IOS gateways to interoperate with other multilevel-precedence and preemption (MLPP)-capable circuit-switched networks.

An MLPP-enabled call has an associated priority level that applications that handle emergencies and congestions use to determine which lower-priority call to preempt in order to dedicate their end-system resources to high-priority communications. This feature addresses the aspect of preemption when interworking with defense-switched networks (DSNs) that are connected through the Cisco IOS gateway.

Description of the SIP

The SIP: Multilevel Precedence and Priority Support feature enables Cisco IOS gateways to interoperate with other MLPP-capable circuit-switched networks. The U.S. Department of Defense (DoD) and Defense Information Service Agency (DISA) mandate that all VoIP network elements support this capability.

MLPP is a service that allows properly validated users to preempt lower-priority phone calls either to targeted stations or through fully subscribed shared resources such as time-division multiplexing (TDM) trunks or conference bridges. With this capability, high-ranking personnel are ensured communication to critical organizations and personnel during a national emergency.

Connections and resources that belong to a call from an MLPP subscriber are marked with a precedence level and domain identifier and are preempted only by calls of higher precedence from MLPP users in the same domain. The DSN switch sets the maximum precedence level for a subscriber statically. When that subscriber originates a call, it is tagged with that precedence level (if none is provided) or with one that the user provides.

Cisco IOS gateways act as transit trunking network elements to map incoming precedence levels to outgoing signaling. This does not provide any schemes to configure the maximum levels for the subscriber lines, or interpret the levels based on the prefixes when a call is offered through a channel-associated signaling/R2 (CAS/R2) type of interface.

Precedence and Service Domains for the SIP

Precedence provides for preferred handling of call-service requests. It involves assigning and validating priority levels to calls and prioritized treatment of MLPP service requests. The nature of precedence assignment is based on an internal decision, in that the user chooses to apply or not to apply assigned precedence level to a call. MLPP precedence is unrelated to other call admission control (CAC) or enhanced emergency services (E911) services. User invocation of an MLPP request is provided through dedicated dial access codes and selectors in the dial string. In particular, a precedence call is requested by the user using the string prefix NP, where P is the requested precedence level and N is the preconfigured MLPP access digit.

The range of precedence values in DSN/Public SS7 Network Format (DSN/Q.735) service domains is shown in the table below.

Table 43: Range of DSN/Q.735 Precedence Values

Precedence Level	Precedence
0	FLASH OVERRIDE

Precedence Level	Precedence
1	FLASH
2	IMMEDIATE
3	PRIORITY
4	ROUTINE

The Defense Red Switched Network (DRSN) service domain has six levels of precedence as shown in the table below.

Table 44: Range of DRSN Precedence Values

Precedence Level	Precedence
0	FLASH OVERRIDE OVERRIDE
1	*FLASH OVERRIDE*
2	FLASH
3	IMMEDIATE
4	PRIORITY
5	ROUTINE

A subscriber A (0100) calling B (0150) that wants to explicitly associate a precedence level (priority) to a particular call, would dial the following digits:

8555-3-0150

^ ^ ^

| | |_____ Called number

| |_____ Call precedence--priority

|_____ MLPP service prefix

If subscriber A is an ordinary customer with an assigned precedence level of 4 (routine), then MLPP automatically treats this call as a routine call.

In SIP and ISDN signaling, however, the precedence levels and domain-name space information are carried discretely in the protocol messages and do not require appropriate prefixes to the dialed digits.

Precedence-Level Support in SIP Signaling

MLPP information in a SIP signal is carried in the Resource-Priority header. The header field marks a SIP request as desiring prioritized resource access depending on the precedence level invoked or assigned to the call originator. The syntax for the Resource-Priority header field is as follows:

Resource-Priority=“Resource-Priority” HCOLON Resource-value * (COMMA Resource-value)

Resource-value=namespace “.” r-priority

namespace=*(alphanum / “-”)

r-priority=*(alphanum / “-”)

Three name spaces are defined by the draft to cater to different service domains:

- dsn--Adopted by U.S. Defense Switched Network and DISA. It defines five precedence values: routine, priority, immediate, flash, flash-override.
- q735--Used by Signaling System 7 (SS7) networks based on ITU Q.735.3. It also defines five precedence values: 4, 3, 2, 1, 0.
- drsn--Used in U.S. DRSN. It defines six precedence values: routine, priority, immediate, flash, flash-override, flash-override-override.

The Cisco IOS gateway supports all three name spaces. In order to facilitate interworking with those network elements that support any one type of name space, the name space is configurable.

Precedence-Level Support in ISDN Signaling

MLPP service is provided by the user using the precedence information element (IE) 41 to carry the precedence levels MLPP service domain in the SETUP message. The standard specifies five level values represented by four bits and only one domain indicator value (0000000--dsn).

Mapping of DRSN name space values into ISDN poses a problem because the standard does not provide a unique value for flash-override-override. The flash-override-override value is represented as 1000 (8). When you use the most significant bit of the four-bit representation, this information is conveyed to other implementations that interpret or support flash-override-override and also ensure that the call is still treated as the highest priority with those implementations that do not use the most significant bit (MSB).

Preemption for the SIP

Preemption is the termination of existing calls of lower precedence and extension of a call of higher precedence to or through a target device. Precedence includes notification and acknowledgment of preempted users and reservation of any shared resources immediately after preemption and before preempted call termination.

Preemption takes one of two forms:

- The called party may be busy with a lower precedence call, which must be preempted in favor of completing higher precedence call from the calling party.
- Network resources may be busy with calls, some of which are of lower precedence than the calls requested by the calling party. One or more of these lower-precedence calls is preempted to complete the higher-precedence call.

Cisco IOS gateways do not implement any type of preemption service logic; that task wholly rests with the DSN switch.

Network Solution and System Flows for the SIP

The figure below shows the system flow for a typical user scenario.

The request from a higher priority interrupts a lower-priority usage at a user terminal, such as an IP phone.

The Primary Rate Interface (PRI) is connected to a DSN WAN through a Cisco IOS gateway. For the purpose of this illustration, we assume that users A, B, C, and D are properly configured in the DSN switch with the appropriate maximum priority levels, as follows:

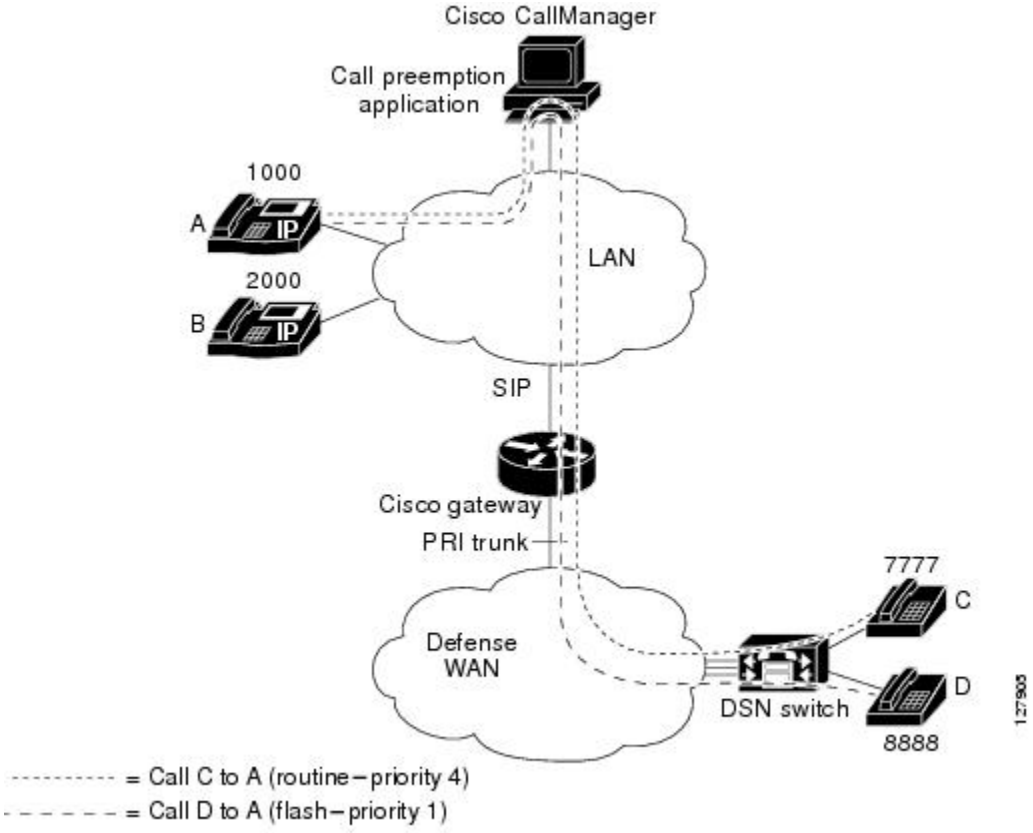
- User A--Priority FLASH

- User B--Priority IMMEDIATE
- User C--Priority ROUTINE
- User D--Priority FLASH

User C establishes a call with User A. User C wants the call to be set to the maximum priority value and so dials the appropriate code. User D tries to call a security advisor. Because User D's call has the highest priority, the figure below provides an overall illustration on how MLPP works in this scenario and the part this new functionality on the Cisco IOS gateway plays in achieving MLPP.

Note The MLPP application is an existing feature and is leveraged while interworking with Cisco IOS gateways through SIP.

Figure 68: Network Solution and System Flow



A typical network solution and call flow is as follows (see the figure above):

- User C (7777) calls User A (1000) and they are in a routine call.
- User D (8888) wants to call User A (1000). User D (8888) goes off-hook and dials 1000.
- The DSN switch interprets the call priority by looking at the dialed digits and maps to the ISDN SETUP message:

Precedence IE: Level - 0000, Service Domain - 0000000

The call is routed through the Cisco IOS gateway.

- The Cisco IOS gateway passes the incoming call with the dialed digits. It maps the incoming ISDN Precedence IE to the Resource-Priority header values. The management system applies the local policy (as configured on the gateway) to choose the name space where it wants to represent the levels:

INVITE Resource-Priority: dsn.flash

The management system validates the dialed number (DN) and identifies that User A (1000) is already in a call.

- The management system looks at the precedence level of the call from User D (8888) (FLASH) and compares this with the precedence of the existing call between User A (1000) and User C (7777) (ROUTINE).
- The management system decides to preempt the User C (7777) to User A (1000) call.

There are two options for the management system to provide the treatment to User C (7777). It either provides the preemption tone from its end if it was transcoding and Real-Time Transport Protocol (RTP) streams were controlled by it. Or if the RTP streams were directly established between the endpoints gateway and the phone, it inserts a suitable cause value in the SIP Reason header and lets the gateway or the DSN switch provide the treatment. The management system presents the cause value either in new Reason header name space preemption or in Q.850 format:

BYE Reason: Preemption; cause=1;text="UA_Preemption"

- The application marks the endpoint User A available for reuse and offers User D's call to it.
- If User D (8888) disconnects the call, the management system clears the resources and preserves the existing call between User A (1000) and User C (7777).

If a higher precedence call comes in during the User D (8888) to User A (1000) call, the management system processes the higher-order preemption. For the IP phones, when a user's profile is assigned to the phone, calls initiated from the phone inherit the precedence of the assigned user.

The next two figures show examples of a Resource-Priority (R-P) header call flows with loose mode and strict mode selected.

Note In the loose mode, unknown values of name space or priority values received in the Resource-Priority header in SIP requests are ignored by the gateway. The request is processed as if the Resource-Priority header were not present.

Figure 69: R-P Header Origination with Loose Mode Selected

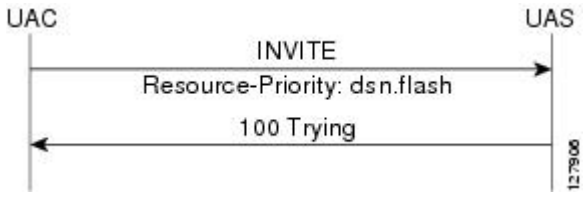
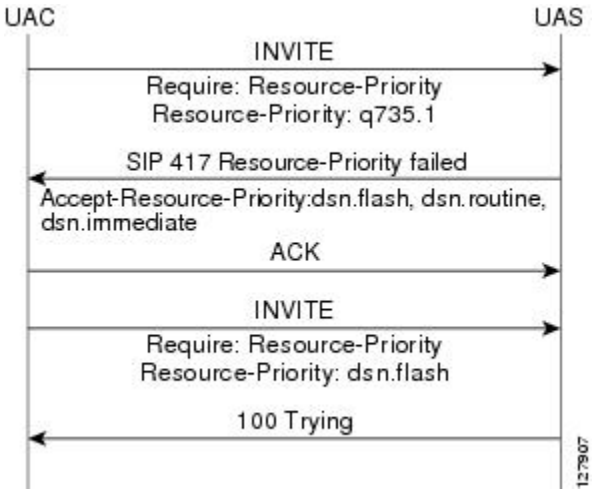


Figure 70: R-P Header Origination with Strict Mode Selected



SIP Support for Media Forking

The SIP Support for Media Forking feature provides the ability to create midcall multiple streams (or branches) of audio associated with a single call and then send those streams of data to different destinations. The feature allows service providers to use technologies such as speech recognition, voice authentication, and text-to-speech conversion to provide sophisticated services to their end-user customers. An example is a web-browsing application that uses voice recognition and text-to-speech (TTS) technology to make reservations, verify shipments, or order products.

Feature benefit is as follows: SIP media streams are created and deleted only through re-Invite messages; no CLI is required.

Media Streams

With the SIP Support for Media Forking feature, you can create up to three Real-Time Transport Protocol (RTP) media streams to and from a single DS0 channel. In addition, separate gateway destinations (IP address or UDP port) are maintained for each of the streams. The streams are bidirectional; media received from the destination gateways are mixed in the DSP before being sent to the DS0 channel, and pulse code modulation (PCM) received from the DS0 is duplicated and sent to the destination gateways.

Originating gateways establish multiple media streams on the basis of Session Description Protocol (SDP) information included in midcall re-Invites received from a destination gateway, third-party call controller, or other SIP signaling entity. Only one SIP call leg is involved in media forking at the gateway, so the SIP signaling entity that initiates the re-Invites must be capable of aggregating the media information for multiple destinations (such as IP address, port number, payload types, or codecs) into one SDP description. Multiple m-lines in the SDP are used to indicate media forking, with each m-line representing one media destination.

Note SIP media streams are created and deleted only through re-Invite messages; no specific CLI is required.

The ability to create midcall multiple streams (or branches) of audio associated with a single call and send those streams of data to different destinations is similar to a three-way or conference call. A media-forked call has some differences. For example, in a three-way call, each party hears all of the other parties. But in a

media-forked call, only the originating caller (the controller) hears the audio (voice and DTMF digits) from all the other participants. The other participants hear audio only from the originating caller and not from each other.

Another difference between a three-way call and a media-forked call is that media streams can be configured on the gateway. Three-way calls send the audio to all of the other parties involved in the call. However, media-forking permits each media stream to be independently configured. For example, one media stream to one party may include both voice and DTMF digits, whereas another media stream to another party may include only DTMF digits.

The feature supports three types of media streams: voice, DTMF-relay only, and voice plus DTMF-relay.

In addition to the following discussion, see the following as appropriate:

- For information on codecs, see "Map Payload Types to Dynamic Payload Codecs".
- For information on payload types see "Multiple Codec Selection Order and Dynamic Payload Codecs".
- For information on DTMF relay, see the "SIP INFO Method for DTMF Tone Generation" section of the "Configuring SIP DTMF Features" chapter.

Voice Media Streams

Voice-only media streams send all audio from the DS0 channel, and the audio is encoded according to the selected codec. Voice-only media streams have the following characteristics:

- DTMF digits are sent as in-band audio.
- All forked streams must use the same codec, which is referred to as simple forking.
- Only the following codecs and their variants are allowed: G.711, G.726, and G.729.
- For the G.726 codecs, dynamic payload types are negotiated in SDP. SDP messages contain capabilities information that is exchanged between gateways. The payload types must be the same for all streams in the call.

DTMF-Relay Media Streams

DTMF relay provides reliable digit relay between VoIP gateways and a standardized means of transporting DTMF tones in RTP packets. DTMF-relay media streams have the following characteristics:

- DTMF-relay media streams do not include voice and do not use a codec. DTMF-relay packets are sent when the originating party presses a DTMF digit.
- Only RTP-NTE can be used in a forked DTMF-relay call. RTP-NTE is used to transport DTMF digits and other telephony events between two endpoints. RTP-NTE prevents the generation of spurious digits at the receiving gateway and is further described in RFC 2833.
- DTMF-relay streams are supported only on calls with two established streams and can appear only as the second stream.
- The payload-type value assigned to DTMF-relay packets in SDP must be the same for all streams that use DTMF-relay. The default payload type for Cisco gateways is 101.

Voice Plus DTMF-Relay Media Streams

Voice plus DTMF-relay media streams send both encoded voice and DTMF-relay packets and have the following characteristics:

- The receiving gateway can distinguish the voice component from the DTMF component.
- Unlike DTMF-relay, voice plus DTMF is supported on any of the established streams of a forked call.
- Only RTP-NTE can be used in a forked DTMF-relay call.
- All streams must use the same codec (simple forking).
- Only the following codecs and their variants are allowed: G.711, G.726, and G.729.
- For the G.726 codecs, dynamic payload types are negotiated in SDP. SDP messages contain capabilities information that are exchanged between gateways. The payload types must be the same for all streams in the call.
- The payload-type value assigned to DTMF-relay packets in SDP must be the same for all streams that use DTMF-relay. The default payload type for Cisco gateways is 101.

Multiple Codec Selection Order and Dynamic Payload Codecs

When using multiple codecs you must create a voice class in which you define a selection order for codecs, and you can then apply the voice class to VoIP dial peers. The **voice class codec** command in global configuration mode allows you to define the voice class that contains the codec selection order. Then you use the **voice-class codec** command in dial-peer configuration mode to apply the class to individual dial peers.

If there are any codecs that use dynamic payload types (g726r16, g726r24), Cisco IOS software assigns the payload types to these codecs in the order in which they appear in the configuration, starting with the first available payload type in the dynamic range. The range for dynamic payload types is from 96 to 127, but Cisco IOS software preassigns the following payload types by default.

Table 45: Codec Dynamic Payload Types and Function

Range	Function
96	fax
97	fax-ack
100	NSE
101	NTE
121	DTMF-relay
122	Fax-relay
123	CAS
125	ClearChan

Because the payload types have been reserved by the default assignments shown in the table, Cisco IOS software automatically assigns 98 to the first dynamic codec in the dial-peer configuration, 99 to the second, and 102 to the third.

Some of these preassigned payload types can be changed with the **modem relay** command. This command allows changes to the available payload types that can be used for codecs.

For outgoing calls on the originating gateway, all of the codecs that are configured in the codec list used by the dial peer are included in the SDP of the Invite message.

On the terminating gateway, Cisco IOS software always uses the dynamic payload types that the originating gateway specified in the SDP of the Invite message. This practice avoids the problem of misaligned payload types for most call types. The exception is when a delayed-media Invite message is received. A delayed-media Invite can be used by a voice portal to signal a terminating gateway before re-Inviting a forking gateway. If a delayed-media Invite is used, the Invite message does not contain SDP information, and the terminating gateway must advertise its own codecs and payload types. It does this in the SDP of its response message (either a 183 or a 200 OK). The terminating gateway assigns payload types to dynamic codecs using the same rules as the originating gateway. However, if there is a difference in either the preassigned dynamic payload types or the order in which the dynamic codecs are listed in the codec list used by the dial peer, the payload types may not be assigned consistently on the originating and terminating gateways. If the terminating gateway selects a different payload type for a dynamic codec, the call may fail.

If a G.726 codec is assigned in the first active stream of the call, there are some scenarios in which the voice portal sends a delayed-media re-Invite message to the second or third terminating gateway. Then, it is necessary to ensure that the originating gateway and the second and third terminating gateways have the same preassigned payload types and the same order of dynamic codecs in the codec list for the dial peer being used for the call. Otherwise, the added media stream may be rejected by the originating gateway if the payload types do not match.

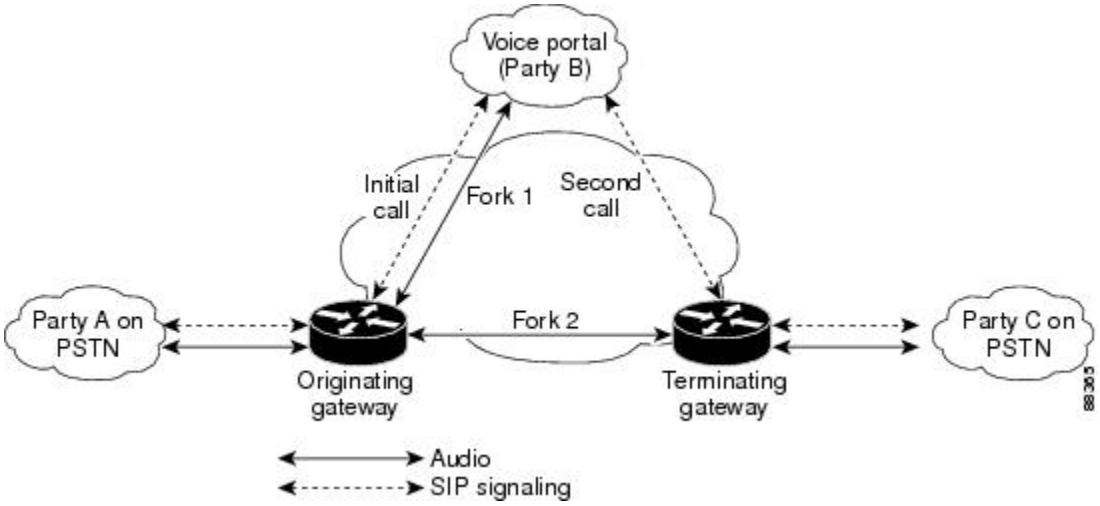
Media Forking Applications

A web-browsing application that uses voice recognition and text-to-speech (TTS) technology to make reservations, verify shipments, or order products is a typical application of media forking. In The figure below, a client (Party A) uses a telephone to browse the web. Party A calls the voice portal (Party B), and the call is routed through the originating gateway. The voice portal operates like a standard voice gateway and terminates calls to a voice response system that has voice recognition and TTS capabilities. This voice response system takes input from Party A by DTMF digits or voice recognition and returns responses (for example, stock quotes retrieved from the web) to Party A.

The voice portal, or Party B, is also capable of third-party call control (3pcc) and can set up a call between Party A and a third participant (Party C) without requiring direct signaling between Party A and Party C. One example of a possible call between Party A and Party C is if Party A found a restaurant listing while browsing the web and wanted to speak directly to the restaurant to make reservations.

Another feature of the voice portal is that once the call between parties A and C is established, the voice portal can continue to monitor the audio from Party A. By doing so, the voice portal can terminate the connection between Party A and Party C when a preestablished DTMF digit or voice command is received. Party B retains the connection between itself and Party A, in case Party A has any further requests. Continuing with the restaurant example, the continuous connection is important if Party A decides to query yet another restaurant. Party A simply goes back to the connection with Party B, who sets up a call with the new restaurant.

Figure 71: Media Forking Application

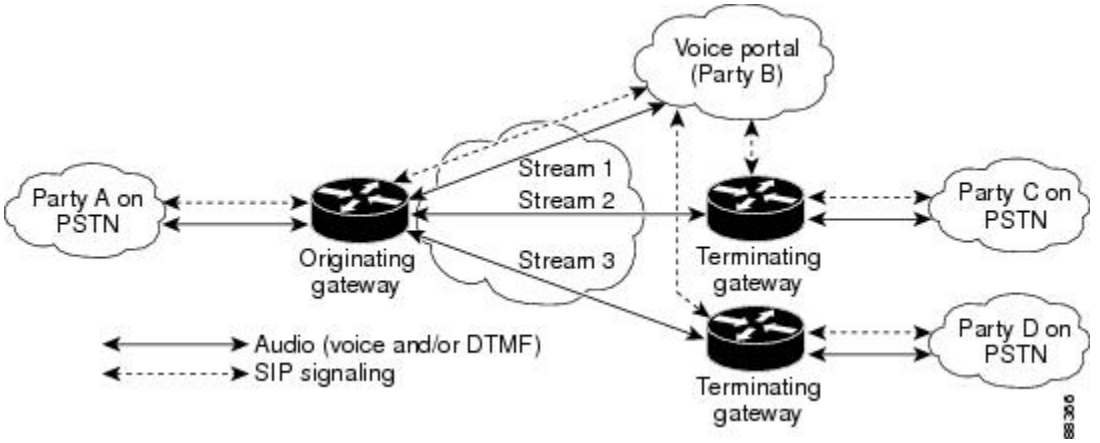


Another important aspect of media forking is that although there can be more than one media destination, there is only one signaling destination (in this case, the voice portal). The call leg that was originally set up (from the originating gateway to the voice portal) is maintained for the life of the session. The media destinations are independent of the signaling destination, so media streams (or new destinations) can be added and removed dynamically through re-Invite messages. Media streams are created and deleted only through re-Invite messages rather than through any CLIs.

If you configure the **timer receive-rtcp**command for a gateway, a Session Initiation Protocol (SIP) media inactivity timer is started for each active media stream. The timer monitors and disconnects calls if no RTCP packets are received within a configurable time period. If any of the timers expire, the entire call is terminated--not just the stream on which the timer expired. If a stream is put on call hold, the timer for that stream is stopped. When the stream is taken off hold, the timer for that stream is started again.

There is a maximum of three VoIP media streams that can be established per call. The figure below shows the maximum number of streams.

Figure 72: Multiple Streams



Media Forking Initiation

Media forking is initiated by specifying multiple media fields (m-lines) in the SDP of a re-Invite message. The rules for adding and deleting multiple m-lines conform to RFC 2543, SIP: Session Initiation Protocol Appendix B .

Multiple streams are not created through CLIs.

How to Configure SIP Connection-Oriented Media Forking and MLPP Features

Note For help with a procedure, see the troubleshooting section listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring SIP Connection-Oriented Media Enhancements for SIP

Note The following steps enable the gateway to check the media source address and port of the first incoming RTP packet, and optionally to specify whether the endpoint is active or passive. Once the media source check is enabled, the gateway modifies or updates the established VoIP RTP session with upstream addressing information extracted from the SDP body of the received request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **nat symmetric check-media-source**
5. **nat symmetric role** {**active** | **passive**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**nat symmetric check-media-source** **Example:** `Router(config-sip-ua)# nat symmetric check-media-source`	Enables the gateway to check the media source address and port of incoming Real-time Transport Protocol (RTP) packets in symmetric NAT environments.
Step 5	**nat symmetric role** {**active** | **passive**} **Example:** `Router(config-sip-ua)# nat symmetric role active`	(Optional) Defines endpoint settings to initiate or accept a connection for symmetric NAT configuration. Keywords are as follows: • **active** --Symmetric NAT endpoint role is active, enabling the endpoint to initiate an outgoing connection. • **passive** --Symmetric NAT endpoint role is passive, enabling the endpoint to accept an incoming connection to the port number on the m= line of the Session Description Protocol (SDP) body of the other endpoint. This is the default.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP Multilevel Precedence and Priority Support

To configure multilevel precedence and priority support on SIP for a VoIP dial peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **destination-pattern** [+]*string*[**T**]
5. **voice-class sip resource priority namespace** [**drsn** | **dsn** | **q735**]
6. **voice-class sip resource priority mode** [**loose** | **strict**]
7. **session protocol sipv2**
8. **session target ipv4:** *destination-address*

9. **session transport** {**udp** | **tcp**}
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 100 voip`	Enters dial-peer voice configuration mode for the specified VoIP dial peer.
Step 4	**destination-pattern** [+]*string*[**T**] **Example:** `Router(config-dial-peer)# destination-pattern 7777`	Enters either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows: • + --Character indicating an E.164 standard number. • *string* --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries: digits 0 to 9, letters A to D, and any special character. • **T** --Control character indicating that the destination-pattern value is a variable-length dial string.
Step 5	**voice-class sip resource priority namespace** [**drsn** | **dsn** | **q735**] **Example:** `Router(config-dial-peer)# voice-class sip resource priority namespace dsn`	Specifies mandatory call-prioritization handling for the initial INVITE message request and specifies a service domain namespace. Keywords are as follows: • **namespace** --Service domain-name space • **drsn** --U.S. RDSN format • **dsn** --U.S. DSN format • **q735** --Public signaling SS7 network format **Note** If the gateway is originating the SIP call, configure the priority values in any of the supported name spaces under the outgoing VoIP dial peer. This decision is based on the gateway's connection to an appropriate domain.

	Command or Action	Purpose
Step 6	**voice-class sip resource priority mode** [**loose** | **strict**] **Example:** `Router(config-dial-peer)# voice-class sip resource priority mode loose`	Specifies mandatory call-prioritization handling for the initial INVITE message request and specifies a resource priority-handling mode. Keywords are as follows: • **mode** --Resource priority handling mode • **loose** --Loose resource priority handling • **strict** --Strict resource priority handling **Note** The originating gateway indicates the receiving SIP endpoint to either handle the call in the indicated priority or ignore the call-priority values if the receiving endpoint fails to understand either the name space domain or the precedence value.
Step 7	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Specifies use of Internet Engineering Task Force (IETF) SIP.
Step 8	**session target ipv4:** *destination-address* **Example:** `Router(config-dial-peer)# session target ipv4:10.10.1.3`	Specifies the network-specific address for the dial peer.
Step 9	**session transport** {**udp** | **tcp**} **Example:** `Router(config-dial-peer)# session transport udp`	Specifies use of a particular session-transport protocol. Keywords are as follows: • **udp** -- User Datagram Protocol (UDP) • **tcp** --Transport Layer Protocol (TCP) The default is UDP.
Step 10	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring SIP Support for Media Forking

Configure Codec Complexity

To configure codec complexity on a Cisco 2600 series, Cisco 3600 series, Cisco 37xx, or Cisco AS5300, or Cisco 7200 series, perform one of the following tasks, according to your router type.

Cisco 2600 Series Cisco 3600 Series Cisco 37xx and Cisco AS5300

For routers that have already been configured but need their codec complexity changed to high: If there is a DS0 group or PRI group assigned to any T1 controllers on the card, the DS0 or PRI groups must be removed. To remove the groups, shut down the voice ports associated with the groups; then follow the procedure below.

Configuring the correct codec complexity is required for media-forked calls. For the Cisco AS5300, codec complexity is determined by the VCWare code that is loaded on the voice feature card (VFC). To download Cisco VCWare software, see the Cisco software download page.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice card** *slot*
4. **codec complexity** {**high** | **medium**} [**ecan-extended**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice card** *slot* **Example:** `Router(config)# voice-card 1`	Enters voice-card configuration mode for the specified voice-card slot location.
Step 4	**codec complexity** {**high** | **medium**} [**ecan-extended**] **Example:** `Router(config-voice-card)# codec complexity high`	Specifies call density and codec complexity according to the codec standard that is being used. Set codec complexity as follows: • Cisco 2691 and Cisco 2600XM series: Set to high for full forking functionality. With high-complexity packaging, each DSP supports two voice channels. • Cisco 3600 series and Cisco 37xx: Set to high for full forking functionality.
Step 5	**exit** **Example:** `Router(config-voice-card)# exit`	Exits the current mode.

Cisco 7200 Series

SUMMARY STEPS

1. **enable**
2. **show interfaces dspfarm** [*slot* / *port*] **dsp** [*number*] [**long** | **short**]
3. **configure terminal**
4. **dspint dspfarm** *slot* / *port*
5. **codec high**
6. **description** *string*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interfaces dspfarm** [*slot* / *port*] **dsp** [*number*] [**long**	**short**] **Example:** `Router# show interface dspfarm 3/0`	Displays DSP voice-channel activity. You cannot change codec complexity if any voice channels are busy; you can do so only if all DSP channels are idle. Keywords and arguments are as follows: • *slot* / *port* --Slot location and port number on the port adapter. • **dsp** *number* --Number of DSP sets to display. Range: 1 to 30. • **long** --Detailed DSP information. • **short** --Brief DSP information.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**dspint dspfarm** *slot* / *port* **Example:** `Router(config)# dspint dspfarm 2/0`	Enters DSP-interface configuration mode for the specified slot/port.
Step 5	**codec high** **Example:** `Router(config-dspfarm)# codec high`	Specifies call density and codec complexity based on a particular codec standard. • Use the **high** keyword with the SIP Support for Media Forking feature. The **high** keyword supports two encoded voice channels. For this feature, the following

	Command or Action	Purpose
		codecs and their variants are supported: G.711, G.726, and G.729.
Step 6	**description** *string* **Example:** `Router(config-dspfarm)# description marketing_dept`	Includes a specific description (string) about the DSP interface. • This information is displayed in the output and does not affect operation of the interface in any way.
Step 7	**exit** **Example:** `Router(config-dspfarm)# exit`	Exits the current mode.

Map Payload Types to Dynamic Payload Codecs

Note The process used by Cisco IOS software to map payload types to dynamic payload codecs is important in media-forked calls because all media streams must use the same payload type.

- VoIP dial peers can list codecs in either of two ways, depending on whether a single codec or multiple codecs are to be assigned to the dial-peer. The following steps configure a single codec in dial-peer mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **codec** *codec* [**bytes** *payload-size*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified dial peer.
Step 4	**codec** *codec* [**bytes** *payload-size*] **Example:** `Router(config-dial-peer)# codec g729r8`	Specifies a codec for the dial peer.Keyword and arguments are as follows: • *codec* --Type of codec. Valid values for use with media forking are the following: • g711alaw • g711ulaw • g726r16 • g726r24 • g726r32 • g729br8 • g729r8 (default) • **bytes** *payload-size* -- Number of bytes in the voice payload of each frame. Values depend on codec type and packet voice protocol.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure Multiple-Codec Selection Order

To configure multiple-codec selection order, perform the following steps.

Note With multiple codecs, you can create a voice class in which you define a selection order for codecs, and you can then apply the voice class to VoIP dial peers. The following procedures create a voice class. For the complete dial-peer configuration procedure, see the *Cisco IOS Voice Command Reference*, Release 12.3 .

- You can configure more than one voice class codec list for your network. Configure the codec lists apply them to one or more dial peers based on what codecs (and the order) you want supported for the dial peers. You need to define selection order if you want more than one codec supported for a given dial peer.
- SIP gateways do not support a codec preference order with H.323 signaling; all codecs listed are given equal preference. In particular, they do not prefer g729r8 over g729br8 if both are defined.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **voice class codec** *tag*
4. **codec preference** *value codec-type* [**bytes** *payload-size*]
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **voice-class codec** *tag*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice class codec** *tag* **Example:** `Router(config)# voice class codec 99`	Enters voice-class configuration mode and assigns an identification tag number for a codec voice class. The argument is as follows: • *tag* --Unique identifier on the router. Range: 1 to 10000.
Step 4	**codec preference** *value codec-type* [**bytes** *payload-size*] **Example:** `Router(config-voice-class)# codec preference 1 g711alaw`	Specifies a list of preferred codecs to use on a dial peer. Keywords and arguments are as follows: • *value* --Order of preference, with 1 being the most preferred and 14 being the least preferred. • *codec-type* --Preferred codec. • **bytes** *payload-size* -- Number of bytes in the voice payload of each frame. Values depend on codec type and packet voice protocol. **Note** SIP gateways do not support a codec preference order with H.323 signaling; all codecs listed are given equal preference. Specifically, they do not prefer g729r8 over g729br8 if both are defined.
Step 5	**exit** **Example:** `Router(config-voice-class)# exit`	Exits the current mode.

	Command or Action	Purpose
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 16 voip`	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 7	**voice-class codec** *tag* **Example:** `Router(config-dial-peer)# voice-class codec 99`	Assigns a previously configured codec selection preference list (the codec voice class that you defined in step 3 above) to the specified VoIP dial peer. **Note** The **voice-class codec** command in dial-peer configuration mode contains a hyphen. The **voice class** command in global configuration mode does not contain a hyphen.
Step 8	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Verifying Connection-Oriented Media and Forking Features for SIP

To verify configuration of connection-oriented media and forking features for SIP, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show dial-peer voice sum**
2. **show running-config**
3. **show sip-ua calls**
4. **show voice dsp**

DETAILED STEPS

Step 1 show dial-peer voice sum

Use this command to verify dial-peer configuration.

Example:

```
Router# show dial-peer voice sum
dial-peer hunt 0
AD PRE PASS
TAG TYPE MIN OPER PREFIX DEST-PATTERN FER THRU SESS-TARGET PORT
110 voip up up    555110. 0 syst ipv4:172.18.195.49
210 voip up up    555330. 0 syst ipv4:172.18.195.49
200 pots up up    5553300 0 2/0/1
101 pots up up    5551100 0 2/0/0
366 voip up up    366.... 0 syst ipv4:172.18.195.49
```

Step 2 **show running-config**

Use this command to display the contents of the currently running configuration file or the configuration for a specific interface.

On a Cisco 2600 series, Cisco 3600 series, Cisco 37xx, or Cisco 7200 series, use this command to verify codec complexity. (For the Cisco AS5300, codec complexity depends on what VCWare image is loaded on the voice feature card.) Command output displays the current voice-card setting. If medium-complexity is specified, the codec complexity setting does not display. If high-complexity is specified, the setting "codec complexity high" displays.

The following sample output shows that high-complexity mode of operation for full media-forking functionality is specified.

Example:

```
Router# s
how running-config
Building configuration...
Current configuration : 1864 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
memory-size iomem 10
voice-card 1
 codec complexity high
!
ip subnet-zero
```

Step 3 **show sip-ua calls**

Use this command to display user-agent client (UAC) and user-agent server (UAS) information on SIP calls. Command output includes information about each media stream (up to three streams for media-forked calls). It is useful in debugging, because it shows if an active call is forked.

The following sample output shows UAC and UAS information on SIP calls. Command output includes information about each media stream (up to three streams for media-forked calls). It is useful in debugging, because it shows if an active call is forked.

Example:

```
Router# show sip-ua calls
SIP UAC CALL INFO
Call 1
SIP Call ID : 515205D4-20B711D6-8015FF77-1973C402@172.18.195.49
 State of the call : STATE_ACTIVE (6)
 Substate of the call : SUBSTATE_NONE (0)
 Calling Number : 555 0200
 Called Number : 5551101
 Bit Flags : 0x12120030 0x220000
 Source IP Address (Sig ): 172.18.195.49
 Destn SIP Req Addr:Port : 172.18.207.18:5063
 Destn SIP Resp Addr:Port: 172.18.207.18:5063
 Destination Name : 172.18.207.18
 Number of Media Streams : 4
 Number of Active Streams: 3
 RTP Fork Object : 0x637C7B60
 Media Stream 1
  State of the stream : STREAM_ACTIVE
  Stream Call ID : 28
```

```
  Stream Type : voice-only (0)
  Negotiated Codec : g711ulaw (160 bytes)
  Codec Payload Type : 0
  Negotiated Dtmf-relay : inband-voice
  Dtmf-relay Payload Type : 0
  Media Source IP Addr:Port: 172.18.195.49:19444
  Media Dest IP Addr:Port : 172.18.193.190:16890
 Media Stream 2
  State of the stream : STREAM_ACTIVE
  Stream Call ID : 33
  Stream Type : voice+dtmf (1)
  Negotiated Codec : g711ulaw (160 bytes)
  Codec Payload Type : 0
  Negotiated Dtmf-relay : rtp-nte
  Dtmf-relay Payload Type : 101
  Media Source IP Addr:Port: 172.18.195.49:18928
  Media Dest IP Addr:Port : 172.18.195.73:18246
 Media Stream 3
  State of the stream : STREAM_ACTIVE
  Stream Call ID : 34
  Stream Type : dtmf-only (2)
  Negotiated Codec : No Codec (0 bytes)
  Codec Payload Type : -1 (None)
  Negotiated Dtmf-relay : rtp-nte
  Dtmf-relay Payload Type : 101
  Media Source IP Addr:Port: 172.18.195.49:18428
  Media Dest IP Addr:Port : 172.16.123.99:34463
 Media Stream 4
  State of the stream : STREAM_DEAD
  Stream Call ID : -1
  Stream Type : dtmf-only (2)
  Negotiated Codec : No Codec (0 bytes)
  Codec Payload Type : -1 (None)
  Negotiated Dtmf-relay : rtp-nte
  Dtmf-relay Payload Type : 101
  Media Source IP Addr:Port: 172.18.195.49:0
  Media Dest IP Addr:Port : 172.16.123.99:0
 Number of UAC calls: 1
SIP UAS CALL INFO
 Number of UAS calls: 0
```

Step 4 **show voice dsp**

Use this command to display the current status of all DSP voice channels, including codecs.

Example:

```
Router# show voice dsp
.
.
.
DSP#0: state IN SERVICE, 2 channels allocated
channel#0: voice port 1/0, codec G711 ulaw, state UP
channel#1: voice port 1/1, codec G711 ulaw, state UP
DSP#1: state IN SERVICE, 2 channels allocated
channel#0: voice port 2/0, codec G711 ulaw, state UP
channel#1: voice port 2/1, codec G711 ulaw, state UP
DSP#2: state RESET, 0 channels allocated
```

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "General Troubleshooting Tips" section.

- Make sure that you can make a voice call.
- If you are using codec types g726r16 or g726r24, use the **debug voip rtp session named-event 101** command for DTMF-relay debugging. Be sure to append the argument *101* to thecommand to prevent the console screen from flooding with messages and all calls from failing.
- Use the **debug ccsip**family of commands for general SIP debugging, including viewing the direction-attribute settings and port and network address-translation traces.

Following is sample output for some of these commands:

Sample Output for the debug ccsip all Command

In the following example, output is displayed with the **role** keyword of the **nat symmetric** command set to active for the originating gateway, and to passive for the terminating gateway.

```
Router# debug ccsip all
All SIP call tracing enabled
Router#
00:02:12:0x6327E424 :State change from (UNDEFINED, SUBSTATE_NONE)  to (STATE_IDLE,
SUBSTATE_NONE)
00:02:12:****Adding to UAC table
00:02:12:adding call id 3 to table
00:02:12:Queued event from SIP SPI :SIPSPI_EV_CC_CALL_SETUP (10)
00:02:12:CCSIP-SPI-CONTROL: act_idle_call_setup
00:02:12: act_idle_call_setup:Not using Voice Class Codec
00:02:12:act_idle_call_setup:preferred_codec set[0] type :g711ulaw bytes:160
00:02:12:sipSPICopyPeerDataToCCB:From CLI:Modem NSE payload = 100, Passthrough = 0,Modem
relay = 0, Gw-Xid = 1
SPRT latency 200, SPRT Retries = 12, Dict Size = 1024
 String Len = 32, Compress dir = 3
00:02:12:****Deleting from UAC table
00:02:12:****Adding to UAC table
00:02:12:Queued event from SIP SPI :SIPSPI_EV_CREATE_CONNECTION (6)
00:02:12:0x6327E424 :State change from (STATE_IDLE, SUBSTATE_NONE)  to (STATE_IDLE,
SUBSTATE_CONNECTING)
00:02:12:0x6327E424 :State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to (STATE_IDLE,
SUBSTATE_CONNECTING)
00:02:12:sipSPIUsetBillingProfile:sipCallId for billing records =
D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
00:02:12:CCSIP-SPI-CONTROL: act_idle_connection_created
00:02:12:CCSIP-SPI-CONTROL: act_idle_connection_created:Connid(1) created to
172.18.200.237:5060, local_port 56992
00:02:12:CCSIP-SPI-CONTROL: sipSPIOutgoingCallSDP
00:02:12: Preferred method of dtmf relay is:6, with payload :101
00:02:12: convert_codec_bytes_to_ptime:Values :Codec:g711ulaw codecbytes :160, ptime:20
00:02:12:sip_generate_sdp_xcaps_list:Modem Relay disabled. X-cap not needed
00:02:12:CCSIP-SPI-CONTROL: Clock Time Zone is UTC, same as GMT:Using GMT
00:02:12:sipSPIAddLocalContact
00:02:12:Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE (7)
00:02:12:sip_stats_method
00:02:12:0x6327E424 :State change from (STATE_IDLE, SUBSTATE_CONNECTING)  to
```

```
(STATE_SENT_INVITE, SUBSTATE_NONE)
00:02:12:Sent:
INVITE sip:2021010124@172.18.200.237:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>
Date:Mon, 01 Mar 1993 00:02:12 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Supported:timer,100rel
Min-SE: 1800
Cisco-Guid:3563045876-351146444-2147852364-2382746380
User-Agent:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Max-Forwards:1
Timestamp:730944132
Contact:<sip:888001@10.15.66.43:5060;user=phone>
Expires:60
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:291
v=0
o=CiscoSystemsSIP-GW-UserAgent 9502 9606 IN IP4 10.15.66.43
s=SIP Call
c=IN IP4 10.15.66.43
t=0 0
m=audio 16398 RTP/AVP 0 100 101
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=direction:active
00:02:12:CCSIP-SPI-CONTROL: act_sentinvite_wait_100
00:02:12:CCSIP-SPI-CONTROL: Clock Time Zone is UTC, same as GMT:Using GMT
00:02:12:sipSPIAddLocalContact
00:02:12:Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE (7)
00:02:12:sip_stats_method
00:02:12:Sent:
INVITE sip:2021010124@172.18.200.237:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>
Date:Mon, 01 Mar 1993 00:02:12 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Supported:timer,100rel
Min-SE: 1800
Cisco-Guid:3563045876-351146444-2147852364-2382746380
User-Agent:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Max-Forwards:1
Timestamp:730944132
Contact:<sip:888001@10.15.66.43:5060;user=phone>
Expires:60
Allow-Events:telephone-event
Content-Type:application/sdp
Content-Length:291
v=0
o=CiscoSystemsSIP-GW-UserAgent 9502 9606 IN IP4 10.15.66.43
s=SIP Call
c=IN IP4 10.15.66.43
t=0 0
m=audio 16398 RTP/AVP 0 100 101
a=rtpmap:0 PCMU/8000
```

```
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=direction:active
00:02:12:Received:
SIP/2.0 100 Trying
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Tue, 04 Jan 2000 23:57:53 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Timestamp:730944132
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow-Events:telephone-event
Content-Length:0
00:02:12:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:172.18.200.237:5060
00:02:12:CCSIP-SPI-CONTROL: act_sentinvite_new_message
00:02:12:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:02:12:sip_stats_status_code
00:02:12: Roundtrip delay 32 milliseconds for method INVITE
00:02:12:0x6327E424 :State change from (STATE_SENT_INVITE, SUBSTATE_NONE)  to
(STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
00:02:13:Received:
SIP/2.0 183 Session Progress
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Tue, 04 Jan 2000 23:57:53 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Timestamp:730944132
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Require:100rel
RSeq:5975
Allow-Events:telephone-event
Contact:<sip:2021010124@172.18.200.237:5060;user=phone>
Content-Type:application/sdp
Content-Disposition:session;handling=required
Content-Length:240
v=0
o=CiscoSystemsSIP-GW-UserAgent 1692 40 IN IP4 172.18.200.237
s=SIP Call
c=IN IP4 172.18.200.237
t=0 0
m=audio 16898 RTP/AVP 0 100
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
a=direction:passive
00:02:13:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:172.18.200.237:5060
00:02:13:CCSIP-SPI-CONTROL: act_recdproc_new_message
00:02:13:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:02:13:sip_stats_status_code
00:02:13: Roundtrip delay 708 milliseconds for method INVITE
00:02:13:sipSPIGetSdpBody :Parse incoming session description
00:02:13:HandleSIP1xxSessionProgress:Content-Disposition received in 18x
response:session;handling=required
00:02:13:sipSPIDoFaxMediaNegotiation()
00:02:13:sipSPIDoMediaNegotiation:Codec (g711ulaw) Negotiation Successful on Static Payload
00:02:13: sipSPIDoPtimeNegotiation:One ptime attribute found - value:20
```

```
00:02:13: convert_ptime_to_codec_bytes:Values :Codec:g711ulaw ptime :20, codecbytes:160
00:02:13: convert_codec_bytes_to_ptime:Values :Codec:g711ulaw codecbytes :160, ptime:20
00:02:13: Parsed the direction:role identified as:0
00:02:13:sipSPIDoDTMFRelayNegotiation:Requested DTMF-RELAY option(s) not found in Preferred
 DTMF-RELAY option list!
00:02:13: sipSPIDoMediaNegotiation:DTMF Relay mode :Inband Voice
00:02:13:sip_sdp_get_modem_relay_cap_params:
00:02:13:sip_sdp_get_modem_relay_cap_params:NSE payload from X-cap = 0
00:02:13:sip_do_nse_negotiation:NSE Payload 100 found in SDP
00:02:13:sip_do_nse_negotiation:Remote NSE payload = local one = 100, Use it
00:02:13:sip_select_modem_relay_params:X-tmr not present in SDP. Disable modem relay
00:02:13:sipSPIDoQoSNegotiation - SDP body with media description
00:02:13:sipSPIUpdCcbWithSdpInfo:SDP Media Information:
Negotiated Codec      :g711ulaw , bytes :160
Early Media           :0
Delayed Media         :0
Bridge Done           :0
New Media             :0
DSP DNLD Reqd         :0
Media Dest addr/Port  :172.18.200.237:16898
Orig Media Addr/Port  :0.0.0.0:0
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROCEEDING)
  to (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROGRESS)
00:02:13:ccsip_process_response_contact_record_route
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROGRESS)
  to (STATE_RECD_PROCEEDING, SUBSTATE_CONNECTING)
00:02:13:Queued event from SIP SPI :SIPSPI_EV_CREATE_CONNECTION (6)
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_CONNECTING)  to
(STATE_RECD_PROCEEDING, SUBSTATE_CONNECTING)
00:02:13:sipSPIRtcpUpdates:rtcp_session info
               laddr = 10.15.66.43, lport = 16398, raddr = 172.18.200.237, rport=16898
00:02:13:sipSPIRtcpUpdates:NO extraction of source address from remote media
00:02:13: sipSPIRtcpUpdates No rtp session in bridge, create a new one
00:02:13:CCSIP-SPI-CONTROL: ccsip_caps_ind
00:02:13:ccsip_get_rtcp_session_parameters:CURRENT VALUES:
ccCallID=3, current_seq_num=0x1500
00:02:13:ccsip_get_rtcp_session_parameters:NEW VALUES:
ccCallID=3, current_seq_num=0xB93
00:02:13:ccsip_caps_ind:Load DSP with negotiated codec :g711ulaw, Bytes=160
00:02:13:sipSPISetDTMFRelayMode:set DSP for dtmf-relay =
CC_CAP_DTMF_RELAY_INBAND_VOICE_AND_OOB
00:02:13:sip_set_modem_caps:Negotiation already Done. Set negotiated Modem caps
00:02:13:sip_set_modem_caps:Modem Relay & Passthru both disabled
00:02:13:sip_set_modem_caps:nse payload = 100, ptru mode = 0, ptru-codec=0, redundancy=0,
xid=0, relay=0, sprt-retry=12, latecncy=200, compres-dir=3, dict=1024, strnlen=32
00:02:13:ccsip_caps_ind:Load DSP with codec :g711ulaw, Bytes=160
00:02:13:CCSIP-SPI-CONTROL: ccsip_caps_ack
00:02:13:CCSIP-SPI-CONTROL: act_recdproc_connection_created
00:02:13:CCSIP-SPI-CONTROL: sipSPICheckSocketConnection:Connid(2) created to
172.18.200.237:5060, local_port 50689
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_CONNECTING)  to
(STATE_RECD_PROCEEDING, SUBSTATE_NONE)
00:02:13:Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE (7)
00:02:13:sip_stats_method
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_NONE)  to
(STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROGRESS)
00:02:13:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Tue, 04 Jan 2000 23:57:53 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Timestamp:730944132
```

```
Server:Cisco-SIPGateway/IOS-12.x
CSeq:101 INVITE
Allow-Events:telephone-event
Contact:<sip:2021010124@172.18.200.237:5060;user=phone>
Content-Type:application/sdp
Content-Length:240
v=0
o=CiscoSystemsSIP-GW-UserAgent 1692 40 IN IP4 172.18.200.237
s=SIP Call
c=IN IP4 172.18.200.237
t=0 0
m=audio 16898 RTP/AVP 0 100
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
a=direction:passive
00:02:13:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:172.18.200.237:5060
00:02:13:Sent:
PRACK sip:2021010124@172.18.200.237:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Mon, 01 Mar 1993 00:02:12 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
CSeq:102 PRACK
RAck:5975 101 INVITE
Content-Length:0
00:02:13:CCSIP-SPI-CONTROL: act_recdproc_new_message
00:02:13:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:02:13:sip_stats_status_code
00:02:13: Roundtrip delay 736 milliseconds for method PRACK
00:02:13:sipSPIGetSdpBody :Parse incoming session description
00:02:13:CCSIP-SPI-CONTROL: sipSPIUACSessionTimer
00:02:13:CCSIP-SPI-CONTROL: act_recdproc_continue_200_processing
00:02:13:CCSIP-SPI-CONTROL: act_recdproc_continue_200_processing:*** This ccb is the parent
00:02:13:sipSPIDoFaxMediaNegotiation()
00:02:13:sipSPIDoMediaNegotiation:Codec (g711ulaw) Negotiation Successful on Static Payload
00:02:13: sipSPIDoPtimeNegotiation:One ptime attribute found - value:20
00:02:13: convert_ptime_to_codec_bytes:Values :Codec:g711ulaw ptime :20, codecbytes:160
00:02:13: convert_codec_bytes_to_ptime:Values :Codec:g711ulaw codecbytes :160, ptime:20
00:02:13: Parsed the direction:role identified as:0
00:02:13:sipSPIDoDTMFRelayNegotiation:Requested DTMF-RELAY option(s) not found in Preferred
 DTMF-RELAY option list!
00:02:13: sipSPIDoMediaNegotiation:DTMF Relay mode :Inband Voice
00:02:13:sip_sdp_get_modem_relay_cap_params:
00:02:13:sip_sdp_get_modem_relay_cap_params:NSE payload from X-cap = 0
00:02:13:sip_do_nse_negotiation:NSE Payload 100 found in SDP
00:02:13:sip_do_nse_negotiation:Remote NSE payload = local one = 100, Use it
00:02:13:sip_select_modem_relay_params:X-tmr not present in SDP. Disable modem relay
00:02:13: sipSPICompareSDP:Flags set:NEW_MEDIA :0 DSPDNLD REQD:0
00:02:13:sipSPIUpdCcbWithSdpInfo Bridge was done and there are no fqdn queries in progress,
 do RTCP updates
00:02:13:sipSPIRtcpUpdates:rtcp_session info
                laddr = 10.15.66.43, lport = 16398, raddr = 172.18.200.237, rport=16898
00:02:13:sipSPIRtcpUpdates:NO extraction of source address from remote media
00:02:13: sipSPIRtcpUpdates rtp session already created in bridge - update
00:02:13:sipSPIUpdCcbWithSdpInfo:SDP Media Information:
Negotiated Codec      :g711ulaw , bytes :160
Early Media           :0
Delayed Media         :0
Bridge Done           :1048576
New Media             :0
DSP DNLD Reqd         :0
```

```
Media Dest addr/Port  :172.18.200.237:16898
Orig Media Addr/Port  :0.0.0.0:0
00:02:13:sipSPIProcessMediaChanges
00:02:13:ccsip_process_response_contact_record_route
00:02:13:CCSIP-SPI-CONTROL: sipSPIProcess200OKforinvite
00:02:13:RequestCloseConnection:Closing connid 1 Local Port 50689
00:02:13:Queued event from SIP SPI :SIPSPI_EV_CLOSE_CONNECTION (8)
00:02:13:Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE (7)
00:02:13:sip_stats_method
00:02:13:0x6327E424 :State change from (STATE_RECD_PROCEEDING, SUBSTATE_PROCEEDING_PROGRESS)
  to (STATE_ACTIVE, SUBSTATE_NONE)
00:02:13:The Call Setup Information is :
Call Control Block (CCB) :0x6327E424
State of The Call        :STATE_ACTIVE
TCP Sockets Used         :NO
Calling Number           :888001
Called Number            :2021010124
Negotiated Codec         :g711ulaw
Negotiated Codec Bytes   :160
Negotiated Dtmf-relay    :0
Dtmf-relay Payload       :0
00:02:13:
Source IP Address (Sig  ):10.15.66.43
Source IP Address (Media):10.15.66.43
Source IP Port    (Media):16398
Destn  IP Address (Media):172.18.200.237
Destn  IP Port    (Media):16898
Destn SIP Req Addr:Port  :172.18.200.237:5060
Destn SIP Resp Addr:Port :0.0.0.0:0
Destination Name         :172.18.200.237
00:02:13:
 Orig Destn IP Address:Port (Media):0.0.0.0:0
00:02:13:udpsock_close_connect:Socket fd:1 closed for connid 1 with remote port:5060
00:02:13:Sent:
ACK sip:2021010124@172.18.200.237:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Mon, 01 Mar 1993 00:02:12 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Max-Forwards:1
Content-Length:0
CSeq:101 ACK
00:02:13:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Tue, 04 Jan 2000 23:57:54 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Server:Cisco-SIPGateway/IOS-12.x
CSeq:102 PRACK
Content-Length:0
00:02:13:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:172.18.200.237:5060
00:02:13:CCSIP-SPI-CONTROL: act_active_new_message
00:02:13:CCSIP-SPI-CONTROL: sact_active_new_message_response
00:02:13:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:02:27:Queued event from SIP SPI :SIPSPI_EV_CC_CALL_DISCONNECT (15)
00:02:27:CCSIP-SPI-CONTROL: act_active_disconnect
00:02:27:RequestCloseConnection:Closing connid 2 Local Port 50689
00:02:27:Queued event from SIP SPI :SIPSPI_EV_CLOSE_CONNECTION (8)
00:02:27:Queued event from SIP SPI :SIPSPI_EV_CREATE_CONNECTION (6)
00:02:27:0x6327E424 :State change from (STATE_ACTIVE, SUBSTATE_NONE)  to (STATE_ACTIVE,
SUBSTATE_CONNECTING)
```

```
00:02:27:0x6327E424 :State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)  to (STATE_ACTIVE,
 SUBSTATE_CONNECTING)
00:02:27:udpsock_close_connect:Socket fd:2 closed for connid 2 with remote port:5060
00:02:27:CCSIP-SPI-CONTROL: sipSPICheckSocketConnection:Connid(1) created to
172.18.200.237:5060, local_port 54607
00:02:27:0x6327E424 :State change from (STATE_ACTIVE, SUBSTATE_CONNECTING)  to (STATE_ACTIVE,
 SUBSTATE_NONE)
00:02:27:CCSIP-SPI-CONTROL: act_active_connection_created Call Disconnect - Sending Bye
00:02:27:Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE (7)
00:02:27:sip_stats_method
00:02:27:0x6327E424 :State change from (STATE_ACTIVE, SUBSTATE_NONE)  to (STATE_DISCONNECTING,
 SUBSTATE_NONE)
00:02:27:Sent:
BYE sip:2021010124@172.18.200.237:5060;user=phone SIP/2.0
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Mon, 01 Mar 1993 00:02:12 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
User-Agent:Cisco-SIPGateway/IOS-12.x
Max-Forwards:1
Timestamp:730944147
CSeq:103 BYE
Content-Length:0
00:02:27:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP  10.15.66.43:5060
From:"888001" <sip:888001@10.15.66.43>;tag=20694-C53
To:<sip:2021010124@172.18.200.237;user=phone>;tag=1069B954-25F
Date:Tue, 04 Jan 2000 23:58:08 GMT
Call-ID:D6EB9E87-14EE11CC-8008A04C-8E05D30C@10.15.66.43
Server:Cisco-SIPGateway/IOS-12.x
Timestamp:730944147
Content-Length:0
CSeq:103 BYE
00:02:27:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:172.18.200.237:5060
00:02:27:CCSIP-SPI-CONTROL: act_disconnecting_new_message
00:02:27:CCSIP-SPI-CONTROL: sact_disconnecting_new_message_response
00:02:27:CCSIP-SPI-CONTROL: sipSPICheckResponse
00:02:27:sip_stats_status_code
00:02:27: Roundtrip delay 16 milliseconds for method BYE
00:02:27:CCSIP-SPI-CONTROL: sipSPICallCleanup
00:02:27:sipSPIIcpifUpdate :CallState:4 Playout:0 DiscTime:14742 ConnTime 13360
00:02:27:0x6327E424 :State change from (STATE_DISCONNECTING, SUBSTATE_NONE)  to (STATE_DEAD,
 SUBSTATE_NONE)
00:02:27:The Call Setup Information is :
Call Control Block (CCB) :0x6327E424
State of The Call        :STATE_DEAD
TCP Sockets Used         :NO
Calling Number           :888001
Called Number            :2021010124
Negotiated Codec         :g711ulaw
Negotiated Codec Bytes   :160
Negotiated Dtmf-relay    :0
Dtmf-relay Payload       :0
00:02:27:
Source IP Address (Sig  ):10.15.66.43
Source IP Address (Media):10.15.66.43
Source IP Port    (Media):16398
Destn  IP Address (Media):172.18.200.237
Destn  IP Port    (Media):16898
Destn SIP Req Addr:Port  :172.18.200.237:5060
Destn SIP Resp Addr:Port :0.0.0.0:0
Destination Name         :172.18.200.237
```

```
00:02:27:
 Orig Destn IP Address:Port (Media):0.0.0.0:0
00:02:27:
 Disconnect Cause (CC)    :16
Disconnect Cause (SIP)   :200
00:02:27:****Deleting from UAC table
00:02:27:Removing call id 3
00:02:27:RequestCloseConnection:Closing connid 1 Local Port 54607
00:02:27:Queued event from SIP SPI :SIPSPI_EV_CLOSE_CONNECTION (8)
00:02:27: freeing ccb 6327E424
00:02:27:udpsock_close_connect:Socket fd:1 closed for connid 1 with remote port:5060
```

Configuration Examples for SIP Connection-Oriented Media Forking and MLPP Features

Connection-Oriented Media Enhancements for SIP Example

```
Router# show running-config
Building configuration...
Current configuration :2791 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
voice-card 2
!
ip subnet-zero
!
no ip domain lookup
ip domain name example.com
ip name-server 172.18.195.113
!
isdn switch-type primary-ni
!
fax interface-type fax-mail
mta receive maximum-recipients 0
ccm-manager mgcp
!
controller T1 2/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2/1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0/0
 ip address 172.18.197.22 255.255.255.0
 half-duplex
!
interface Serial0/0
```

```
 no ip address
 shutdown
!
interface TokenRing0/0
 no ip address
 shutdown
 ring-speed 16
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial2/0:23
 no ip address
 no logging event link-status
 isdn switch-type primary-ni
 isdn incoming-voice voice
 isdn outgoing display-ie
 no cdp enable
!
interface Serial2/1:23
 no ip address
 no logging event link-status
 isdn switch-type primary-ni
 isdn incoming-voice voice
 isdn outgoing display-ie
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
no ip http server
ip pim bidir-enable
!
call rsvp-sync
!
voice-port 2/0:23
!
voice-port 2/1:23
!
voice-port 3/0/0
!
voice-port 3/0/1
!
mgcp ip qos dscp cs5 media
mgcp ip qos dscp cs3 signaling
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 646 voip
destination-pattern 5552222
 session protocol sipv2
 session target ipv4:10.0.0.1
!
dial-peer voice 700 pots
 destination-pattern 700#T
port 0:D
!
gateway
!
sip-ua
```

```
nat symmetric check-media-src
max-forwards 5
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

SIP Multilevel Precedence and Priority Support Example

The following shows the result when the SIP: Multilevel Precedence and Priority Support feature is configured:

```
Router# show running-config
Building configuration...
Current configuration:2964 bytes
!
version 12.3
no parser cache
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname router1
!
boot-start-marker
boot-end-marker
!
logging buffered 1000000 debugging
!
!
dial-peer voice 9876 voip
 destination-pattern 9876
 voice-class sip resource priority namespace drsn
 voice-class sip resource priority mode strict
 session protocol sipv2
 session target ipv4:172.18.194.183
 session transport udp
!
dial-peer voice 222 pots
 incoming called-number
 direct-inward-dial
!
dial-peer voice 333 pots
 shutdown
 destination-pattern 9876
 prefix 9876
!
sip-ua
 retry invite 1
 retry bye 4
 retry cancel 4
 retry prack 4
 retry notify 4
retry refer 4
 retry info 4
 sip-server ipv4:172.19.194.186
 reason-header override
!
line con 0
 exec-timeout 0 0
```

```
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
!
end
```

SIP Support for Media Forking Examples

This section provides the following configuration and trace examples:

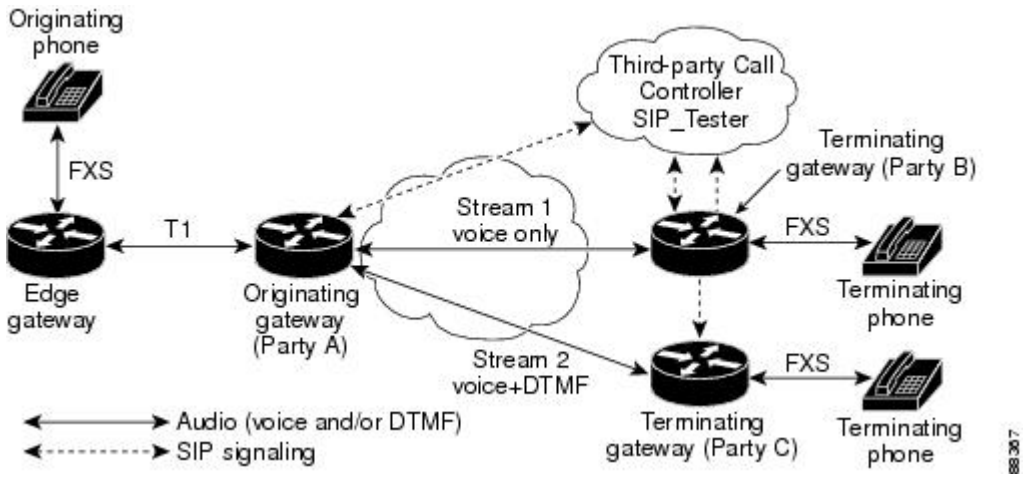
Note IP addresses and hostnames in these examples are fictitious.

SIP Network Using Media Forking

This configuration example shows a sample SIP network that uses media forking. The figure below shows a sample network where Party A dials Party B (555-2201). The dial peer for Party B on the originating gateway points to the IP address of SIP_Tester, which is acting as the third-party call controller. The Invite message is sent to SIP_Tester, who then forwards it to Party B. The typical SIP protocol exchange takes place to set up the first stream of the call. The user information portion of the SIP URL for SIP_Tester is 9999, so the dial peers on Party B and Party C are configured with 9999.

SIP_Tester initiates the establishment of the second stream by sending an initial Invite message with no SDP to Party C. Party C rings the terminating phone and responds to SIP_Tester with cause code 183 and an SDP that advertises its media capability. When the terminating phone answers, Party C responds to SIP_Tester with a 200 OK. SIP_Tester creates a re-Invite message with two media lines (m-lines) and sends it to Party A, who creates the second stream to Party C. Party A responds with an ACK that contains its local media information in the SDP. SIP_Tester forwards the ACK with SDP to Party C. A forked call is established.

Figure 73: Sample SIP Network Using Media Forking



Edge Gateway

The edge gateway configuration is used to convert a foreign-exchange-station (FXS) interface to a T1 interface. It is not involved in media forking or VoIP.

```
Router# show running-config
Building configuration...
Current configuration : 4455 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
logging rate-limit console 10 except errors
!
voice-card 1
!
ip subnet-zero
!
ip domain-name example.com
ip name-server 172.26.11.21
!
no ip dhcp-client network-discovery
isdn switch-type primary-dms100
isdn voice-call-failure 0
call rsvp-sync
!
controller T1 1/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1/1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Serial1/0:23
 no ip address
 no logging event link-status
 isdn switch-type primary-dms100
 isdn incoming-voice voice
 isdn T310 4000
 no cdp enable
!
interface Serial1/1:23
 no ip address
 no logging event link-status
 isdn switch-type primary-dms100
 isdn incoming-voice voice
 no fair-queue
 no cdp enable
!
interface FastEthernet3/0
 ip address 172.18.193.136 255.255.0.0
 duplex auto
 speed auto
!
ip classless
ip route 172.16.0.0 255.0.0.0 FastEthernet3/0
no ip http server
```

```
!
snmp-server packetsize 4096
snmp-server manager
!
voice-port 1/0:23
!
voice-port 1/1:23
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
dial-peer cor custom
!
dial-peer voice 5552 pots
 destination-pattern 5552...
 port 1/1:23
 prefix 5552
!
dial-peer voice 5555 pots
 destination-pattern 5555101
 port 2/0/1
!
line con 0
 exec-timeout 0 0
 transport preferred none
line aux 0
line vty 0 4
 exec-timeout 0 0
 password password1
 login
!
end
```

Party A

```
Router# show running-config
Building configuration...
Current configuration : 1864 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
memory-size iomem 10
voice-card 1
 codec complexity high
!
ip subnet-zero
!
ip domain-name example.com
ip name-server 172.26.11.21
!
isdn switch-type primary-dms100
isdn voice-call-failure 0
!
voice service voip
 sip
```

```
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
controller T1 1/0
 framing esf
 linecode b8zs
!
controller T1 1/1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0/0
 ip address 172.18.193.14 255.255.0.0
 half-duplex
 fair-queue 64 256 235
 ip rsvp bandwidth 7500 7500
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
interface Serial1/1:23
 no ip address
 no logging event link-status
 isdn switch-type primary-dms100
 isdn incoming-voice voice
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.193.1
no ip http server
!!
call rsvp-sync
!
voice-port 1/1:23
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 2100 voip
 destination-pattern 55521..
 session target ipv4:172.18.193.88
!
dial-peer voice 2200 voip
 destination-pattern 55522..
 session protocol sipv2
 session target ipv4:172.18.207.18:5062
 dtmf-relay rtp-nte
 codec g711ulaw
!
dial-peer voice 9999 voip
 destination-pattern 9999
 session protocol sipv2
 session target ipv4:172.18.207.18:5062
!
dial-peer voice 5557 pots
 destination-pattern 55571..
```

```
 direct-inward-dial
 port 1/1:23
!
sip-ua
 retry invite 3
 retry response 3
 retry bye 3
 retry cancel 3
 timers trying 501
!
line con 0
 exec-timeout 0 0
 transport preferred none
line aux 0
line vty 0 4
 password password1
 login
line vty 5 15
 login
!
no scheduler allocate
!
end
```

Party B

```
Router# show running-config
Building configuration...
Current configuration : 1769 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
memory-size iomem 10
clock timezone gmt 1
ip subnet-zero
!
ip domain-name example.com
ip name-server 172.26.11.21
!
interface FastEthernet0/0
 ip address 172.18.193.88 255.255.0.0
 no ip mroute-cache
 duplex auto
 speed auto
 fair-queue 64 256 235
 no cdp enable
 ip rsvp bandwidth 7500 7500
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.193.1
no ip http server
!
snmp-server engineID local 00000009020000107BDC8FA0
snmp-server community public RO
snmp-server packetsize 2048
call rsvp-sync
!
voice-port 1/0/0
 no supervisory disconnect lcfo
!
```

```
voice-port 1/0/1
 no supervisory disconnect lcfo
!
dial-peer cor custom
!
dial-peer voice 2100 pots
 destination-pattern 5552100
 port 1/0/0
!
dial-peer voice 2101 pots
 destination-pattern 5552101
 port 1/0/1
!
dial-peer voice 2200 pots
 destination-pattern 5552200
 port 1/0/0
!
dial-peer voice 2201 pots
 destination-pattern 5552201
 port 1/0/1
!
dial-peer voice 9999 voip
 destination-pattern 9999
 session protocol sipv2
 session target ipv4:172.18.207.18:5062
 codec g711ulaw
!
sip-ua
 retry invite 3
 retry response 3
 retry bye 3
 retry cancel 3
 timers trying 501
!
line con 0
 exec-timeout 0 0
 transport preferred none
line aux 0
line vty 0 4
 password password1
 login
line vty 5 15
 login
!
end
```

Party C

```
Router# show running-config
Building configuration...
Current configuration : 1638 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
memory-size iomem 10
ip subnet-zero
!
ip domain-name example.com
ip name-server 172.26.11.21
!
```

```
interface Ethernet0/0
 ip address 172.18.193.80 255.255.0.0
 half-duplex
 fair-queue 64 256 235
 ip rsvp bandwidth 7500 7500
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.193.1
no ip http server
!
call rsvp-sync
!
voice-port 1/0/0
 no supervisory disconnect lcfo
!
voice-port 1/0/1
 no supervisory disconnect lcfo
dial-peer cor custom
!
dial-peer voice 3100 pots
 destination-pattern 5553100
 port 1/0/0
!
dial-peer voice 3101 pots
 destination-pattern 5553101
 port 1/0/1
!
dial-peer voice 3200 pots
 destination-pattern 5553200
 port 1/0/0
!
dial-peer voice 3201 pots
 destination-pattern 5553201
 port 1/0/1
!
dial-peer voice 9999 voip
 destination-pattern 9999
 session protocol sipv2
 session target ipv4:172.18.207.18:5062
 dtmf-relay rtp-nte
 codec g711ulaw
!
sip-ua
 retry invite 3
 retry response 3
 retry bye 3
 retry cancel 3
!
line con 0
 exec-timeout 0 0
 transport preferred none
line aux 0
line vty 0 4
 exec-timeout 0 0
 password password1
 login
 transport input none
 escape-character BREAK
line vty 5 15
```

```
 login
!
end
```

Party A Initial-Call-Setup Trace

The following is the initial call-setup trace for Party A.

```
Router# debug ccsip message
*Mar  1 00:32:02.431: Sent:
INVITE sip:5552201@172.18.207.18:5062;user=phone SIP/2.0
Via: SIP/2.0/UDP  172.18.193.14:5060
From: "5555101" <sip:5555101@172.18.193.14>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>
Date: Mon, 01 Mar 1993 00:32:02 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
Supported: timer,100rel
Min-SE:  1800
Cisco-Guid: 27401485-351474124-2149117103-281843893
User-Agent: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 730945922
Contact: <sip:5555101@172.18.193.14:5060;user=phone>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 299
v=0
o=CiscoSystemsSIP-GW-UserAgent 2763 7166 IN IP4 172.18.193.14
s=SIP Call
c=IN IP4 172.18.193.14
t=0 0
m=audio 16412 RTP/AVP 0 100 101
c=IN IP4 172.18.193.14
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
*Mar  1 00:32:02.499: Received:
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "5555101" <sip:5555101@172.18.193.14;user=phone>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
CSeq: 101 INVITE
Require: 100rel
RSeq: 5413
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Type: application/sdp
Content-Length: 223
v=0
o=SIP_Tester 1239625037 1770029373 IN IP4 172.18.207.18
s=SIP Prot Test Call
t=0 0
m=audio 17236 RTP/AVP 0 100
c=IN IP4 172.18.193.88
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
```

```
a=ptime:20
*Mar  1 00:32:02.539: Sent:
PRACK sip:9999@172.18.207.18:5062;user=phone SIP/2.0
Via: SIP/2.0/UDP  172.18.193.14:5060
From: "5555101" <sip:5555101@172.18.193.14>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
Date: Mon, 01 Mar 1993 00:32:02 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
CSeq: 102 PRACK
RAck: 5413 101 INVITE
Content-Length: 0
*Mar  1 00:32:02.563: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "5555101" <sip:5555101@172.18.193.14;user=phone>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
CSeq: 102 PRACK
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Length: 0
*Mar  1 00:32:03.609: Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "5555101" <sip:5555101@172.18.193.14;user=phone>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
CSeq: 101 INVITE
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Type: application/sdp
Content-Length: 223
v=0
o=SIP_Tester 1239625037 1770029374 IN IP4 172.18.207.18
s=SIP Prot Test Call
t=0 0
m=audio 17236 RTP/AVP 0 100
c=IN IP4 172.18.193.88
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
*Mar  1 00:32:03.633: Sent:
ACK sip:9999@172.18.207.18:5062;user=phone SIP/2.0
Via: SIP/2.0/UDP  172.18.193.14:5060
From: "5555101" <sip:5555101@172.18.193.14>;tag=1D556B-24F1
To: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
Date: Mon, 01 Mar 1993 00:32:02 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
Max-Forwards: 6
Content-Length: 0
CSeq: 101 ACK
```

Party B Initial-Call-Setup Trace

The following is the initial call-setup trace for Party B. Also, call status is displayed with the **show sip-ua calls** command.

```
Router# debug ccsip message
*Mar  1 00:43:13.655: Received:
INVITE sip:5552201@172.18.193.88;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
```

```
To: "5552201" <sip:5552201@172.18.193.88;user=phone>
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 487666621@172.18.207.18
Supported: 100rel
CSeq: 101 INVITE
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Type: application/sdp
Content-Length: 278
v=0
o=SIP_Tester 1818337819 831652457 IN IP4 172.18.207.18
s=SIP Prot Test Call
t=0 0
m=audio 16452 RTP/AVP 0 100 101
c=IN IP4 172.18.193.14
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
*Mar  1 00:43:13.683: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 00:43:13 gmt
Call-ID: 487666621@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow-Events: telephone-event
Content-Length: 0
*Mar  1 00:43:13.715: Sent:
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 00:43:13 gmt
Call-ID: 487666621@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Require: 100rel
RSeq: 5450
Allow-Events: telephone-event
Contact: <sip:5552201@172.18.193.88:5060;user=phone>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 243
v=0
o=CiscoSystemsSIP-GW-UserAgent 1886 7999 IN IP4 172.18.193.88
s=SIP Call
c=IN IP4 172.18.193.88
t=0 0
m=audio 17936 RTP/AVP 0 100
c=IN IP4 172.18.193.88
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
*Mar  1 00:43:13.779: Received:
PRACK sip:5552201@172.18.193.88;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 01:01:01 GMT
```

```
Call-ID: 487666621@172.18.207.18
CSeq: 102 PRACK
RAck: 0 101 INVITE
Content-Length: 0
*Mar  1 00:43:13.791: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 00:43:13 gmt
Call-ID: 487666621@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 102 PRACK
Content-Length: 0
*Mar  1 00:43:17.251: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 00:43:13 gmt
Call-ID: 487666621@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow-Events: telephone-event
Contact: <sip:5552201@172.18.193.88:5060;user=phone>
Content-Type: application/sdp
Content-Length: 243
v=0
o=CiscoSystemsSIP-GW-UserAgent 1886 7999 IN IP4 172.18.193.88
s=SIP Call
c=IN IP4 172.18.193.88
t=0 0
m=audio 17936 RTP/AVP 0 100
c=IN IP4 172.18.193.88
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
*Mar  1 00:43:17.343: Received:
ACK sip:5552201@172.18.193.88;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5552201" <sip:5552201@172.18.193.88;user=phone>;tag=279390-CB
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 487666621@172.18.207.18
CSeq: 101 ACK
Content-Length: 0
Router# show sip-ua calls
SIP UAC CALL INFO
   Number of UAC calls: 0
SIP UAS CALL INFO
Call 1
SIP Call ID                : 487666621@172.18.207.18
   State of the call       : STATE_ACTIVE (6)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 9999
   Called Number           : 5552201
   Bit Flags               : 0x1212003A 0x20000
   Source IP Address (Sig ): 172.18.193.88
   Destn SIP Req Addr:Port : 172.18.207.18:5062
   Destn SIP Resp Addr:Port: 172.18.207.18:5062
   Destination Name        : 172.18.207.18
   Number of Media Streams : 1
   Number of Active Streams: 1
```

```
RTP Fork Object         : 0x0
Media Stream 1
 State of the stream : STREAM_ACTIVE (5)
 Stream Call ID : 9
 Stream Type : voice-only (0)
 Negotiated Codec : g711ulaw (160 bytes)
 Codec Payload Type : 0
 Negotiated Dtmf-relay : inband-voice (0)
 Dtmf-relay Payload : 0
 Media Source IP Addr:Port: 172.18.193.88:17936
 Media Dest IP Addr:Port : 172.18.193.14:16452
Number of UAS calls: 1
```

Party A Add-Second-Stream Trace

The following is the second stream trace added by Party A. Also, call status is displayed with the **show sip-ua calls** command.

```
Router# debug ccsip message
*Mar  1 00:33:05.178: Received:
INVITE sip:5555101@172.18.193.14;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
To: "5555101" <sip:5555101@172.18.193.14;user=phone>;tag=1D556B-24F1
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
Supported: 100rel
CSeq: 101 INVITE
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Type: application/sdp
Content-Length: 635
v=0
o=SIP_Tester 1239625037 1770029375 IN IP4 172.18.207.18
s=SIP Prot Test Call
t=0 0
m=audio 17236 RTP/AVP 0 100
c=IN IP4 172.18.193.88
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
m=audio 17114 RTP/AVP 0 100 101 101 100
c=IN IP4 172.18.193.80
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
*Mar  1 00:33:05.222: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
To: "5555101" <sip:5555101@172.18.193.14>;tag=1D556B-24F1
```

```
Date: Mon, 01 Mar 1993 00:33:05 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow-Events: telephone-event
Contact: <sip:5555101@172.18.193.14:5060;user=phone>
Content-Type: application/sdp
Content-Length: 431
v=0
o=CiscoSystemsSIP-GW-UserAgent 2763 7167 IN IP4 172.18.193.14
s=SIP Call
c=IN IP4 172.18.193.14
t=0 0
m=audio 16412 RTP/AVP 0 100
c=IN IP4 172.18.193.14
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
m=audio 18802 RTP/AVP 0 101 100
c=IN IP4 172.18.193.14
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
*Mar  1 00:33:05.234: Received:
ACK sip:5555101@172.18.193.14;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: <sip:5552201@172.18.207.18;user=phone>;tag=tester-tag
To: "5555101" <sip:5555101@172.18.193.14;user=phone>;tag=1D556B-24F1
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
CSeq: 101 ACK
Content-Length: 0
Router# show sip-ua calls
SIP UAC CALL INFO
Call 1
SIP Call ID                : 1A3F2B6-14F311CC-801AECAF-10CC98B5@172.18.193.14
   State of the call       : STATE_ACTIVE (6)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 5555101
   Called Number           : 5552201
   Bit Flags               : 0x12120030 0x20000
   Source IP Address (Sig ): 172.18.193.14
   Destn SIP Req Addr:Port : 172.18.207.18:5062
   Destn SIP Resp Addr:Port: 172.18.207.18:5062
   Destination Name        : 172.18.207.18
   Number of Media Streams : 2
   Number of Active Streams: 2
   RTP Fork Object         : 0x83064DC8
   Media Stream 1
    State of the stream : STREAM_ACTIVE (5)
    Stream Call ID : 11
    Stream Type : voice-only (0)
    Negotiated Codec : g711ulaw (160 bytes)
    Codec Payload Type : 0
    Negotiated Dtmf-relay : inband-voice (0)
    Dtmf-relay Payload : 0
    Media Source IP Addr:Port: 172.18.193.14:16412
    Media Dest IP Addr:Port : 172.18.193.88:17236
   Media Stream 2
    State of the stream : STREAM_ACTIVE (5)
```

```
    Stream Call ID : 12
    Stream Type : voice+dtmf (1)
    Negotiated Codec : g711ulaw (160 bytes)
    Codec Payload Type : 0
    Negotiated Dtmf-relay : rtp-nte (6)
    Dtmf-relay Payload : 101
    Media Source IP Addr:Port: 172.18.193.14:18802
    Media Dest IP Addr:Port : 172.18.193.80:17114
   Number of UAC calls: 1
SIP UAS CALL INFO
   Number of UAS calls: 0
```

Party C Add-Second-Stream Trace

The following is the second stream trace added by Party C. Also, call status is displayed with the **show sip-ua calls** command.

```
Router# debug ccsip message
*Mar  1 00:44:19.763: Received:
INVITE sip:5553201@172.18.193.80;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5553201" <sip:5553201@172.18.193.80;user=phone>
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 2108310431@172.18.207.18
Supported: 100rel
CSeq: 101 INVITE
Contact: <sip:9999@172.18.207.18:5062;user=phone>
Content-Length: 0
*Mar  1 00:44:19.792: Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5553201" <sip:5553201@172.18.193.80;user=phone>;tag=2895C8-53B
Date: Mon, 01 Mar 1993 00:44:19 GMT
Call-ID: 2108310431@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow-Events: telephone-event
Content-Length: 0
*Mar  1 00:44:19.828: Sent:
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5553201" <sip:5553201@172.18.193.80;user=phone>;tag=2895C8-53B
Date: Mon, 01 Mar 1993 00:44:19 GMT
Call-ID: 2108310431@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Require: 100rel
RSeq: 6083
Allow-Events: telephone-event
Contact: <sip:5553201@172.18.193.80:5060;user=phone>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 523
v=0
o=CiscoSystemsSIP-GW-UserAgent 8259 5683 IN IP4 172.18.193.80
s=SIP Call
c=IN IP4 172.18.193.80
t=0 0
m=audio 18988 RTP/AVP 0 100 101
c=IN IP4 172.18.193.80
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
*Mar  1 00:44:20.985: Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5553201" <sip:5553201@172.18.193.80;user=phone>;tag=2895C8-53B
Date: Mon, 01 Mar 1993 00:44:19 GMT
Call-ID: 2108310431@172.18.207.18
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow-Events: telephone-event
Contact: <sip:5553201@172.18.193.80:5060;user=phone>
Content-Type: application/sdp
Content-Length: 523
v=0
o=CiscoSystemsSIP-GW-UserAgent 8259 5683 IN IP4 172.18.193.80
s=SIP Call
c=IN IP4 172.18.193.80
t=0 0
m=audio 18988 RTP/AVP 0 100 101
c=IN IP4 172.18.193.80
a=rtpmap:0 PCMU/8000
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
*Mar  1 00:44:20.997: Received:
ACK sip:5553201@172.18.193.80;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.207.18:5062
From: "SIP_Tester" <sip:9999@172.18.207.18:5062;user=phone>;tag=tester-tag
To: "5553201" <sip:5553201@172.18.193.80;user=phone>;tag=2895C8-53B
Date: Mon, 01 Mar 1993 01:01:01 GMT
Call-ID: 2108310431@172.18.207.18
CSeq: 101 ACK
Content-Type: application/sdp
Content-Length: 277
v=0
o=SIP_Tester 2029259292 42666129 IN IP4 172.18.207.18
s=SIP Prot Test Call
t=0 0
m=audio 16728 RTP/AVP 0 101 100
c=IN IP4 172.18.193.14
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:100 X-NSE/8000
a=fmtp:100 192-194
a=ptime:20
Router# show sip-ua calls
SIP UAC CALL INFO
   Number of UAC calls: 0
SIP UAS CALL INFO
Call 1
SIP Call ID                : 186464186@172.18.207.18
   State of the call       : STATE_ACTIVE (6)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 9999
   Called Number           : 5553201
```

```
Bit Flags               : 0x1212003A 0x20000
Source IP Address (Sig ): 172.18.193.80
Destn SIP Req Addr:Port : 172.18.207.18:5062
Destn SIP Resp Addr:Port: 172.18.207.18:5062
Destination Name        : 172.18.207.18
Number of Media Streams : 1
Number of Active Streams: 1
RTP Fork Object         : 0x0
Media Stream 1
 State of the stream : STREAM_ACTIVE (5)
 Stream Call ID : 7
 Stream Type : voice+dtmf (1)
 Negotiated Codec : g711ulaw (160 bytes)
 Codec Payload Type : 0
 Negotiated Dtmf-relay : rtp-nte (6)
 Dtmf-relay Payload : 101
 Media Source IP Addr:Port: 172.18.193.80:19352
 Media Dest IP Addr:Port : 172.18.193.14:16770
Number of UAS calls: 1
```

Additional References

The following sections provide references related to the SIP Connection-Oriented Media, Forking, and MLPP features.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
SIP commands	*Cisco IOS Voice Command Reference*
Tcl IVR and VoiceXML	*Cisco IOS Tcl IVR and VoiceXML Application Guide*
Cisco VoiceXML	*Cisco VoiceXML Programmer's Guide*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

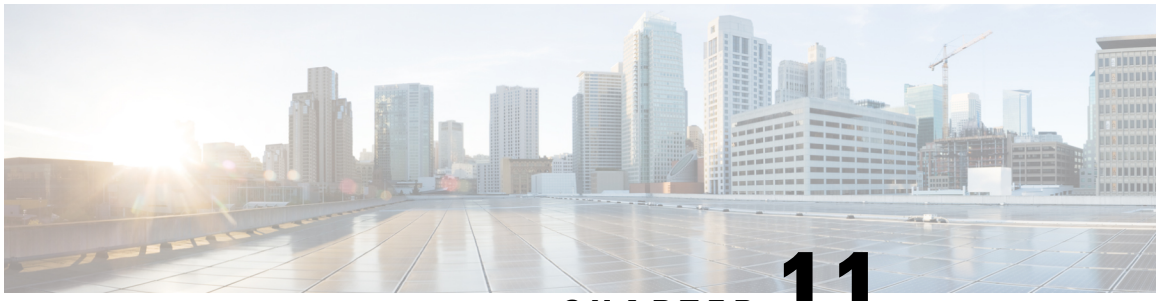
MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 11

Transparent Tunneling of QSIG and Q.931 over SIP TDM Gateway and SIP-SIP Cisco Unified Border Element

Transparent Tunneling of QSIG and Q.931 over SIP TDM Gateway and SIP-SIP Cisco Unified Border Element was introduced on Cisco IOS SIP gateways in phases. In the first phase, the Transparent Tunneling of QSIG over SIP TDM Gateway feature added the ability to transparently tunnel Q-signaling (QSIG) protocol ISDN messages across the Session Initiation Protocol (SIP) trunk. With this feature, QSIG messages (supplementary services carried within Q.931 FACILITY-based messages) can be passed end to end across a SIP network. However, in Cisco IOS Release 12.4(15)XY, deployment of this feature is limited to QSIG messages over SIP Time-Division Multiplexing (TDM) gateways. In later releases, the ISDN Q.931 Tunneling over SIP TDM Gateway feature adds support for transparent tunneling of all Q.931 messages over SIP and for the Transparent Tunneling of QSIG and Q.931 over a SIP-SIP Cisco Unified Border Element.

Transparent tunneling is accomplished by encapsulating QSIG or Q.931 messages within SIP message bodies. These messages are encapsulated using “application/qsig” or “application/x-q931” Multipurpose Internet Mail Extensions (MIME) to tunnel between SIP endpoints. Using MIME to tunnel through Cisco SIP messaging does not include any additional QSIG/Q.931 services to SIP interworking.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Transparent Tunneling of QSIG and Q.931 over SIP TDM Gateway and SIP-SIP Cisco Unified Border Element".

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Transparent Tunneling of QSIG or Q.931 over SIP

The Transparent Tunneling of QSIG over SIP TDM Gateway feature is intended for TDM PBX toll bypass and call center applications. In its first release (Cisco IOS Release 12.4(15)XY), only tunneling of QSIG messages is supported and only on TDM gateways. From Cisco IOS release 12.4(15)XZ and 12.4(20)T onward, support is added for the ISDN Q.931 Tunneling over SIP TDM Gateway and Transparent Tunneling of QSIG and Q.931 over SIP-SIP Cisco Unified Border Element.

Before configuring transparent tunneling of QSIG and Q.931 over a SIP trunk, verify the SIP configuration within the VoIP network for the appropriate originating and terminating gateways as described in documentation listed in "Prerequisites for Transparent Tunneling of QSIG or Q.931 over SIP".

Restrictions for Transparent Tunneling of QSIG or Q.931 over SIP

Transparent tunneling of QSIG or Q.931 does not function unless both the originating gateway (OGW) and the terminating gateway (TGW) are configured using the same ISDN switch type. Additionally, this function is supported only on SIP-to-SIP configurations on Cisco Unified Border Element. Tunneling of QSIG or Q.931 is not supported on SIP-to-H.323 or H.323-to-H.323 configurations on Cisco Unified Border Element.

Information About Transparent Tunneling of QSIG or Q.931 over SIP

To configure transparent tunneling of QSIG or Q.931 over SIP, you should understand the following concepts:

Use of the QSIG or Q.931 Protocols

Q-series documents, controlled by the International Telecommunication Union (ITU), define the network Layer. The Q.931 document defines the Layer 3 protocol that serves as the connection control protocol for ISDN signaling--it is used primarily to manage the initiation, maintenance, and termination of connections over a digital network.

The Q signaling (QSIG) protocol is based on the Q.931 standard and is used for ISDN communications in a Private Integrated Services Network (PISN). The QSIG protocol makes it possible to pass calls from one circuit switched network, such as a PBX or private integrated services network exchange (PINX), to another. QSIG messages are, essentially, a subset of Q.931 messages that ensure the essential Q.931 FACILITY-based functions successfully traverse the network regardless of the various hardware involved.

Q.931 tunneling over Cisco IOS SIP gateways was introduced as the ability to transparently tunnel only QSIG messages--the FACILITY-based Q.931 messages. Beginning with Cisco IOS Release 12.4(15)XZ and Cisco

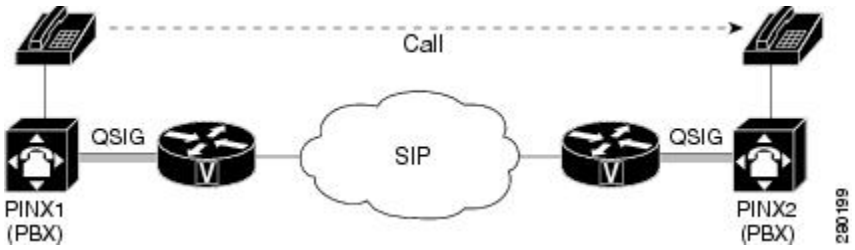
IOS Release 12.4(20)T, tunneling of all Q.931 messages (SETUP, ALERTING, CONNECT, and RELEASE COMPLETE messages in addition to FACILITY-based messages) is supported on Cisco IOS SIP gateways. However, for clarity, the descriptions and examples in this document focus primarily on QSIG messages.

Purpose of Tunneling QSIG or Q.931 over SIP

TDM Gateways

Transparently tunneling QSIG or Q.931 messages over SIP through SIP TDM gateways allows calls from one PINX to another to be passed through a SIP-based IP network with the equivalent functionality of passing through an H.323 network--without losing the functionality of the QSIG or Q.931 protocol to establish the call. To do this, QSIG or Q.931 messages are encapsulated within SIP messages (see the figure below).

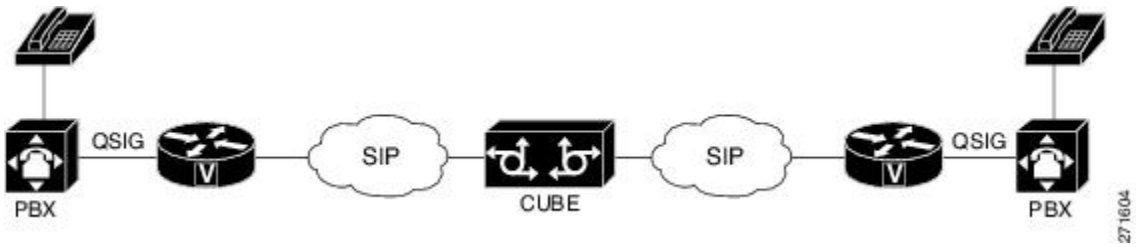
Figure 74: Tunneling QSIG (or Q.931) Messages Across a SIP Trunk



Cisco Unified Border Elements

Transparently tunneling QSIG or Q.931 over SIP through a Cisco Unified Border Element allows calls from one network to be passed through a SIP-to-SIP Cisco Unified Border Element connection to a bordering network (see the figure below).

Figure 75: Tunneling QSIG (or Q.931) Messages Through a SIP-SIP Cisco Unified Border Element



Encapsulation of QSIG in SIP Messaging

QSIG messages are tunneled by encapsulating them as a MIME body in a SIP INVITE message on the OGW. Then, the MIME body is extracted from the SIP message by the TGW at the other end of the SIP network. To tunnel QSIG messages to a TGW on another network, configure and use a SIP-to-SIP Cisco Unified Border Element connection between each network over which the SIP INVITE must travel to reach the TGW. This tunneling process helps preserve all QSIG capabilities associated with a call or call-independent signal as it travels to its destination.

The following events make it possible to tunnel QSIG messaging across a SIP network:

- The ingress gateway (OGW) receives a QSIG call (or signal) establishment request (a SETUP message) and generates a corresponding SIP INVITE request.
- A corresponding SIP INVITE message is created and will contain the following:
  - A Request-URI--message part containing a destination derived from the called party number information element (IE) in the QSIG SETUP message. The destination can be the egress (TGW or the Cisco Unified Border Element) for exiting the SIP network or it can be the required destination, leaving SIP proxies to determine which gateway will be used.
  - A From header--message header containing a uniform resource identifier (URI) for either the OGW or calling party itself.
  - A Session Description Protocol (SDP) offer--a message part proposing two media streams, one for each direction.
  - A Multipart-MIME body--message part containing the tunneled QSIG data.
- In addition to normal user agent (UA) handling of a SIP response, the OGW performs a corresponding action when it receives a SIP response, as follows:
  - OGW receives 18*x* response with tunneled content--identifies the QSIG message (FACILITY, ALERTING, or PROGRESS) and sends a corresponding ISDN message.
  - OGW receives 3*xx* , 4*xx* , 5*xx* , or 6*xx* final response--attempts alternative action to route the initial QSIG message or clears the call or signal using an appropriate QSIG cause value (DISCONNECT, RELEASE, or RELEASE COMPLETE). When the OGW receives a valid encapsulated QSIG RELEASE COMPLETE message, the OGW should use the cause value included in that QSIG message to determine the cause value.

Note You should expect a SIP 415 final response message (Unsupported Media Type) if the user agent server (UAS) is unable to process tunneled QSIG or Q.931 messages.

- • OGW receives a SIP 200 OK response--performs normal SIP processing, which includes sending an ACK message. Additionally, the OGW will encapsulate the QSIG message in the response to the PSTN side and will connect the QSIG user information channel to the appropriate media streams as called out in the SDP reply.

Note A nonzero port number for each media stream must be provided in a SIP 200 OK response to the OGW before the OGW receives the QSIG CONNECT message. Otherwise, the OGW will behave as if the QSIG T301 timer expired.

- The TGW sends and the OGW receives a 200 OK response--the OGW sends an ACK message to the TGW and all successive messages during the session are encapsulated into the body of SIP INFO request messages. There are two exceptions:
  - When a SIP connection requires an extended handshake process, renegotiation, or an update, the gateway may encapsulate a waiting QSIG message into a SIP re-INVITE or SIP UPDATE message during QSIG call establishment.
  - When the session is terminated, gateways send a SIP BYE message. If the session is terminated by notice of a QSIG RELEASE COMPLETE message, that message can be encapsulated into the SIP BYE message.

Mapping of QSIG Message Elements to SIP Message Elements

This section lists QSIG message elements and their associated SIP message elements when QSIG messages are tunneled over a SIP trunk.

• QSIG FACILITY/NOTIFY/INFO	<-->	SIP INFO
• QSIG SETUP	<-->	SIP INVITE
• QSIG ALERTING	<-->	SIP 180 RINGING
• QSIG PROGRESS	<-->	SIP 183 PROGRESS
• QSIG CONNECT	<-->	SIP 200 OK
• QSIG DISCONNECT	<-->	SIP BYE/CANCEL/4*xx* --6*xx* Response

How to Transparently Tunnel QSIG over SIP

To create a tunnel for QSIG messages across a SIP trunk, you must configure signaling forward settings on both the OGW and the TGW.

In the IP TDM gateway scenario, a gateway receives QSIG messages from PSTN and the ISDN module passes the raw QSIG message and, additionally, creates and includes a Generic Transparency Descriptor (GTD) that is passed with the raw QSIG message across the IP leg of the call.

In the SIP TDM gateway scenario, there are two options--raw message (rawmsg) and unconditional. The rawmsg option specifies tunneling of only raw message (application/qsig or application/x-q931). The unconditional option specifies tunneling of all additional message bodies, such as GTD and raw message (application/qsig or application/x-q931).

Use the **signaling forward**command at the global configuration level to configure the feature for the entire gateway. You can also enable the QSIG tunneling feature for only a specific interface. If you enable this feature at both the global and dial peer configuration level and the option specified for the interface is different than for the gateway, the interface setting will override the global setting. The processes for specifying either option at both levels are included in the following sections:

Configuring Signaling Forward Settings for a Gateway

To create a tunnel for QSIG messages across a SIP trunk using the same signaling forward setting for all interfaces on a gateway, configure the signaling forward settings in voice service voip configuration mode.

Signaling Forward Settings for a Gateway

The two options--raw messages (rawmsg) and unconditional--are mutually exclusive, which means you can specify only one option at the global configuration level. To enable and specify the signaling forward option, use the **signaling forward** command in voice service voip configuration mode.

Note To override the global setting for a specific interface, use the **signaling forward** command at the dial-peer level (see "Configuring Signaling Forward Settings for an Interface").

Before you begin

To create QSIG tunnels using the signaling forward configuration, configure both gateways. You can configure gateways globally or you can configure one or more interfaces on a gateway. In either case, you must include the recommended configuration for PRACK to avoid message/data loss.

Note It is not necessary that both gateways are configured with the same signaling forward option but, if they are not, only raw QSIG messages can be tunneled. However, it is recommended that you tunnel QSIG messages with at least one interface configured on both gateways. If only one gateway is configured, QSIG tunneling might work in one direction but may not work properly in both directions.

You must also specify the central office switch type on the ISDN interface for both the OGW and the TGW. Use the **isdn switch-type** command in global or dial peer configuration mode to enable and specify the switch type for QSIG or Q.931 support (see "Signaling Forward Settings for a Gateway").

Furthermore, before the **isdn switch-type** setting can function properly, you must assign network-side functionality for the primary-qsig switch type (either at the global or dial-peer level) using the **isdn protocol-emulate** command (see "Signaling Forward Settings for a Gateway").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. voice service voip
4. Do one of the following:
   - **signaling forward** *message-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	voice service voip **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode and specifies a voice-encapsulation type globally.
Step 4	Do one of the following: • **signaling forward** *message-type* **Example:** `Router(conf-voi-serv)# signaling forward rawmsg` **Example:** `Router(conf-voi-serv)# signaling forward unconditional`	Enables tunneling of QSIG raw messages (application-qsig) only. or Enables tunneling of all QSIG message bodies unconditionally.

Configuring Signaling Forward Settings for an Interface

To create a tunnel for QSIG messages across a SIP trunk on a specific interface on a gateway, configure the signaling forward settings in dial peer configuration mode.

Signaling Forward Settings for an Interface

The two options--raw messages (rawmsg) and unconditional--are mutually exclusive, which means you can specify only one option per interface at the dial-peer level. To enable and specify the signaling forward option for an interface, use the **signaling forward** command in dial peer configuration mode.

Note To set the signaling forward option for an entire gateway, use the **signaling forward** command at the global level (see "Configuring Signaling Forward Settings for a Gateway").

Before you begin

To create QSIG tunnels using the signaling forward configuration, configure at least one interface on both gateways. You can also configure all interfaces at once by configuring the gateway globally. In either case, you must include the recommended configuration for PRACK to avoid data loss.

Note It is not necessary that both gateways are configured with the same signaling forward option but, if they are not, only raw QSIG messages can be tunneled. However, it is recommended that you tunnel QSIG messages with at least one interface configured on both gateways. If only one gateway is configured, QSIG tunneling might work in one direction but may not work properly in both directions.

You must also specify the central office switch type on the ISDN interface for both the OGW and the TGW. Use the **isdn switch-type** command in global or dial peer configuration mode to enable and specify the switch type for QSIG or Q.931 support (see "Signaling Forward Settings for an Interface").

Furthermore, before the **isdn switch-type** setting can function properly, you must assign network-side functionality for the primary-qsig switch type (either at the global or dial-peer level) using the **isdn protocol-emulate** command (see "Signaling Forward Settings for an Interface").

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *number* **voip**
4. Do one of the following:
   - **signaling forward** *message-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *number* **voip** **Example:** `Router(config)# dial-peer voice 3 voip`	Enters voice-service configuration mode and specifies a voice-encapsulation type for a specific interface.
Step 4	Do one of the following: • **signaling forward** *message-type* **Example:** `Router(config-dial-peer)# signaling forward rawmsg` **Example:** `Router(config-dial-peer)# signaling forward unconditional`	Enables tunneling of QSIG raw messages (application-qsig) only. or Enables tunneling of all QSIG message bodies unconditionally.

Configuration Examples for Transparent Tunneling of QSIG over SIP

This section provides the following configuration examples:

- Configuration at the global level:
  - Tunneling QSIG Raw Messages over SIP on an OGW or TGW Example, on page 483
  - Tunneling QSIG Messages Unconditionally over SIP on an OGW or TGW Example, on page 483
- Configuration at the dial peer (interface) level:

Tunneling QSIG Raw Messages over SIP on an OGW or TGW Example

The following example shows how to configure transparent tunneling of only QSIG raw messages (application-qsig) through a SIP TDM gateway on a SIP trunk at either the OGW or TG:

```
!
voice service voip
 signaling forward rawmsg
 sip
  rel1xx require "100rel"
!
```

Tunneling QSIG Messages Unconditionally over SIP on an OGW or TGW Example

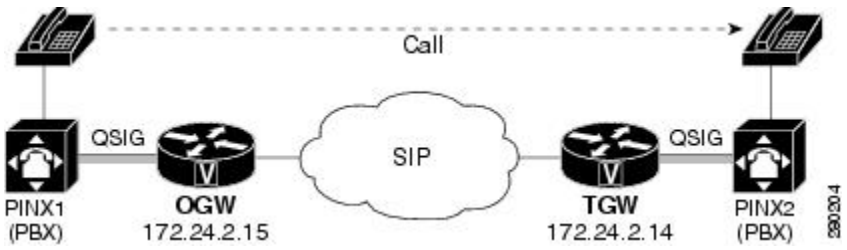
The following example shows how to configure transparent tunneling of QSIG messages unconditionally through a SIP TDM gateway on a SIP trunk at either the OGW or TGW:

```
!
voice service voip
 signaling forward unconditional
 sip
  rel1xx require "100rel"
!
```

Tunneling QSIG Raw Messages over SIP on an OGW and TGW Interface Example

The following example shows how to configure transparent tunneling of only QSIG raw messages (application-qsig) on a gateway interface in a SIP network (see the figure below):

Figure 76: Tunneling of Only QSIG Raw Messages over a SIP Trunk (Interface-Level)



Configuration for OGW (172.24.2.15) Tunneling only QSIG Raw Mmessages

```
!
dial-peer voice 7777 voip
description OGW-OUT-TGW
destination-pattern 222
signaling forward rawmsg
session protocol sipv2
session target ipv4:172.24.2.14
!
```

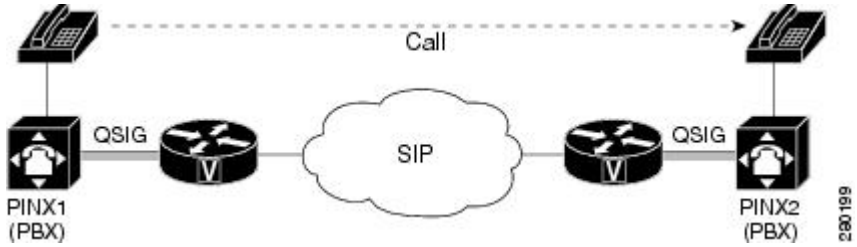
Configuration for TGW (172.24.2.14) Tunneling only QSIG Raw Mmessages

```
!
dial-peer voice 333 voip
description TGW_RSVP_IN-DP
session protocol sipv2
signaling forward rawmsg
incoming called-number 222
!
```

Tunneling QSIG Messages Unconditionally over SIP on an OGW or TGW Interface Example

The following example shows how to configure transparent tunneling of QSIG messages unconditionally over a gateway interface in a SIP network (see the figure below):

Figure 77: Tunneling of QSIG Messages Unconditionally over a SIP Trunk (Interface-Level)



Configuration for OGW (172.24.2.14) Tunneling QSIG Messages Unconditionally

```
dial-peer voice 7777 voip
 description OGW-OUT-TGW
 destination-pattern 222
 signaling forward unconditional
 session protocol sipv2
 session target ipv4:172.24.2.14
```

Configuration for TGW (172.24.2.15) Tunneling QSIG Messages Unconditionally

```
dial-peer voice 333 voip
 description TGW-RSVP-IN-DP
 session protocol sipv2
 signaling forward unconditional
 incoming called-number 222
```

Additional References

The following sections provide references related to the Transparent Tunneling of QSIG and Q.931 over SIP-TDM Gateway and SIP-SIP Cisco Unified Border Element features.

Related Documents

Related Topic	Document Title
Cisco IOS dial peer overview	"Dial Peer Configuration on Voice Gateway Routers Configuration Guide"
Cisco IOS dial technologies command information	*Cisco IOS Dial Technologies Command Reference*
Cisco IOS dial technologies configuration information	*Cisco IOS Dial Technologies Configuration Guide*
Cisco IOS SIP configuration information	*Cisco IOS SIP Configuration Guide*
Cisco IOS voice command information	*Cisco IOS Voice Command Reference*
Cisco IOS voice configuration information	*Cisco IOS Voice Configuration Library*
Cisco Unified CME command information	Cisco Unified Communications Manager Express Command Reference
Cisco Unified CME configuration information	Cisco Unified CME Support Documentation Home Page

Standards

Standard	Title
Standard ECMA-355	Corporate Telecommunication Networks - Tunnelling of QSIG over SIP

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download existing MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3204	MIME Media Types for ISUP and QSIG Objects
RFC 4497	Interworking Between the Session Initiation Protocol (SIP) and QSIG

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Transparent Tunneling of QSIG and Q.931 over SIP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for Transparent Tunneling of QSIG over SIP

Feature Name	Releases	Feature Information
Transparent Tunneling of QSIG over SIP-TDM Gateway	12.4(15)XY 12.4(20)T	This feature provides transparent tunneling of ISDN communications that use the QSIG protocol across an IP network. The QSIG tunneling feature does not add any QSIG services to SIP interworking. Additionally, for Cisco IOS Release 12.4(15)XY, the QSIG tunneling feature targets only time-division multiplexing (TDM) SIP gateways. This feature uses no new or modified commands.
ISDN Q.931 Tunneling over SIP TDM Gateway	12.4(15)XZ 12.4(20)T	This feature expands transparent tunneling of QSIG messages to all other Q.931 messages (SETUP, ALERTING, CONNECT, and RELEASE COMPLETE). The QSIG and Q.931 tunneling feature does not add any QSIG or Q.931 services to SIP interworking.
Transparent Tunneling of QSIG and Q.931 over SIP-SIP Cisco Unified Border Element	12.4(15)XZ 12.4(20)T	This feature extends support of QSIG and Q.931 tunneling to the Cisco Unified Border Element.

Glossary

ISDN--Integrated Services Digital Network.

MIME--Multipurpose Internet Mail Extensions.

OGW--originating gateway (ingress gateway).

PBX--Private Branch Exchange.

PINX--private integrated services network exchange.

PISN--private integrated services network.

QSIG--Q Signaling protocol.

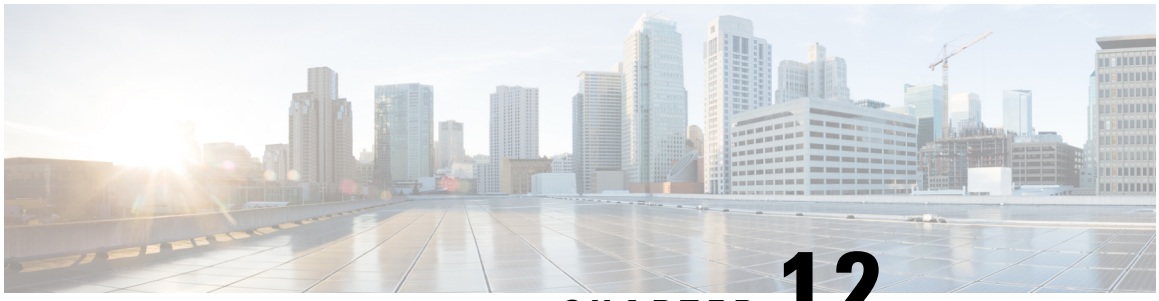
SDP--Session Description Protocol.

SIP--Session Initiation Protocol.

TDM--Time-Division Multiplexing.

TGW--terminating gateway (egress gateway).

URI--uniform resource identifier.



CHAPTER 12

PAI or PPI Header in Incoming and Outgoing SIP Calls

Prior to the introduction of the PAI or PPI Header in Incoming and Outgoing SIP Calls feature, the P-Asserted-Identity (PAI) or the P-Preferred-Identity (PPI) privacy header was supported for outgoing calls at the global level. The PAI or PPI Header in Incoming and Outgoing SIP Calls feature is an enhancement to support PAI or PPI header for incoming and outgoing calls at the global level and dial-peer configuration mode.

This module describes how to enable support for the PAI or the PPI privacy header in incoming and outgoing Session Initiation Protocol (SIP) requests or response messages.

Finding Feature Information for PAI or PPI Header in Incoming and Outgoing SIP Calls

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information for Handling PAI or PPI Header in Incoming and Outgoing SIP Calls, on page 494.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Contents

Information About PAI or PPI Header in Incoming and Outgoing SIP Calls

PAI or PPI Header Overview

For incoming SIP requests or response messages, when the PAI or PPI privacy header is set, the SIP gateway builds the PAI or PPI header into the common SIP stack, thereby providing support to handle the call data present in the PAI or PPI header. To process the data from the PAI or PPI header of incoming SIP calls, we recommend that you enable the asserted ID for the incoming dial peer.

For outgoing SIP requests or response messages, when the PAI or PPI privacy header is set, privacy information is sent using the PAI or PPI header.

Note If the PAI or PPI asserted ID is not enabled either in dial-peer configuration mode or at the global level, the call data present in the PAI or PPI header of incoming SIP calls is ignored.

Support for PAI or PPI Header at the Global Level

At the global level, the support for the PAI or PPI header in incoming and outgoing calls is provided using the **asserted-id** command. The **asserted-id** command enables the support for generating the PAI or PPI header, or the Remote-Party-ID (RPID) or FROM header data, to populate outbound calling information.

Support for PAI or PPI Header in Dial-Peer Configuration Mode

In dial-peer configuration mode, the support for the PAI or PPI header in incoming and outgoing calls is provided using the **voice-class sip asserted-id** command. The **voice-class sip asserted-id** command enables the support for generating the PAI or PPI header, or the Remote-Party-ID (RPID) or FROM header data, to populate outbound calling information.

How to Configure PAI or PPI Header in Incoming and Outgoing SIP Calls

Configuring the PAI or PPI Privacy Header at the Global Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **asserted-id {pai | ppi}**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters voice service VoIP-SIP configuration mode.
Step 5	**asserted-id {pai | ppi}** **Example:** `Router(conf-serv-sip)# asserted-id pai`	Configures the privacy header for incoming and outgoing SIP requests and response messages. • **pai** --Specifies the PAI type privacy header. • **ppi** --Specifies the PPI type privacy header.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(conf-serv-sip)# end`	Exits voice service VoIP-SIP configuration mode and returns to privileged EXEC mode.

Configuring the PAI or PPI Privacy Header in Dial-Peer Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial peer voice** *tag*
4. **voice-class sip asserted-id** {**pai** | **ppi** | **system**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial peer voice** *tag* **Example:** `Router(config)# dial peer voice 1`	Enters dial-peer configuration mode.
Step 4	**voice-class sip asserted-id** {**pai** | **ppi** | **system**} **Example:** `Router(config-dial-peer)# voice-class sip asserted-id pai`	Configures the privacy header for incoming and outgoing SIP requests and response messages. • **pai** --Specifies the PAI type privacy header. • **ppi** --Specifies the PPI type privacy header. • **system** --Uses the global-level configuration settings to configure the dial peer.
Step 5	**end** **Example:**	Exits dial-peer configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-dial-peer)# end`	

Configuration Examples for PAI or PPI Header in Incoming and Outgoing SIP Calls

Example Configuring the PAI or PPI Privacy Header at the Global Level

The following example shows how to enable support for the PAI privacy header:

```
Router> enable
Router# configure terminal
Router(config)# voice service voip
Router(conf-voi-serv)# sip
Router(conf-serv-sip)# asserted-id pai
```

Example Configuring the PAI or PPI Privacy Header in Dial-Peer Configuration Mode

The following example shows how to enable support for the PPI header:

```
Router> enable
Router# configure terminal
Router(config)# dial peer voice 1
Router(conf-voi-serv)# voice-class sip asserted-id ppi
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Voice commands	Cisco IOS Voice Command Reference
Cisco IOS Voice Configuration Library	For more information about Cisco IOS voice features, including feature documents, and troubleshooting information: http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary/vcl.htm

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	--

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Handling PAI or PPI Header in Incoming and Outgoing SIP Calls

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 47: Feature Information for PAI or PPI Header in Incoming and Outgoing SIP Calls

Feature Name	Releases	Feature Information
PAI or PPI Header in Incoming and Outgoing SIP Calls	12.4(24)T 15.1(3)T	Prior to the introduction of the PAI or PPI Header in Incoming and Outgoing SIP Calls feature, the PAI or the PPI privacy header was supported for outgoing calls at global level. The PAI or PPI Header in Incoming and Outgoing SIP Calls feature is an enhancement to support the PAI or the PPI privacy header for incoming and outgoing calls at the global level and dial-peer configuration mode. The following commands were introduced or modified: **asserted-id**, **voice-class sip asserted-id**.

CHAPTER 13

Configuring SIP ISDN Features

This chapter discusses the following SIP features that support ISDN:

- ISDN Calling Name Display
- Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks
- SIP Carrier Identification Code (CIC)
- SIP: CLI for Caller ID When Privacy Exists
- SIP: ISDN Suspend/Resume Support
- SIP PSTN Transport Usingthe Cisco Generic Transparency Descriptor (GTD)

Feature History for ISDN Calling Name Display

Release	Modification
12.3(4)T	This feature was introduced.

Feature History for Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks

Release	Modification
12.3(7)T	This feature was introduced.

Feature History for SIP Carrier Identification Code

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for SIP: CLI for Caller ID When Privacy Exists Feature

Release	Modification
12.4(4)T	This feature was introduced.

Feature History for SIP: ISDN Suspend/Resume Support

Release	Modification
12.2(15)T	This feature was introduced.

Feature History for SIP PSTN Transport Using the Cisco Generic Transparency Descriptor

Release	Modification
12.3(1)	This feature was introduced.

Finding Support Information for Platforms and Cisco Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Prerequisites for SIP ISDN Support

ISDN Calling Name Display Feature

- Configure Generic Transparency Descriptor (GTD) on your SIP network.

Note For information on SIP support for communicating ISDN information using GTD bodies, see the "SIP PSTN Transport Using the Cisco Generic Transparency Descriptor".

- Enable the Remote-Party-ID header on your SIP network. In general, Remote-Party-ID is enabled by default and no configuration is necessary. The Remote-Party-ID header provides translation capabilities for ISDN screening and presentation indicators in call setup messages.

Note For information on the Remote-Party-ID header, see the “SIP Extensions for Caller Identity and Privacy” section.

- Use this feature in a uni-directional deployment beginning with an originating gateway. For example, the flow must be from a gateway to a phone or gateway to an application server.

Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks Feature

- Configure the SIP protocol.

SIP: CLI for Caller ID When Privacy Exists

- Establish a working IP network.
- Configure VoIP.
- Ensure that the gateway has voice functionality configured for SIP.

Note For information about configuring voice functionality, see the *Cisco IOS Voice Configuration Library*.

SIP: ISDN Suspend/Resume Support Feature

- Configure ISDN switch types on the gateway to support Suspend and Resume messages.

SIP PSTN Transport Using the Cisco Generic Transparency Descriptor Feature

- Configure your VoIP network, including the following components:
  - Cisco PGW 2200 signaling controller (SC) in Cisco MGC Software Release 9.2(2)

Note The Cisco PGW 2200 SC is formerly known as the Cisco Media Gateway Controller (MGC) and the Cisco SC 2200 signaling controller.

-
  - Cisco Signaling Link Terminal (Cisco SLT), which performs Signaling System 7 (SS7) signal preprocessing for a Cisco PGW 2200 SC
  - Cisco IOS gateways to allow sending and processing of SS7 ISUP messages in GTD format: Cisco IOS Release 12.3(1)
  - Cisco SS7 Interconnect for Voice Gateways solution

Restrictions for SIP ISDN Support

SIP Carrier Identification Code Feature

- SIP gateways receive the CIC parameter in SIP INVITE or 302 REDIRECT messages only.
- SIP gateways do not add or configure CIC parameters.
- The TNS IE in the ISDN SETUP message does not map to the CIC parameter in a SIP INVITE request. It is only the CIC parameter that maps to the TNS IE in the outgoing ISDN SETUP message.

Note The workaround created in Cisco IOS Release 12.3(2)XB is no longer supported with the release of this feature. The workaround handled the CIC parameter by including it in the called-party number. The To header in the SIP INVITE message that contained the called-party number was prefixed with 101xxxx, where xxxx was the CIC parameter. The number was then sent to the ISDN in the SETUP message. When the ISDN received the number, for example, 101032119193921234 the ISDN ignored the 101 and then routed the call to carrier 0321, as if 0321 was in the TNS IE of the outgoing SETUP message. The rest of the number, formatted as the called-party number, was forwarded to the carrier.

- Support for the CIC parameter is addressed by the expired IETF draft-yu-tel-url-02.txt. The SIP Carrier Identification Code feature does not encompass all areas that are addressed in the draft.

SIP: ISDN Suspend/Resume Support Feature

- SIP ISDN Suspend/Resume support is available only for ISDN PRI trunks connected at the gateway.

SIP PSTN Transport Using the Cisco Generic Transparency Descriptor Feature

- Redundant Link Manager (RLM) is a requirement for the SIP PSTN Transport Using the Cisco Generic Transparency Descriptor feature. As a result, only the following platforms that use RLM are supported: Cisco AS5300, Cisco AS5350, and Cisco AS5400.

Note For information on RLM, see *Redundant Link Manager (RLM)*.

- SIP-T also transparently transmits ISUP messages across a SIP network, but the process is not supported in this feature.

Restrictions for ISDN UDI to SIP Clear-Channel

The ISDN UDI to SIP Clear-Channel feature is overridden when the **bearer-cap** [**3100hz** | **speech**] command is configured in voice-port configuration mode.

Information About SIP ISDN Support

To configure SIP ISDN support features, you should understand the following concepts:

ISDN Calling Name Display

With releases earlier than Cisco IOS Release 12.2(15)ZJ, when a call came in from the ISDN network to a SIP gateway, the calling name as presented in ISDN Q.931 messages (Setup and/or Facility) was not transported end-to-end over the VoIP cloud to a SIP endpoint (a SIP IP phone). With this feature, SIP signaling on Cisco IOS gateways has been enhanced to update the calling name and number information in SIP headers as per the recommended SIP standards. Also included is the complete translation of ISDN screening and presentation indicators, allowing SIP customers basic caller ID privileges.

Caller ID in ISDN Networks

In ISDN networks, caller ID (sometimes called CLID or ICLID for incoming calling line identification) is an analog service offered by a central office (CO) to supply calling party information to subscribers. Caller ID allows the calling party number and name to appear on a device such as a telephone display.

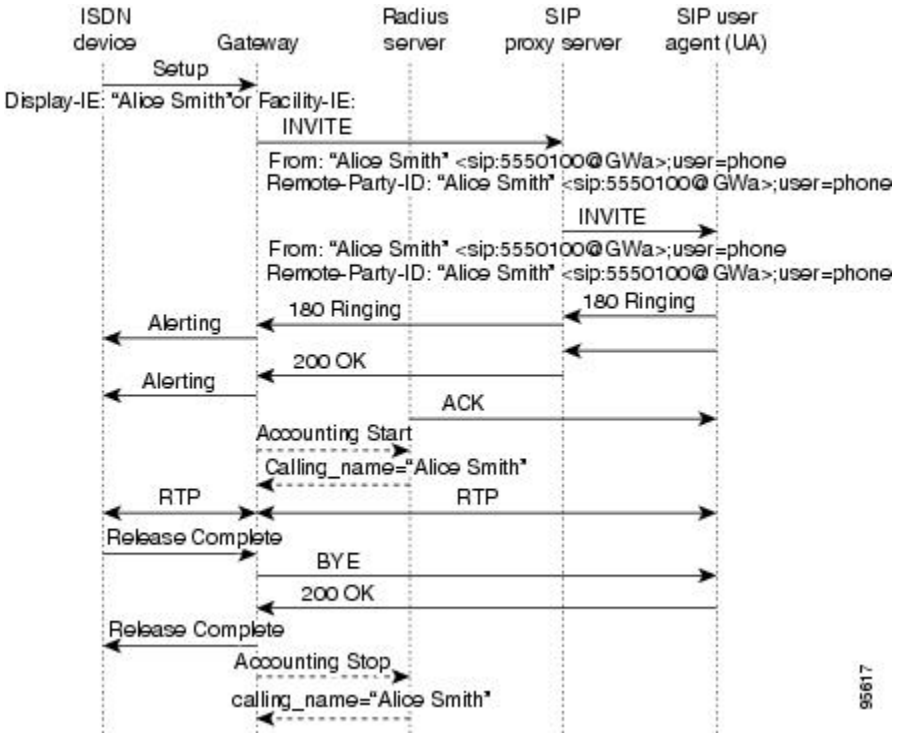
ISDN messages signal call control and are composed of information elements (IEs) that specify screening and presentation indicators. ISDN messages and their IEs are passed in GTD format. GTD format enables transport of signaling data in a standard format across network components and applications. The standard format enables other devices to scan and interpret the data. The SIP network extracts the calling name from the GTD format and sends the calling name information to the SIP customer.

ISDN and SIP Call Flows Showing the Remote-Party-ID Header

The figure below shows the SIP gateway receiving an ISDN Setup message that contains a Display (or Facility) IE indicating the calling name. Receiving the message initiates call establishment.

The Remote-Party-ID header sent by the SIP gateway identifies the calling party and carries presentation and screening information. The Remote-Party-ID header, which can be modified, added, or removed as a call session is being established, enables call participant privacy indication, screening, and verification.

Figure 78: Calling Name in Display or Facility IE of an ISDN Setup Message

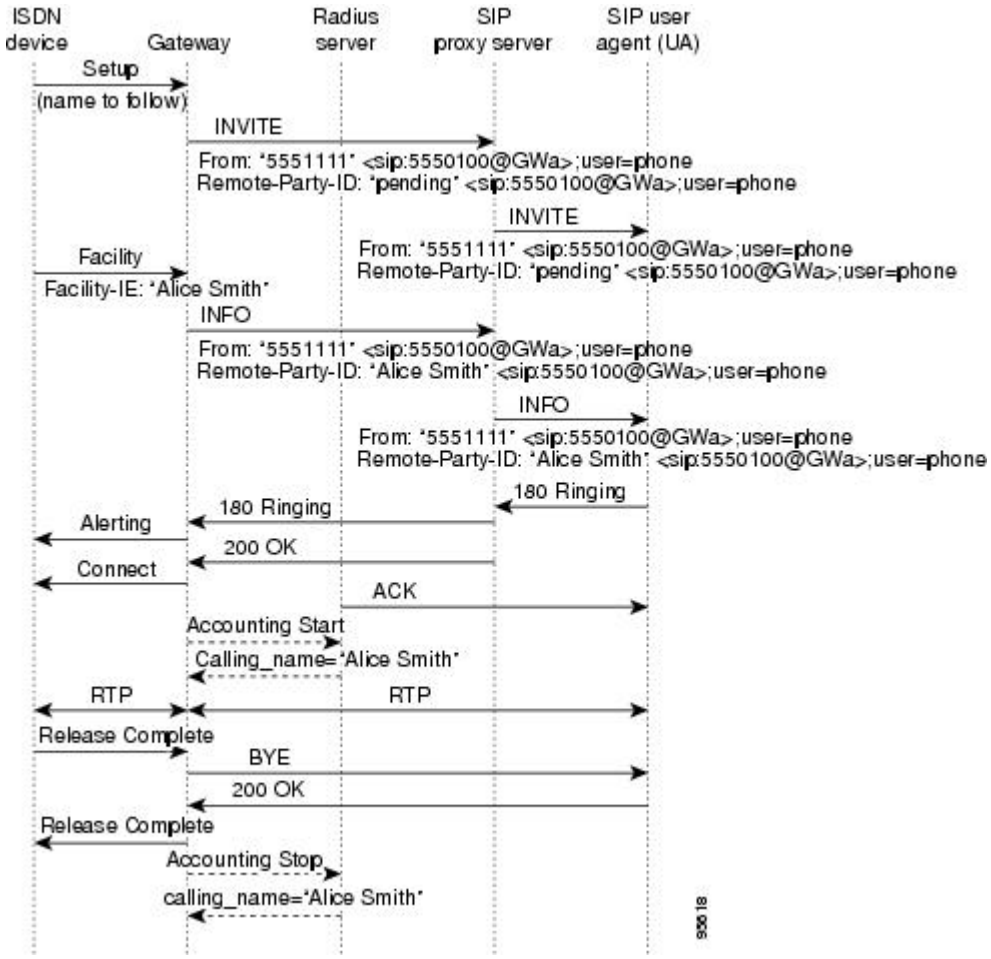


The figure below shows that the original ISDN Setup message sent by the ISDN device does not contain a Facility IE. The SIP gateway receives the ISDN Setup message indicating that the calling name is to be delivered in a subsequent ISDN Facility message. The SIP gateway then sets the display name of the Remote-Party-ID to *pending* . The presence of *pending* in a calling Remote-Party-ID of an INVITE denotes that the display name is to follow.

The functionality of a calling name sent in a subsequent message requires that:

- The ISDN switch type has the ability to indicate that the name follows in the next Facility message after the initial ISDN Setup message.
- The SIP gateway has the ability to interpret the subsequent Facility message into a SIP message. The SIP INFO message is used to interpret the Facility received from the ISDN device.

Figure 79: Calling Name in Facility IE of an ISDN Facility Message



Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks

The Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks feature enables call management applications to identify specific ISDN bearer (B) channels used during a voice gateway call for billing purposes. With the identification of the B channel, SIP gateways can enable port-specific features such as voice recording and call transfer.

In Cisco IOS releases prior to 12.3(7)T, fields used to store call leg information regarding the telephony port do not include B channel information. B channel information is used to describe incoming ISDN call legs. The Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks feature allows SIP

and H.323 gateways to receive B-channel information from incoming ISDN calls. The acquired B channel information can be used during call transfer or to route a call.

SIP gateways use the **ds0-num** command to enable receiving the B channel of a telephony call leg. H.323 gateways use a different command, which allows users to run the two protocols on one gateway simultaneously.

Note For information on using this feature on H.323 gateways, see Configuring H.323 Gateways.

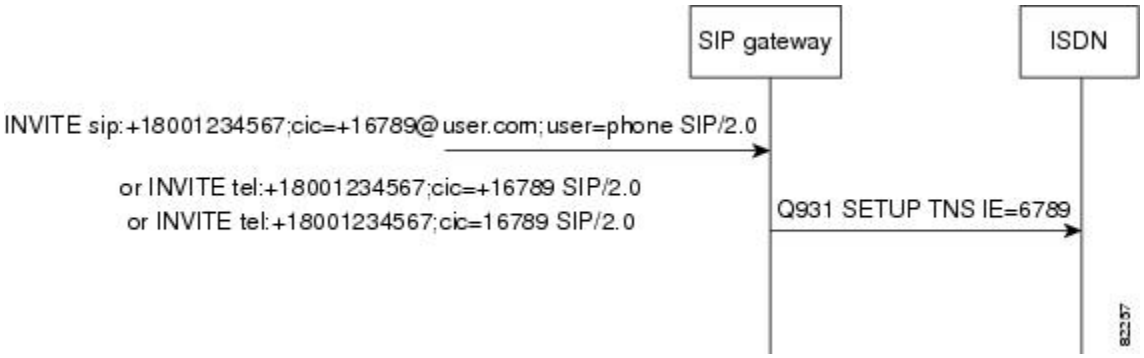
For SIP, if the **ds0-num** command is configured, the ISDN B-channel information is carried in the Via header of outgoing SIP requests.

SIP Carrier Identification Code

SIP gateways can receive and transmit the carrier identification code (CIC) parameter, allowing equal access support over many different networks. CIC enables transmission of the CIC parameter from the SIP network to the ISDN.

The CIC parameter is used in routing tables to identify the network that serves the remote user when a call is routed over many different networks. The parameter is carried in SIP INVITE requests and 302 REDIRECTs, and maps to the ISDN Transit Network Selection Information Element (TNS IE) in the outgoing ISDN SETUP message (see the figure below). The TNS IE identifies the requested transportation networks and allows different providers equal access support based on customer choice.

Figure 80: Path of INVITE request with CIC Parameter to SIP Gateway Receiving and to ISDN



The CIC parameter is supported in SIP URLs, which identify a user's address and appear similar to e-mail addresses: user@host. It is also supported in the telephone-subscriber part of a TEL URL, which takes the basic form of *tel:telephone subscriber number* , where *tel* requests the local entity to place a voice call, and *telephone subscriber number* is the number to receive the call.

The CIC parameter can be a three-digit or a four-digit code. However, if it is a three-digit code, it is prefixed by a zero as in the following example:

```
cic=+1234 = TNS IE 0234.
```

SIP CLI for Caller ID When Privacy Exists

The SIP: CLI for Caller ID When Privacy Exists feature is comprised of three main components, as follows:

SIP Caller ID Removable to Improve Privacy

The caller ID information is passed through from the ISDN-to-SIP by copying the number in the Calling Party Number information element (IE) in an ISDN Setup message into the Calling Number field of the SIP Remote-Party-ID and From headers.

The Calling Name from the ISDN Display IE is copied into the SIP Display Name field in the SIP Remote-Party-ID and From headers. The Calling Party Number IE contains a Presentation Indicator field that is set to presentation allowed, presentation restricted, number not available due to interworking, or reserved. Presentation allowed and presentation restricted are translated into privacy set to off or privacy set to null, respectively, in the SIP Remote-Party-ID header field.

However, for added privacy, the SIP: CLI for Caller ID When Privacy Exists feature introduces CLI to completely remove the Calling Number and Display Name from an outgoing message's From header if presentation is prohibited. This prohibits sending the SIP Remote Party ID header, because the Cisco gateway does not send SIP Remote-Party ID headers without both a Display Name and Calling Number.

Note The SIP: Caller ID Removable to Improve Privacy option is available both globally and at the dial-peer level.

See the figure below for call flows and the tables below for additional presentation mapping.

Figure 81: Call Flow for Blocking Caller ID Information When Privacy Exists

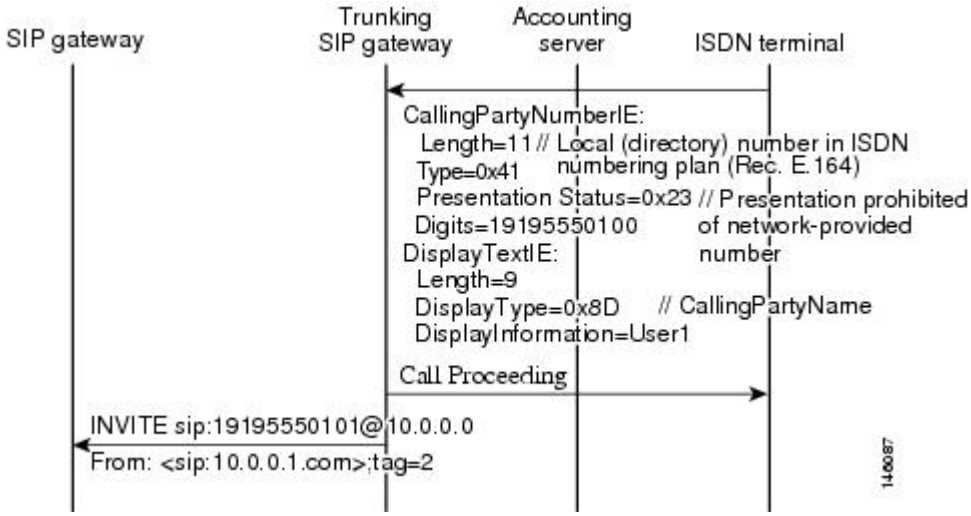


Table 48: Presentation to Privacy Mapping with CLI Disabled

Presentation Indicator	From Remote Party ID (RPID)
Presentation Allowed	From: "User1" <sip:19195550100@10.0.0.0>;tag=1 Remote-Party-ID: "User1" <sip:19195550100@10.0.0.0>;party=calling;privacy=off
Presentation Prohibited	From: "User1" <sip:19195550100@10.0.0.0>;tag=1 Remote-Party-ID: "User1"<sip:19195550100@10.0.0.0>;party=calling;privacy=full

Table 49: Presentation to Privacy Mapping with CLI Enabled

Presentation Indicator	From RPID
Presentation Allowed	From: “User1” <sip:19195550100@10.0.0.0>;tag=1 Remote-Party-ID: “User1”<sip:19195550100@10.0.0.0>;party=calling;privacy=off
Presentation Prohibited	From: <sip:10.0.0.0>;tag=1 Remote Party ID not sent

SIP Calling Number Substitution for the Display Name When the Display Name is Unavailable

When the Display information element (IE) in a PSTN-to-SIP call is not available with a Setup message, the Cisco gateway leaves the Display Name field in the SIP Remote-Party-ID and From headers blank.

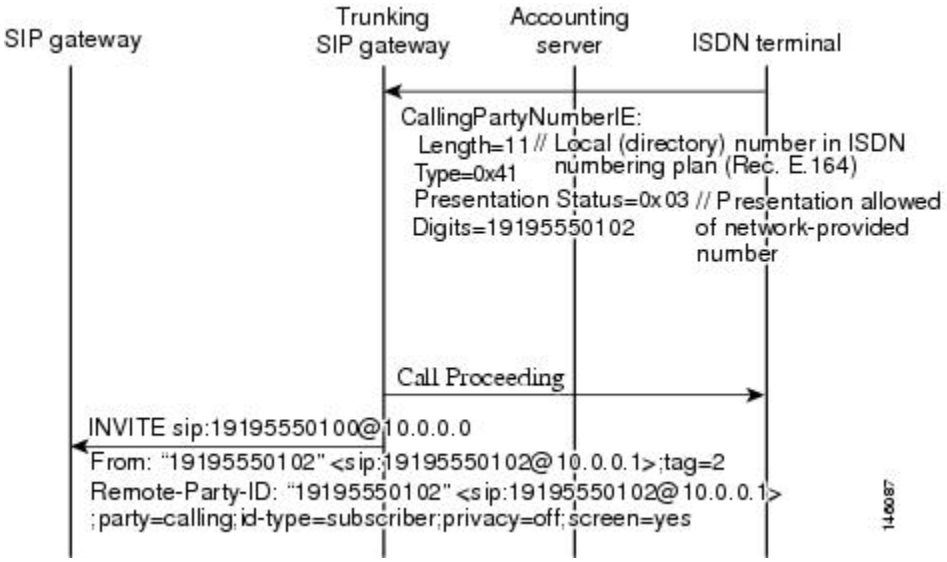
When presentation is allowed, the SIP: CLI for Caller ID When Privacy Exists feature enables the substitution of the Calling Number for the missing Display Name in the SIP Remote-Party-ID and From headers. Upon receipt of a Setup message where a name to follow is indicated, the Calling Number is not copied into the Display Name.

Also, the SIP Extensions for Caller Identity and Privacy on SIP gateway feature added the ability to hardcode calling name and number in the SIP Remote-Party-ID and From headers. The SIP Extensions for Caller Identify and Privacy feature settings take precedence over the SIP: CLI for Caller ID When Privacy Exists feature settings.

Note The SIP: Calling Number Substitution for the Display Name When the Display Name is Unavailable option is available both globally and at the dial-peer level.

See the figure below for the call flow where the Calling Number is substituted for the Display Number.

Figure 82: Call Flow for Substituting the Calling Number for the Display Name When the Display Name is Unavailable



SIP Calling Number Passing as Network-Provided or User-Provided

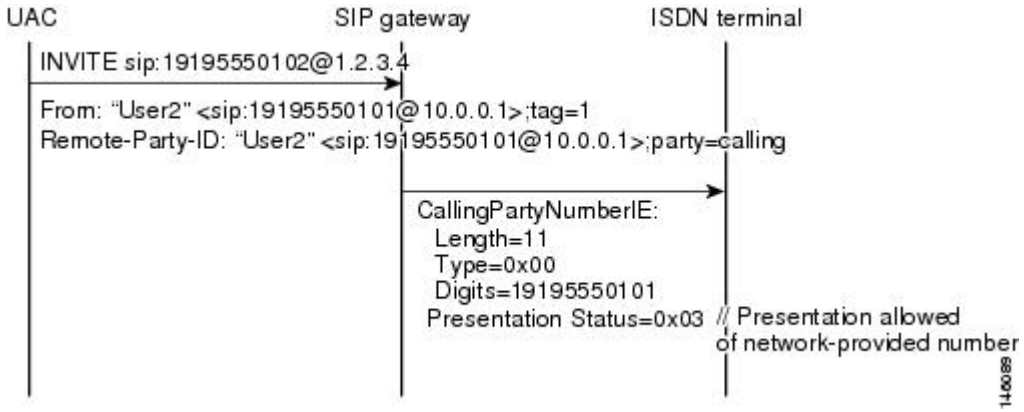
ISDN numbers can be passed along as network-provided or user-provided in an ISDN Calling Party information element (IE) Screening Indicator field. The Cisco gateway automatically sets the Screening Indicator to user-provided in SIP-to-ISDN calls.

The SIP: CLI for Caller ID When Privacy Exists feature allows toggling between user-provided and network-provided ISDN numbers for the screening indicator. Therefore, after bits 1 and 2 are set to reflect network-provided, any existing screening information is lost. However, presentation information in bits 6 and 7 is preserved.

Note The Call Flow for Passing Through the Calling Number as Network-Provided option is available both globally and at the dial-peer level.

See the figure below for the call flow when the calling number is passed along as network-provided.

Figure 83: Call Flow for Passing Through the Calling Number as Network-Provided



SIP ISDN Suspend Resume Support

Suspend and Resume are basic functions of ISDN and ISDN User Part (ISUP) signaling procedures and now are a part of SIP functionality. Suspend is described in ITU Q.764 as a message that indicates a temporary cessation of communication that does not release the call. A Suspend message can be accepted during a conversation. A Resume message is received after a Suspend message and is described in ITU Q.764 as a message that indicates a request to recommence communication. If the calling party requests to release the call, the Suspend and Resume sequence is overridden.

SIP Call-Hold Process

When a SIP originating gateway receives an ISDN Suspend message, the originating gateway informs the terminating gateway that there is a temporary cessation of media; that is, the call is placed on hold. There are two ways that SIP gateways receive notice of a call hold. The first way is for the originating gateway to use a connection IP address of 0.0.0.0 (c=0.0.0.0) in the Session Description Protocol (SDP). The information in the SDP is sent in a re-Invite to the terminating gateway. The second way is for the originating gateway to use a=sendonly in the SDP of a re-Invite.

Note Earlier than Cisco IOS Release 12.3(8)T, a SIP gateway could initiate call hold only by using c=0.0.0.0. As of Cisco IOS Release 12.3(8)T, a gateway can initiate call hold by using either c=0.0.0.0 or a=sendonly.

The purpose of the c=0.0.0.0 line is to notify the terminating gateway to stop sending media packets. When the hold is cancelled and communication is to resume, an ISDN Resume message is sent. The SIP originating gateway takes the call off hold by sending out a re-Invite with the actual IP address of the remote SIP entity in the c= line (in place of 0.0.0.0).

Multiple media fields (m-lines) in the SDP of a re-Invite message are used to indicate media forking, with each m-line representing one media destination. SIP gateways negotiate multiple media streams by using multiple m- and/or c-lines. When an originating gateway receives an ISDN Suspend on a gateway that has negotiated multiple media streams, all of the media streams are placed on hold. The originating gateway sends out a re-Invite that has a c= line that advertises the IP address as 0.0.0.0 on all streams. The originating gateway also mutes the SIP calls for each media stream so that no media is sent to the terminating gateway. When the originating gateway receives an ISDN Resume, it initiates a re-Invite with the original SDP and takes the call off hold.

If the media inactivity timer is configured on the network, the timer is stopped for all active streams. The purpose of the media inactivity timer is to monitor and disconnect calls if no Real-Time Control Protocol (RTCP) packets are received within a configurable time period. However, on initiating the call hold, the originating gateway disables the media inactivity timer for that particular call, so the call remains active. The terminating gateway behaves in the same way when it receives the call-hold re-Invite from the originating gateway. When the call resumes, the originating gateway re-enables the Media Inactivity Timer.

Note For information on the timer, see the “SIP Media Inactivity Timer” section.

All billing and accounting procedures are unaffected by the SIP: ISDN Suspend/Resume Support feature.

SIP PSTN Transport Using the Cisco Generic Transparency Descriptor

The SIP PSTN Transport Using the Cisco Generic Transparency Descriptor feature adds SIP support for ISDN User Part (ISUP) Transport using Generic Transparency Descriptor (GTD). The ISUP data received on the originating gateway (OGW) is preserved and passed in a common text format to the terminating gateway (TGW).

Feature benefits include the following:

- The ISUP data is reconstructed on the basis of the protocol at the egress side of the network, without any concern for the ISDN or ISUP variant on the ingress side of the network.
- By providing the ISDN or ISUP information in text format, the information can also be used by applications inside the core SIP network. An example of one such application is a route server that can use certain ISDN or ISUP information to make routing decisions.
- The transport of ISUP encapsulated in GTD maintains compatibility with the H.323 protocol.

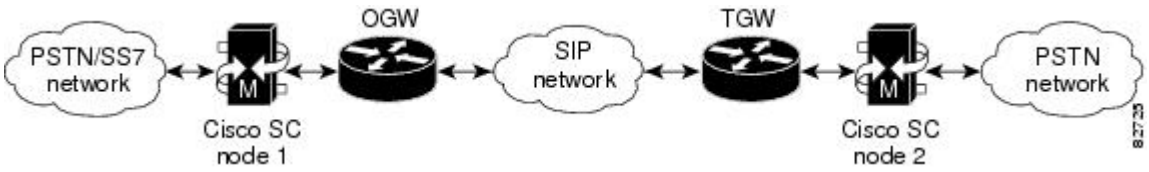
SIP ISUP Transparency Using GTD Overview

The SIP PSTN Transport Using the Cisco Generic Transparency Descriptor feature adds SIP support for ISUP transport using GTD. That is, ISUP data received on the OGW is preserved and presented in a common ASCII format to the TGW.

GTD objects can be used to represent ISUP messages, parameters, and R2 signals. These GTD objects are encapsulated into existing signaling protocols, such as SIP, facilitating end-to-end transport. The transport of ISUP encapsulated in GTD ASCII format already exists for H.323; SIP PSTN Transport Using the Cisco Generic Transparency Descriptor provides feature parity. Using GTD as a transport mechanism for signaling data in Cisco IOS software provides a common format for sharing signaling data between various components in a network and for interworking various signaling protocols.

To attain ISUP transparency in VoIP Networks, the gateway needs to externally interface with the Cisco SC node. The Cisco SC node is the combination of hardware (Cisco PGW 2200 and Cisco SLTs) and signaling controller software that provides the signaling controller function. The Cisco SC node transports the signaling traffic between the SC hosts and the SS7 signaling network. A brief example of the process of an ISDN message containing an ISUP GTD message that comes into the Cisco OGW from a Cisco SC node is described below and shown in the figure below.

Figure 84: ISUP Transparency Implementation



The process in the figure above is as follows:

1. Cisco SC node 1 receives an ISUP message from the public switched telephone network (PSTN). This node is now responsible for mapping the ISUP message into a GTD format and encapsulating this GTD body within the ISDN message that is sent to the OGW.
2. The SIP user agent on the OGW extracts the GTD body from the Q931 message and encapsulates it into a corresponding SIP message as a multipart MIME attachment.
3. The SIP message is sent by the OGW over the SIP network to the TGW.
4. The TGW encapsulates the GTD in the outgoing ISDN message whish is sent to SC node 2. The SC then remaps the GTD to ISUP before passing it to the PSTN.

SIP INFO Message Generation and Serialization

The SIP PSTN Transport Using the Cisco Generic Transparency Descriptor (GTD) feature adds client and server support for the SIP INFO message in all phases of a call. INFO messages are used to carry ISUP messages that were encapsulated into GTD format, but that do not have a specific mapping to any SIP response or request. These ISUP messages can be received in any phase of the call.

Note For specific mapping messages, see "ISUP-to-SIP Message Mapping".

The gateway does not support sending out overlapping SIP INFO messages. For example, a second INFO message cannot be sent out while one is still outstanding. Multiple PSTN messages that map to SIP INFO messages are sent out serially.

Transporting ISDN Messages in GTD Format

Support for ISDN messages in GTD format is limited to the ISDN Setup message. Only the following parameters are encoded and decoded:

- Originating-line information
- Bearer capability
- Calling-party number
- Called-party number
- Redirecting number

Whereas ISDN to GTD parameter mapping is enabled by default, you must configure the gateway to transport ISUP messages through SIP signaling.

The ISDN parameters can be transported using either GTD or SIP headers. Before the SIP PSTN Transport Using the Cisco Generic Transparency Descriptor feature, only SIP headers provided ISDN parameters. For instance, the user portion of a SIP From header can carry the ISDN Calling Party information element.

SIP headers generally contain the same information that is provided by GTD, because the headers are built on the OGW using information gained from the PSTN. However, there are situations in which the data may be in conflict. The inconsistent data occurs if the header was updated by an intermediate proxy or application server. In cases of conflict, the SIP header is used to construct the ISDN parameters on the TGW, because it generally contains the most recent information.

SIP Generation of Multiple Message Bodies

Before this feature, the SIP gateway handled only SDP as a message body type. With SIP PSTN Transport Using the Cisco Generic Transparency Descriptor, it is now possible for the gateway to generate and properly format messages that contain both SDP and GTD message body types.

Any SIP message that contains both SDP and GTD bodies may be large enough to require link-level fragmentation when User Datagram Protocol (UDP) transport is used, which could result in excessive retransmissions. TCP transport can be used if fragmentation becomes a performance issue.

ISUP-to-SIP Message Mapping

SIP PSTN Transport Using the Cisco Generic Transparency Descriptor attempts to map particular ISUP messages to an equivalent SIP message. This mapping is defined in the table below.

Table 50: Mapping of Supplemental ISUP Messages to SIP Messages

ISUP Message Type	ISDN (NI2C) Message Type	SIP Message Type
ACM	Alerting	180/183 Progress messages
ANM	Connect	200 OK to the INVITE request
CON	Connect	200 OK to the INVITE request

ISUP Message Type	ISDN (NI2C) Message Type	SIP Message Type
CPG	Progress	180/183 Progress messages
IAM	Setup	INVITE request
REL	Disconnect	BYE/CANCEL/4xx/5xx/6xx
RES	Resume	INVITE request
SUS	Suspend	INVITE

Note There are many other PSTN or SS7 messages that are mapped into GTD formats within an ISDN message by the SC node. If the mapping is not listed in the table, the message is treated with the SIP INFO method.

ISDN UDI to SIP Clear-Channel

The ISDN UDI to SIP Clear-Channel feature maps the ISDN bearer capability to an appropriate codec on the Session Initiation Protocol (SIP) trunk. When an ISDN bearer capability message is received as an Unrestricted Digital Information (UDI), only the clear-channel codec is used for negotiation on the SIP trunk. When the ISDN bearer capability message is non-UDI, like speech, the specific voice codecs are used for negotiation on the SIP trunk. The ISDN UDI to SIP Clear-Channel feature is applicable only for clear-channel and voice codecs. Integrated Service Router (ISR) gateways receive calls on ISDN trunks and forward them to SIP IP trunks.

The ISDN UDI to SIP Clear-Channel feature advertises only the clear-channel codec when the ISDN has the bearer capability of UDI (this is meant for data calls), and advertises only voice codecs when the ISDN bearer capability is speech. This behavior is true when clear-channel codecs and voice codecs are configured either individually or together through voice-class codecs. The call is terminated with ISDN cause code 65 (bearer capability not implemented) if either:

- UDI is received, but the clear channel is not configured.
- Non-UDI bearer capability is received but only the clear channel is configured.

How to Configure SIP ISDN Support Features

For help with a procedure, see the troubleshooting section listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring ISDN Calling Name Display

To enable SIP IP phones to display caller-name identification for calls that originate on an ISDN network, perform the following task.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **voice service voip**
4. **signaling forward** {**none** | **unconditional**}
5. **exit**
6. **interface serial** *slot* / *port* : *timeslot*
7. **isdn supp-service name calling**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode.
Step 4	**signaling forward** {**none**	**unconditional**} **Example:** `Router(conf-voi-serv)# signaling forward unconditional`	Specifies whether or not the originating gateway (OGW) forwards the signaling payload to the terminating gateway (TGW). Keywords are as follows: • **none** --Prevent the gateway from passing the signaling payload to the TGW. • **unconditional** --Forward the signaling payload received in the OGW to the TGW, even if the attached external route server has modified the GTD payload.
Step 5	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits the current mode.
Step 6	**interface serial** *slot* / *port* : *timeslot* **Example:** `Router(config)# interface serial 1/0:23`	Specifies a serial interface created on a channelized E1 or channelized T1 controller. You must explicitly specify a serial interface. Arguments are as follows: • *slot* / *port* --Slot and port where the channelized E1 or T1 controller is located. The slash is required.

	Command or Action	Purpose
		• **:** *time-slot* --For ISDN, the D-channel time slot, which is the 23 channel for channelized T1 and the 15 channel for channelized E1. The colon is required.
Step 7	**isdn supp-service name calling** **Example:** `Router(config-if)# isdn supp-service name calling`	Sets the calling-name display parameters sent out an ISDN serial interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits the current mode.

Configuring Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **ds0-num**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters VoIP voice-service configuration mode.

	Command or Action	Purpose
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**ds0-num** **Example:** `Router(conf-serv-sip)# ds0-num`	Adds B-channel information to outgoing SIP messages.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIP Carrier Identification Code

SUMMARY STEPS

1. **debug ccsip messages**
2. **debug isdn q931**

DETAILED STEPS

Step 1 **debug ccsip messages**

Use this command to show all SIP SPI message tracing. Use it on a terminating gateway to verify the incoming CIC parameter.

Examples:

This example shows output of an INVITE request that uses a SIP URL and contains a CIC parameter:

Example:

```
Router# debug ccsip messages
00:03:01: Received:
INVITE sip:5550101;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.18.202.62:5060
From: <sip:4440001@172.18.202.62>;tag=24176150-1A11
To: <sip:5550101@172.18.202.60;user=phone>
Date: Mon, 08 Mar 1993 00:11:51 GMT
Call-ID: 590F6480-1A7011CC-80B5CC57-1D726644@172.18.202.62
Supported: 100rel
Cisco-Guid: 1494180992-443552204-2159266903-494036548
User-Agent: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731549511
Contact: <sip:4440001@172.18.202.62:5060;user=phone>
Expires: 180
```

```
Allow-Events: telephone-event, x-com-cisco-telephone-event, x-com-cisco-fail-telephone-event
Content-Type: application/sdp
Content-Length: 160
```

The following shows output of an INVITE request that uses a TEL URL and contains a CIC parameter:

Example:

```
Router# debug ccsip messages
00:01:00: Received:
INVITE tel:+5550101;cic=+16789 SIP/2.0
Via: SIP/2.0/UDP 172.18.202.62:5060
From: <sip:4440001@172.18.202.62>;tag=24158B04-1D45
To: <sip:5550101@172.18.202.60;user=phone>
Date: Mon, 08 Mar 1993 00:09:51 GMT
Call-ID: 114C6D4C-1A7011CC-80B0CC57-1D726644@172.18.202.62
Supported: 100rel
Cisco-Guid: 290221388-443552204-2158939223-494036548
User-Agent: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Max-Forwards: 6
Timestamp: 731549391
Contact: <sip:4440001@172.18.202.62:5060;user=phone>
Expires: 180
Allow-Events: telephone-event, x-com-cisco-telephone-event, x-com-cisco-fail-telephone-event
Content-Type: application/sdp
Content-Length: 160
```

Step 2 **debug isdn q931**

Use this command to display information about call setup and teardown of ISDN network connections (layer 3) between the local router (user side) and the network. Use it to verify the contents of the CIC parameter and the TNS IE.

Example:

This example shows output of an outgoing call SETUP that contains the TNS IE. Output is the same for either a SIP or TEL URL.

Example:

```
Router# debug isdn q931
00:01:00: ISDN Se2/0:23: TX -> SETUP pd = 8 callref = 0x0001
00:01:00: Bearer Capability i = 0x8090A2
00:01:00: Channel ID i = 0xA98397
00:01:00: Calling Party Number i = 0x0081, '4440001', Plan:Unknown, Type:Unknown
00:01:00: Called Party Number i = 0xA8, '5550101', Plan:National, Type:National
00:01:00: Transit Net Select i = 0xA1, '6789'
```

Configuring SIP CLI for Caller ID When Privacy Exists

Configuring SIP Blocking Caller ID Information Globally When Privacy Exists

The Call-ID information is private information. In ISDN there is a private setting that can be set to protect this information. However, whenever SIP gets the Call-ID information, it does not hide the private information, rather, it just sets a field to reflect that it is private and not to display it on a Call-ID display. But, the data is still viewable in the SIP message requests. This option allows the Cisco gateway to delete the Call-ID information from the SIP message requests so it cannot be read on the network.

Upon receiving an ISDN Setup message with the calling-party information element, the Cisco gateway translates the presentation indicator to set privacy to full for restricted presentation or to set privacy to off for unrestricted presentation in the Remote-Party-ID header field. The SIP: CLI for Caller ID When Privacy Exists feature introduces a CLI switch that either allows stripping the Calling Number and Display Name from the From and Remote-Party-ID fields in the SIP message requests or passes on the information. However, in cases of unrestricted presentation, the gateway passes the caller ID information, regardless of the CLI setting.

The global commands to strip the Calling Name and Calling Number from the Remote-Party-ID and From headers are as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **clid strip pi-restrict all**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service-VoIP configuration mode.
Step 4	**clid strip pi-restrict all** **Example:** `Router(config-voip-serv)# clid strip pi-restrict all`	Enters block call ID information when privacy exists in global configuration mode.
Step 5	**exit** **Example:** `Router(config-voip-serv)# exit`	Exits the current mode.

Configuring Dial-Peer Level SIP Blocking of Caller ID Information When Privacy Exists

The dial-peer specific command to strip the Calling Number from the Remote-Party-ID and From headers is as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice dial-peer-number voip**
4. **clid strip pi-restrict all**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice dial-peer-number voip** **Example:** `Router(config)# dial-peer voice 100 voip`	Enters dial-peer configuration mode.
Step 4	**clid strip pi-restrict all** **Example:** `Router(config-dial-peer)# clid strip pi-restrict all`	Enters block call ID information when privacy exists in dial-peer configuration mode.
Step 5	**exit** **Example:** `Router# exit`	Exits the current mode.

Configuring Globally the SIP Calling Number for Display Name Substitution When Display Name Is Unavailable

When this is enabled, if there is no Display Name field but there is a number, it copies the number into the Display Name field, so the number is displayed on the recipient's Call-ID display.

The Cisco gateway omits the Display Name field if no display information is received. This feature also introduces a CLI switch that allows the Calling Number to be copied into the Display Name field, as long as presentation is not prohibited.

The steps for substituting the Calling Number for the Display Name when it is unavailable in the Remote-Party-ID and From headers are as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **clid substitute name**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service-VoIP configuration mode.
Step 4	**clid substitute name** **Example:** `Router(config-voip-serv)# clid substitute name`	Substitutes the calling number for the display name when the display name is unavailable in the global configuration mode.
Step 5	**exit** **Example:** `Router(config-voip-serv)# exit`	Exits the current mode.

Configuring Dial-Peer-Level SIP Substitution of the Calling Number

The dial-peer-specific steps for substituting the Calling Number for the Display Name when it is unavailable in the Remote-Party-ID and From headers are as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice dial-peer-number voip**
4. **clid substitute name**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice dial-peer-number voip** **Example:** `Router(config-dial-peer)# dial-peer voice 100 voip`	Enters dial-peer configuration mode.
Step 4	**clid substitute name** **Example:** `Router(config-dial-peer)# clid substitute name`	Substitutes the calling number for the display name when the display name is unavailable in dial-peer configuration mode.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Globally the SIP Pass-Through of the Passing Calling Number as Network-Provided

This field shows whether the Call-ID information was supplied by the network or not. This is for screening purposes.

Formerly the Calling Number from the session initiation protocol to public switched telephone network (SIP-to-PSTN) was always translated to user-provided. This feature introduces a CLI switch to toggle between branding numbers as user-provided or network-provided.

The steps for globally setting set the Screening Indicator to network-provided are as follows:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **voice service voip**
4. **clid network-provided**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service-VoIP configuration mode.
Step 4	**clid network-provided** **Example:** `Router(config-voip-serv)# clid network-provided`	Enters the network-provided calling number in voice-service-VoIP configuration mode.
Step 5	**exit** **Example:** `Router(config-voip-serv)# exit`	Exits the current mode.

Configuring at the Dial-Peer Level the SIP Pass-Through of Passing the Calling Number as Network-Provided

The dial-peer specific command to set the Screening Indicator to network-provided is as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice dial-peer-number voip**
4. **clid network-provided**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice dial-peer-number voip** **Example:** `Router(config)# dial-peer voice 100 voip`	Enters dial-peer configuration mode.
Step 4	**clid network-provided** **Example:** `Router(config-dial-peer)# clid network-provided`	Enters the network-provided calling number in dial-peer configuration mode.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Globally the SIP Pass-Through of the Passing Calling Number as User-Provided

The steps for globally setting set the Screening Indicator to user-provided are as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **no clid network-provided**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service-VoIP configuration mode.
Step 4	**no clid network-provided** **Example:** `Router(config-voip-serv)# no clid network-provided`	Enters the network-provided calling number in voice-service-VoIP configuration mode.
Step 5	**exit** **Example:** `Router(config-voip-serv)# exit`	Exits the current mode.

Configuring at the Dial-Peer Level the SIP Pass-Through of Passing the Calling Number as User-Provided

The dial-peer specific command to set the Screening Indicator to user-provided is as follows:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice dial-peer-number voip**
4. **no clid network-provided**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dial-peer voice dial-peer-number voip** **Example:** `Router(config)# dial-peer voice 100 voip`	Enters dial-peer configuration mode.
Step 4	**no clid network-provided** **Example:** `Router(config-dial-peer)# no clid network-provided`	Enters the user-provided calling number in dial-peer configuration mode.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring SIP ISDN Suspend Resume Support

Suspend and Resume functionality is enabled by default. However, the functionality is also configurable. To configure Suspend and Resume for all dial peers on the VoIP network, perform the steps below on both originating and terminating gateways.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **suspend-resume**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.

	Command or Action	Purpose
Step 4	**suspend-resume** **Example:** `Router(config-sip-ua)# suspend-resume`	Enables support for Suspend and Resume.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring SIP PSTN Transport Using the Cisco Generic Transparency Descriptor

To forward the GTD payload to the gateway either for all dial peers on the VoIP network or for individual dial peers, perform the following steps.

Before you begin

- Configure the Cisco PGW2200 to encapsulate SS7 ISUP messages in GTD format before using the **signaling forward** command with the Cisco PGW 2200 signaling controller on the Cisco gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **voice service voip**
4. **signaling forward** {**none** | **unconditional**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following:	Enters one of the following configuration modes:

	Command or Action	Purpose
	• **voice service voip** **Example:** `Router(config)# voice service voip` **Example:** **Example:** **Example:** **Example:** **dial-peer voice** *tag* {**pots** | **voip** | **mmoip** | **vofr** | **voatm**} **Example:** `Router(config)# dial-peer voice 100 voip`	• Voice-service configuration mode for all dial peers on the VoIP network • Dial-peer voice configuration mode for an individual dial peer
Step 4	**signaling forward** {**none** | **unconditional**} **Example:** `Router(conf-voi-serv)# signaling forward unconditional`	Specifies whether or not the OGW forwards the signaling payload to the TGW. Keywords are as follows: • **none** --Prevent the gateway from passing the signaling payload to the TGW. • **unconditional** --Forward the signaling payload received in the OGW to the TGW, even if the attached external route server has modified it. **Note** The **conditional**keyword is not supported for SIP configuration. If you specify that keyword, the gateway treats it as if you had specified **none**.
Step 5	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits the current mode.

Verifying SIP ISDN Support Features

To verify configuration of SIP ISDN support features, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show running-config**
2. **show dial-peer voice**
3. **show sip-ua status**

DETAILED STEPS

Step 1 **show running-config**

Use this command to display the configuration and verify that the correct dial peers were changed.

Step 2 **show dial-peer voice**

Use this command, for each dial peer configured, to verify that the dial-peer configuration is correct.

Step 3 **show sip-ua status**

Use this command to display whether Suspend and Resume support is enabled or disabled.

The following sample output shows that Suspend and Resume support is enabled.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 1 (rfc 2052)
SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Session name line (s=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
SIP support for ISDN SUSPEND/RESUME: ENABLED
```

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "General Troubleshooting Tips" section.

- Make sure that you can make a voice call.
- Use the **debug ccsip messages**command as shown in the examples below.
- Use the **debug ccsip messages** command to enable traces for SIP messages, such as those that are exchanged between the SIP user-agent client (UAC) and the access server.
- Use the **debug isdn q931**command to display information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network.

Following is sample output for some of these commands:

Sample Output for the debug ccsip messages Command

The following is a sample INVITE request with B-channel information added as an extension parameter "x-ds0num" to the Via header. The format of the B-channel billing information is: 0 is the D-channel ID, 0 is the T1 controller, and 1 is the B-channel.

```
Router# debug ccsip messages
INVITE sip:3100802@172.18.193.99:5060 SIP/2.0
Via: SIP/2.0/UDP  172.18.193.100:5060;x-ds0num="ISDN 0:D 0:DS1 1:DS0"
From: <sip:3100801@172.18.193.100>;tag=21AC4-594
To: <sip:3100802@172.18.193.99>
Date: Thu, 28 Dec 2000 16:15:28 GMT
Call-ID: 7876AC6C-DC1311D4-8005DBCA-A25DA994@172.18.193.100
Supported: 100rel
Cisco-Guid: 1981523172-3692237268-2147670986-2724047252
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO
CSeq: 101 INVITE
Max-Forwards: 6
Remote-Party-ID: <sip:3100801@172.18.193.100>;party=calling;screen=no;privacy=off
Timestamp: 978020128
Contact: <sip:3100801@172.18.193.100:5060>
Expires: 300
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 254
^M
v=0
o=CiscoSystemsSIP-GW-UserAgent 45 7604 IN IP4 172.18.193.100
s=SIP Call
c=IN IP4 172.18.193.100
t=0 0
m=audio 19492 RTP/AVP 18 0
c=IN IP4 172.18.193.100
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:0 PCMU/8000
```

The following sample INVITE request shows the Via header if the incoming trunk is T3. The format of the B-channel billing information is: 7/0 is the T3 controller, 1 is the T1 controller, and 2 is the B channel.

```
Router# debug ccsip messages
Via: SIP/2.0/UDP 172.18.193.120:5060; x-ds0num="ISDN 7/0:D 1:D1 2:DS0"
```

Configuring ISDN UDI to SIP Clear-Channel Feature

Perform this task to configure the ISDN UDI to SIP Clear-Channel feature.

Configuring the ISDN UDI to SIP Clear-Channel feature only maps the ISDN UDI bearer capability to the clear-channel codec. However, it does not select the encapsulation type to be used for the clear-channel codec. You must select the clear-channel codec encapsulation at the global level or the dial-peer level after performing this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **bearer-capability clear-channel udi** [**bidirectional**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters service SIP configuration mode.

	Command or Action	Purpose
Step 5	**bearer-capability clear-channel udi** [**bidirectional**] **Example:** `Router(conf-serv-sip)# bearer-capability clear-channel udi bidirectional`	Enables clear-channel codec to UDI bearer capability mapping and UDI bearer capability to clear-channel codec mapping.
Step 6	**end** **Example:** `Router(conf-serv-sip)# end`	Exits service SIP configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

The following commands can help troubleshoot the ISDN UDI to SIP Clear-Channel feature:

- **debug ccsip all**
- **debug isdn q931**
- **show voice call summary**
- **show call active voice compact**
- **show voip rtp connections**
- **show isdn status**

Configuration Examples for SIP ISDN Support Features

ISDN Calling Name Display Examples

Note IP addresses and hostnames in examples are fictitious.

```
Router# show running-config
Building configuration...
Current configuration : 3845 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
resource-pool disable
```

```
clock timezone GMT 5
clock summer-time GMT recurring
!
no aaa new-model
ip subnet-zero
ip tcp path-mtu-discovery
ip name-server 172.18.192.48
!
isdn switch-type primary-ni
isdn voice-call-failure 0
isdn alert-end-to-end
!
voice call send-alert
!
voice service voip
 signaling forward unconditional
 sip
!
fax interface-type fax-mail
!
controller T1 0
 framing esf
 crc-threshold 0
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
 description lucent_pbx
!
controller T1 1
 shutdown
 framing esf
 crc-threshold 0
 linecode ami
 description summa_pbx
!
controller T1 2
 shutdown
 framing esf
 crc-threshold 0
 linecode ami
!
controller T1 3
 framing esf
 crc-threshold 0
 clock source line secondary 1
 linecode b8zs
 pri-group timeslots 1-24
!
translation-rule 100
 Rule 1 ^1 1 ANY national
 Rule 2 2% 2 ANY unknown
 Rule 4 4% 4 ANY unknown
 Rule 5 5% 5 ANY unknown
 Rule 6 6% 6 ANY unknown
 Rule 7 7% 7 ANY unknown
 Rule 8 8% 8 ANY unknown
 Rule 9 9% 9 ANY unknown
!
interface Ethernet0
 ip address 172.18.193.100 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 ip rsvp bandwidth 1 1
!
```

```
interface Serial0:23
 no ip address
 isdn switch-type primary-ni
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn supp-service name calling
 isdn disconnect-cause 1
 fair-queue 64 256 0
 no cdp enable
!
interface Serial3:23
 no ip address
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn supp-service name calling
 isdn T310 30000
 isdn disconnect-cause 1
 isdn bchan-number-order descending
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 10.1.1.2 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.193.1
ip route 0.0.0.0 0.0.0.0 172.18.193.129
ip route 0.0.0.0 0.0.0.0 172.18.207.129
ip route 0.0.0.0 0.0.0.0 172.18.16.129
ip route 0.0.0.0 0.0.0.0 Ethernet0
ip route 0.0.0.0 0.0.0.0 172.18.197.1
ip route 0.0.0.0 255.255.255.0 Ethernet0
ip route 10.2.0.1 255.255.255.255 172.18.16.135
ip route 172.18.0.0 255.255.0.0 Ethernet0
no ip http server
!
map-class dialer test
 dialer voice-call
dialer-list 1 protocol ip permit
!
control-plane
!
voice-port 0:D
!
dial-peer voice 10 pots
 application session.t.old
 destination-pattern 5550100
 prefix 5550100
!
dial-peer voice 4 voip
 application session
 destination-pattern 5550120
 session protocol sipv2
 session target ipv4:172.18.193.99
 incoming called-number 5550125
!
dial-peer voice 1 pots
 application session
```

```
 destination-pattern 5550125
 incoming called-number 5550155
 port 0:D
 prefix 95550125
!
dial-peer voice 18 voip
 application session
 destination-pattern 36601
 session protocol sipv2
 session target ipv4:172.18.193.187
 codec g711ulaw
!
dial-peer voice 25 voip
 destination-pattern 5550155
 session protocol sipv2
 session target ipv4:172.18.192.232
!
dial-peer voice 5678 pots
 destination-pattern 5678
 port 3:D
 prefix 5678
!
dial-peer voice 56781 voip
 incoming called-number 5678
!
sip-ua
!
line con 0
line aux 0
line vty 0 4
 password password1
 login
!
```

end

Signal ISDN B-Channel ID to Enable Application Control of Voice Gateway Trunks Example

```
Router# show running-config
Building configuration...
Current configuration : 3394 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 h323
  billing b-channel
 sip
  ds0-num
ip dhcp pool vespa
 network 192.168.0.0 255.255.255.0
```

```
 option 150 ip 192.168.0.1
 default-router 192.168.0.1
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 10.8.17.22 255.255.0.0
 half-duplex
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 10.8.15.4 1718
!
router rip
 network 10.0.0.0
 network 192.168.0.0
!
ip default-gateway 10.8.0.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.8.0.1
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
```

```
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:10.8.17.40 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP Carrier Identification Code Examples

CIC Parameter in SIP URL

This configuration example shows support for the CIC parameter in the user information part of the SIP URL. A SIP URL identifies a user's address and appears similar to an e-mail address, such as user@host, where *user* is the telephone number and *host* is either a domain name or a numeric network address. For example, the request line of an outgoing INVITE request might appear as:

```
INVITE sip:+5550100;cic=+16789@example.com;user=phone SIP/2.0
```

Where *+5550100; cic=+16789* signifies the user information, example.c*om* the domain name, and the *user=phone* parameter distinguishes that the user address is a telephone number rather than a username.

CIC Parameter in TEL URL

This configuration example shows support for the CIC parameter in the telephone-subscriber part of the TEL URL. A TEL URL takes the basic form of *tel:telephone subscriber number* , where *tel* requests the local entity to place a voice call, and *telephone subscriber number* is the number to receive the call. For example:

```
tel:+5550100;cic=+16789
```

The additional CIC parameter can be in any of the following three formats:

```
cic=+16789
cic=+1-6789
cic=6789
```

CIC Parameter and Visual Separators

This configuration example shows support for the CIC parameter in different formats --with and without visual separators. However, the CIC parameter usually has no visual separators. All of the following formats are accepted:

```
+12345
cic=+12345
cic=2345
```

Copying the CIC Parameter into the Resulting INVITE Request

This configuration example shows that the CIC parameter can be copied from the user information part of a 3xx Contact SIP URL into the resulting INVITE request.

For example, if a 302 REDIRECT response from a proxy appears like:

```
Contact: <sip:+5550100;cic=+16789@example.com;user=phone>
```

or like:

```
Contact: <sip:+5550100;cic=6789@example.com;user=phone>
```

The result is an INVITE request that sends the CIC with a +1 prefixed to it.

```
INVITE sip:+5550100;cic=+16789@example.com;user=phone SIP/2.0
```

SIP CLI for Caller ID When Privacy Exists Examples

The following shows an example of the SIP: CLI for Caller ID When Privacy Exists feature when enabled globally and disabled on the dial-peer level:

```
Router# show running-config
Building configuration...
Current configuration: 1234 bytes
!
version 12.4
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
!
hostname pip
!
boot-start-marker
boot system tftp user1/c3660-is-mz 172.18.207.15
boot-end-marker
!
logging buffered 1000000 debugging
enable secret 5 $1$li0u$IkIqPXzKq4uKme.LhzGut0
enable password password1
!
no aaa new-model
!
resource policy
!
clock timezone GMT 0
clock summer-time EDT recurring
ip subnet-zero
```

```
ip tcp path-mtu-discovery
!
ip cef
ip domain name example.sip.com
ip host sip-server1 172.18.193.100
ip host CALLGEN-SECURITY-V2 10.76.47.38 10.30.0.0
ip name-server 172.18.192.48
no ip dhcp use vrf connected
!
ip vrf btknet
rd 8262:2000
!
voice call send-alert
!
voice service voip <-  SIP: CLI for Caller ID When Privacy Exists feature enabled globally
clid substitute name
clid strip pi-restrict all
clid network-provided
sip
!
voice class codec 1
codec preference 1 g729r8
codec preference 2 g711alaw
codec preference 3 g711ulaw
codec preference 4 g729br8
codec preference 5 g726r32
codec preference 6 g726r24
codec preference 7 g726r16
codec preference 8 g723ar53
codec preference 9 g723r53
codec preference 10 g723ar63
codec preference 11 gsmefr
codec preference 12 gsmfr
codec preference 13 g728
!
voice class codec 2
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
voice class codec 99
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
fax interface-type fax-mail
!
interface FastEthernet0/0
ip address 172.18.195.49 255.255.255.0
duplex auto
speed auto
no cdp enable
ip rsvp bandwidth 96 96
!
interface FastEthernet0/1
ip address 172.18.193.190 255.255.255.0
shutdown
duplex auto
speed auto
no cdp enable
!
no ip http server
!
ip classless
```

```
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 172.16.0.0 255.0.0.0 172.18.195.1
!
snmp-server community public RO
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
mgcp behavior rsip-range tgcp-only
!
dial-peer cor custom
!
dial-peer voice 100 pots
destination-pattern 9001
!
dial-peer voice 3301 voip
destination-pattern 9002
session protocol sipv2
session target ipv4:172.18.193.87
incoming called-number 9001
codec g711ulaw
no vad
!
dial-peer voice 3303 voip
destination-pattern 777
session protocol sipv2
session target ipv4:172.18.199.94
!
dial-peer voice 36601 voip
destination-pattern 36601
no modem passthrough
session protocol sipv2
session target ipv4:172.18.193.98
!
dial-peer voice 5 voip
destination-pattern 5550100
session protocol sipv2
session target ipv4:172.18.197.182
codec g711ulaw
!
dial-peer voice 36602 voip
destination-pattern 36602
session protocol sipv2
session target ipv4:172.18.193.120
incoming called-number 9001
dtmf-relay rtp-nte
codec g711ulaw
!
dial-peer voice 111 voip
destination-pattern 111
session protocol sipv2
session target ipv4:172.18.193.251
!
dial-peer voice 5550199 voip <- SIP: CLI for Caller ID When Privacy Exists feature disabled
 on dial-peer
destination-pattern 3100801
session protocol sipv2
session target ipv4:10.102.17.208
codec g711ulaw
!
dial-peer voice 333 voip
```

```
preference 2
destination-pattern 333
modem passthrough nse codec g711ulaw
voice-class codec 99
session protocol sipv2
session target ipv4:172.18.193.250
dtmf-relay rtp-nte
no vad
!
dial-peer voice 9003 pots
preference 2
destination-pattern 9003
!
dial-peer voice 90032 voip
preference 1
destination-pattern 9003
session protocol sipv2
session target ipv4:172.18.193.97
!
dial-peer voice 1 pots
!
num-exp 5550100 5550199
num-exp 5550199 5550100
gateway
timer receive-rtp 1200
!
sip-ua
srv version 1
retry response 1
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
exec-timeout 0 0
password password1
login
!
no process cpu extended
no process cpu autoprofile hog
ntp clock-period 17180176
ntp server 192.0.10.150 prefer
!
end
```

The following shows an example of the SIP: CLI for Caller ID When Privacy Exists feature when disabled globally and disabled on the dial-peer level:

```
Router# show running-config
Building configuration...
Current configuration: 1234 bytes
!
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
!
hostname pip
!
boot-start-marker
boot system tftp user1/c3660-is-mz 172.18.207.15
boot-end-marker
!
logging buffered 1000000 debugging
enable secret 5 $1$li0u$IkIqPXzKq4uKme.LhzGut0
enable password password1
!
```

```
no aaa new-model
!
resource policy
!
clock timezone GMT 0
clock summer-time EDT recurring
ip subnet-zero
ip tcp path-mtu-discovery
!
ip cef
ip domain name example.sip.com
ip host sip-server1 172.18.193.100
ip host CALLGEN-SECURITY-V2 10.76.47.38 10.30.0.0
ip name-server 172.18.192.48
no ip dhcp use vrf connected
!
ip vrf btknet
rd 8262:2000
!
voice call send-alert
!
voice service voip <-  SIP: CLI for Caller ID When Privacy Exists feature disabled globally
sip
!
voice class codec 1
codec preference 1 g729r8
codec preference 2 g711alaw
codec preference 3 g711ulaw
codec preference 4 g729br8
codec preference 5 g726r32
codec preference 6 g726r24
codec preference 7 g726r16
codec preference 8 g723ar53
codec preference 9 g723r53
codec preference 10 g723ar63
codec preference 11 gsmefr
codec preference 12 gsmfr
codec preference 13 g728
!
voice class codec 2
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
voice class codec 99
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
fax interface-type fax-mail
!
interface FastEthernet0/0
ip address 172.18.195.49 255.255.255.0
duplex auto
speed auto
no cdp enable
ip rsvp bandwidth 96 96
!
interface FastEthernet0/1
ip address 172.18.193.190 255.255.255.0
shutdown
duplex auto
speed auto
no cdp enable
```

```
!
no ip http server
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 172.16.0.0 255.0.0.0 172.18.195.1
!
snmp-server community public RO
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
mgcp behavior rsip-range tgcp-only
!
dial-peer cor custom
!
dial-peer voice 100 pots
destination-pattern 9001
!
dial-peer voice 3301 voip
destination-pattern 9002
session protocol sipv2
session target ipv4:172.18.193.87
incoming called-number 9001
codec g711ulaw
no vad
!
dial-peer voice 3303 voip
destination-pattern 777
session protocol sipv2
session target ipv4:172.18.199.94
!
dial-peer voice 36601 voip
destination-pattern 36601
no modem passthrough
session protocol sipv2
session target ipv4:172.18.193.98
!
dial-peer voice 5 voip
destination-pattern 5550100
session protocol sipv2
session target ipv4:172.18.197.182
codec g711ulaw
!
dial-peer voice 36602 voip
destination-pattern 36602
session protocol sipv2
session target ipv4:172.18.193.120
incoming called-number 9001
dtmf-relay rtp-nte
codec g711ulaw
!
dial-peer voice 111 voip
destination-pattern 111
session protocol sipv2
session target ipv4:172.18.193.251
!
dial-peer voice 5550199 voip <- SIP: CLI for Caller ID When Privacy Exists feature disabled
 on dial-peer
destination-pattern 5550199
session protocol sipv2
```

```
session target ipv4:10.102.17.208
codec g711ulaw
!
dial-peer voice 333 voip
preference 2
destination-pattern 333
modem passthrough nse codec g711ulaw
voice-class codec 99
session protocol sipv2
session target ipv4:172.18.193.250
dtmf-relay rtp-nte
no vad
!
dial-peer voice 9003 pots
preference 2
destination-pattern 9003
!
dial-peer voice 90032 voip
preference 1
destination-pattern 9003
session protocol sipv2
session target ipv4:172.18.193.97
!
dial-peer voice 1 pots
!
num-exp 5550100 5550199
num-exp 5550101 5550198
gateway
timer receive-rtp 1200
!
sip-ua
srv version 1
retry response 1
!
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
exec-timeout 0 0
password password1
login
!
no process cpu extended
no process cpu autoprofile hog
ntp clock-period 17180176
ntp server 192.0.10.150 prefer
!
end
```

The following shows an example of the SIP: CLI for Caller ID When Privacy Exists feature when disabled globally and enabled on the dial-peer level:

```
Router# show running-config
Building configuration...
Current configuration: 1234 bytes
!
version 12.4
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
!
hostname pip
!
```

```
boot-start-marker
boot system tftp judyg/c3660-is-mz 172.18.207.15
boot-end-marker
!
logging buffered 1000000 debugging
enable secret 5 $1$li0u$IkIqPXzKq4uKme.LhzGut0
enable password password1
!
no aaa new-model
!
resource policy
!
clock timezone GMT 0
clock summer-time EDT recurring
ip subnet-zero
ip tcp path-mtu-discovery
!
ip cef
ip domain name example.sip.com
ip host sip-server1 172.18.193.100
ip host CALLGEN-SECURITY-V2 10.76.47.38 10.30.0.0
ip name-server 172.18.192.48
no ip dhcp use vrf connected
!
ip vrf btknet
rd 8262:2000
!
voice call send-alert
!
voice service voip <-  SIP: CLI for Caller ID When Privacy Exists feature disabled globally
sip
!
voice class codec 1
codec preference 1 g729r8
codec preference 2 g711alaw
codec preference 3 g711ulaw
codec preference 4 g729br8
codec preference 5 g726r32
codec preference 6 g726r24
codec preference 7 g726r16
codec preference 8 g723ar53
codec preference 9 g723r53
codec preference 10 g723ar63
codec preference 11 gsmefr
codec preference 12 gsmfr
codec preference 13 g728
!
voice class codec 2
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
voice class codec 99
codec preference 1 g729r8
codec preference 2 g711ulaw
codec preference 3 g711alaw
!
fax interface-type fax-mail
!
interface FastEthernet0/0
ip address 172.18.195.49 255.255.255.0
duplex auto
speed auto
no cdp enable
```

```
ip rsvp bandwidth 96 96
!
interface FastEthernet0/1
ip address 172.18.193.190 255.255.255.0
shutdown
duplex auto
speed auto
no cdp enable
!
no ip http server
!
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 172.16.0.0 255.0.0.0 172.18.195.1
!
snmp-server community public RO
!
control-plane
!
voice-port 1/0/0
!
voice-port 1/0/1
!
mgcp behavior rsip-range tgcp-only
!
dial-peer cor custom
!
dial-peer voice 100 pots
destination-pattern 9001
!
dial-peer voice 3301 voip
destination-pattern 9002
session protocol sipv2
session target ipv4:172.18.193.87
incoming called-number 9001
codec g711ulaw
no vad
!
dial-peer voice 3303 voip
destination-pattern 777
session protocol sipv2
session target ipv4:172.18.199.94
!
dial-peer voice 36601 voip
destination-pattern 36601
no modem passthrough
session protocol sipv2
session target ipv4:172.18.193.98
!
dial-peer voice 5 voip
destination-pattern 5550102
session protocol sipv2
session target ipv4:172.18.197.182
codec g711ulaw
!
dial-peer voice 36602 voip
destination-pattern 36602
session protocol sipv2
session target ipv4:172.18.193.120
incoming called-number 9001
dtmf-relay rtp-nte
codec g711ulaw
!
dial-peer voice 111 voip
```

```
destination-pattern 111
session protocol sipv2
session target ipv4:172.18.193.251
!
dial-peer voice 5550100 voip <- SIP: CLI for Caller ID When Privacy Exists feature enabled
 on dial-peer
destination-pattern 5550100
session protocol sipv2
session target ipv4:10.102.17.208
codec g711ulaw
clid strip pi-restrict all
clid network-provided
clid substitute name
!
dial-peer voice 333 voip
preference 2
destination-pattern 333
modem passthrough nse codec g711ulaw
voice-class codec 99
session protocol sipv2
session target ipv4:172.18.193.250
dtmf-relay rtp-nte
no vad
!
dial-peer voice 9003 pots
preference 2
destination-pattern 9003
!
dial-peer voice 90032 voip
preference 1
destination-pattern 9003
session protocol sipv2
session target ipv4:172.18.193.97
!
dial-peer voice 1 pots
!
num-exp 5550100 5550199
num-exp 5550101 5550198
gateway
timer receive-rtp 1200
!
sip-ua
srv version 1
retry response 1
!
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
exec-timeout 0 0
password password1
login
!
no process cpu extended
no process cpu autoprofile hog
ntp clock-period 17180176
ntp server 192.0.10.150 prefer
!
end
```

SIP ISDN Suspend Resume Support Example

The following example shows SIP Suspend and Resume disabled on the gateway (SIP Suspend and Resume is enabled by default on the gateway).

Note IP addresses and hostnames in examples are fictitious.

```
Router# show running-config
Building configuration...
Current configuration : 3845 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
boot-start-marker
boot-end-marker
!
no logging buffered
!
resource-pool disable
clock timezone GMT 5
clock summer-time GMT recurring
!
no aaa new-model
ip subnet-zero
ip tcp path-mtu-discovery
ip name-server 172.18.192.48
!
isdn switch-type primary-ni
isdn voice-call-failure 0
isdn alert-end-to-end
!
voice call send-alert
!
voice service voip
 signaling forward unconditional
 sip
!
fax interface-type fax-mail
!
controller T1 0
 framing esf
 crc-threshold 0
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
 description lucent_pbx
!
controller T1 1
 shutdown
 framing esf
 crc-threshold 0
 linecode ami
 description summa_pbx
!
controller T1 2
 shutdown
```

```
 framing esf
 crc-threshold 0
 linecode ami
!
controller T1 3
 framing esf
 crc-threshold 0
 clock source line secondary 1
 linecode b8zs
 pri-group timeslots 1-24
!
translation-rule 100
 Rule 1 ^1 1 ANY national
 Rule 2 2% 2 ANY unknown
 Rule 4 4% 4 ANY unknown
 Rule 5 5% 5 ANY unknown
 Rule 6 6% 6 ANY unknown
 Rule 7 7% 7 ANY unknown
 Rule 8 8% 8 ANY unknown
 Rule 9 9% 9 ANY unknown
!
interface Ethernet0
 ip address 172.18.193.100 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 ip rsvp bandwidth 1 1
!
interface Serial0:23
 no ip address
 isdn switch-type primary-ni
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn supp-service name calling
 isdn disconnect-cause 1
 fair-queue 64 256 0
 no cdp enable
!
interface Serial3:23
 no ip address
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn supp-service name calling
 isdn T310 30000
 isdn disconnect-cause 1
 isdn bchan-number-order descending
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 10.1.1.2 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.193.1
ip route 0.0.0.0 0.0.0.0 172.18.193.129
ip route 0.0.0.0 0.0.0.0 172.18.207.129
ip route 0.0.0.0 0.0.0.0 172.18.16.129
ip route 0.0.0.0 0.0.0.0 Ethernet0
ip route 0.0.0.0 0.0.0.0 172.18.197.1
```

```
ip route 0.0.0.0 255.255.255.0 Ethernet0
ip route 10.2.0.1 255.255.255.255 172.18.16.135
ip route 172.18.0.0 255.255.0.0 Ethernet0
no ip http server
!
map-class dialer test
 dialer voice-call
dialer-list 1 protocol ip permit
!
control-plane
!
voice-port 0:D
!
dial-peer voice 10 pots
 application session.t.old
 destination-pattern 5550100
 prefix 5550100
!
dial-peer voice 4 voip
 application session
 destination-pattern 5550120
 session protocol sipv2
 session target ipv4:172.18.193.99
 incoming called-number 5550125
!
dial-peer voice 1 pots
 application session
 destination-pattern 5550125
 incoming called-number 5550155
 port 0:D
 prefix 95550125
!
dial-peer voice 18 voip
 application session
 destination-pattern 36601
 session protocol sipv2
 session target ipv4:172.18.193.187
 codec g711ulaw
!
dial-peer voice 25 voip
 destination-pattern 5550155
 session protocol sipv2
 session target ipv4:172.18.192.232
!
dial-peer voice 5678 pots
 destination-pattern 5678
 port 3:D
 prefix 5678
!
dial-peer voice 56781 voip
 incoming called-number 5678
!
sip-ua
 no suspend-resume
 retry invite 1
 retry bye 1
line con 0
line aux 0
line vty 0 4
 password password1
 login
!
```

end

SIP PSTN Transport Using the Cisco Generic Transparency Descriptor Examples

Configuring GTD Globally

The following examples shows that GTD is configured.

```
Router# show running-config
Building configuration...
Current configuration : 4192 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
hostname router
!
voice service voip
 signaling forward unconditional
 sip
 .
```

Configuring GTD for an Individual Dial Peer

The following example shows GTD configured with unconditional forwarding on two dial peers:

```
Router# show running-config
Building configuration...
Current configuration : 4169 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
hostname router
.
.
.
dial-peer voice 36 voip
 incoming called-number 3100802
 destination-pattern 3100801
 signaling forward unconditional
 session protocol sipv2
 session target ipv4:192.0.2.209
!
dial-peer voice 5 voip
 destination-pattern 5555555
```

```
 signaling forward unconditional
 session protocol sipv2
 session target ipv4:172.18.192.218
.
.
.
```

Example: Configuring the ISDN UDI to SIP Clear-Channel Feature

The following example shows how to configure the ISDN UDI to SIP Clear-Channel feature on an ISDN SIP gateway:

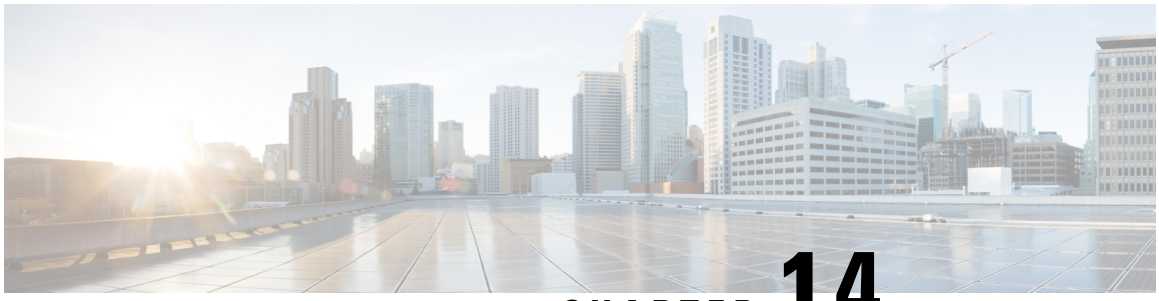
```
Router> enable
Router# configure terminal
Router(config)# voice service voip
Router(conf-voi-serv)# sip
Router(conf-serv-sip)# bearer-capability clear-channel udi bidirectional
Router(conf-serv-sip)# end
```

Additional References

General SIP References

References Mentioned in This Chapter (listed alphabetically)

- *Configuring H.323 Gateways* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/callc_c/h323_c/323confg/3gwconf.htm
- *Redundant Link Manager (RLM)* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/as5400/sw_conf/ios_121/pull_rlm.htm



CHAPTER 14

Configuring SIP DTMF Features

This chapter describes the following SIP features that support dual-tone multifrequency (DTMF) signaling:

- RFC 2833 DTMF Media Termination Point (MTP) Passthrough
- DTMF Events Through SIP Signaling
- DTMF Relay for SIP Calls Using Named Telephone Events
- SIP INFO Method for DTMF Tone Generation
- SIP NOTIFY-Based Out-of-Band DTMF Relay Support
- SIP KPML-Based Out-of-Band DTMF Relay Support
- SIP Support for Asymmetric SDP

Feature History for the RFC 2833 DTMF MTP Passthrough

Release	Modification
12.4(11)T	This feature was introduced.

Feature History for DTMF Events Through SIP Signaling

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for DTMF Relay for SIP Calls Using NTE

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB1	This feature was implemented on an additional platform.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Feature History for SIP INFO Method for DTMF Tone Generation

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for SIP NOTIFY-Based Out-of-Band DTMF Relay Support

Release	Modification
12.3(4)T	This feature was introduced.

Feature History for SIP KPML-Based Out-of-Band DTMF Relay Support

Release	Modification
12.4(9)T	This feature was introduced.

Feature History for the SIP Support for Asymmetric SDP

Release	Modification
12.4(15)T	This feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for SIP DTMF

RFC 2833 DTMF MTP Passthrough Feature

- The RFC 2833 DTMF MTP Passthrough feature adds support for passing Dual-Tone Multifrequency (DTMF) tones transparently between Session Initiation Protocol (SIP) endpoints that require either transcoding or use of the RSVP Agent feature. If the T38 Fax Relay feature is also configured on this IP network, configure the voice gateways to use a payload type other than PT97 or PT98 for fax relay negotiation, or depending on whether the SIP endpoints support different payload types, configure Cisco Unified CME to use a payload type other than PT97 or PT98 for DTMF.

DTMF Events Through SIP Signaling Feature

- The DTMF Events Through SIP Signaling feature adds support for sending telephone-event notifications via SIP NOTIFY messages from a SIP gateway. The events for which notifications are sent out are DTMF events from the local Plain Old Telephone Service (POTS) interface on the gateway. Notifications are not sent for DTMF events received in the Real-Time Transport Protocol (RTP) stream from the recipient user agent.

DTMF Relay for SIP Calls Using NTEs Feature

- The SIP NTE DTMF relay feature is available only for SIP calls on Cisco VoIP gateways. The SIP NTE DTMF relay feature supports only hookflash relay and does not support hookflash generation for advanced features such as call waiting and conferencing.

SIP INFO Method for DTMF Tone Generation Feature

- Minimum signal duration is 100 ms. If a request is received with a duration less than 100 ms, the minimum duration of 100 ms is used by default.
- Maximum signal duration is 5000 ms. If a request is received with a duration longer than 5000 ms, the maximum duration of 5000 ms is used by default.
- If no duration parameter is included in a request, the gateway defaults to a signal duration of 250 ms.

SIP NOTIFY-Based Out-of-Band DTMF Relay Support Feature

- To support Skinny Client Control Protocol (SCCP) IP phones, originating and terminating SIP gateways can use NOTIFY-based out-of-band DTMF relay. NOTIFY-based out-of-band DTMF relay is a Cisco proprietary function.
- You can configure support only on a SIP VoIP dial peer.

SIP KPML-Based Out-of-Band DTMF Relay Support Feature

- For incoming dial peers, if you configure multiple DTMF negotiation methods, the first value you configure takes precedence, then the second, and then the third.

- For incoming dial peers, the first out-of-band negotiation method takes precedence over other DTMF negotiation methods, except when the **dtmf-relay rtp-nte**command has precedence; in this case, the **dtmf-relay sip-kpml** command takes precedence over other out-of-band negotiation methods.
- For incoming dial peers, if both the **dtmf-relay rtp-nte** and **dtmf-relay sip-kpml** commands and notification mechanisms are enabled and negotiated, the gateway relies on RFC 2833 notification to receive digits and a SUBSCRIBE for KPML is not initiated.
- SIP KPML support complies to the IEFT draft “draft-ietf-sipping-kpml-04.txt” with the following limitations:
  - The SIP gateway always initiates SUBSCRIBE in the context of an established INVITE dialog. The gateway supports receiving SUBSCRIBE in the context of an established INVITE dialog, as well as out-of-call context requests with a leg parameter in the Event header. If the request code does not match an existing INVITE dialog, the gateway sends a NOTIFY with KPML status-code 481 and sets Subscription-State to terminated.
  - The gateway does not support the Globally Routable User Agent (GRUU) requirement. The Contact header in the INVITE/200 OK message is generated locally from the gateway’s contact information.
  - The gateway always initiates persistent subscriptions, but the gateway receives and processes persistent and one-shot subscriptions.
  - The gateway supports only single-digit reporting. There is no need for inter-digit timer support. The only regular expressions supported are those which match a single digit. For example:

<regex>x</regex>--Matches any digit 0 through 9

<regex>1</regex>--Matches digit 1

<regex>[x#*ABCD]</regex>--Matches to any digit 0 through 9, # (the pound sign), * (an asterisk), or A, B, C, or D

<regex>[24]</regex>--Matches digits 2 or 4

<regex>[2-9]</regex>--Matches on any digit 2 through 9

<regex>[^2-9]</regex>--Matches digits 0 or 1

- - The gateway does not support long key presses, which are detected and reported as a single digit press.
  - Digit suppression is not supported (pre tag for suppressing inband digits).
  - Individual stream selection is not supported. A SUBSCRIBE request for KPML applies to all audio streams in the dialog (stream element and reverse are not supported).
- You can configure support only on a SIP VoIP dial peer.
- In Cisco Unified Border Element (Cisco UBE), RTP-NTE to RTP-NTE DTMF interworking is not supported when you use High Density Voice Network Module (NM-HDV) for transcoding.

Prerequisites for SIP DTMF

DTMF Relay for SIP Calls Using NTEs Feature

- Ensure that you have a working VoIP network using SIP on Cisco gateways.

SIP INFO Method (sip-info)

This section describes the SIP INFO Method for DTMF Tone Generation feature, which uses the SIP INFO method to generate dual-tone multifrequency (DTMF) tones on the telephony call leg. SIP methods or request message types, request a specific action be taken by another user agent or proxy server. The SIP INFO message is sent along the signaling path of the call. With the feature, upon receipt of a SIP INFO message with DTMF relay content, the gateway generates the specified DTMF tone on the telephony end of the call.

The SIP INFO Method for DTMF Tone Generation feature is always enabled, and is invoked when a SIP INFO message is received with DTMF relay content. This feature is related to the SIP NOTIFY-Basec Out-of-Band DTMF Relay Support feature, which provides the ability for an application to be notified about DTMF events using SIP NOTIFY messages. Together, the two features provide a mechanism to both send and receive DTMF digits along the signaling path.

Note For information on sending DTMF event notification using SIP NOTIFY messages, see "DTMF Events Through SIP Signaling".

SIP INFO Messages

The SIP INFO method is used by a user agent to send call signaling information to another user agent with which it has an established media session. The following example shows a SIP INFO message with DTMF content:

```
INFO sip:2143302100@172.17.2.33 SIP/2.0
Via: SIP/2.0/UDP 172.80.2.100:5060
From:  <sip:9724401003@172.80.2.100>;tag=43
To:  <sip:2143302100@172.17.2.33>;tag=9753.0207
Call-ID: 984072_15401962@172.80.2.100
CSeq: 25634 INFO
Supported: 100rel
Supported: timer
Content-Length: 26
Content-Type: application/dtmf-relay
Signal= 1
Duration= 160
```

This sample message shows a SIP INFO message received by the gateway with specifics about the DTMF tone to be generated. The combination of the From, To, and Call-ID headers identifies the call leg. The signal and duration headers specify the digit, in this case 1, and duration, 160 milliseconds in the example, for DTMF tone play.

How to Review SIP INFO Messages

The SIP INFO method is used by a UA to send call signaling information to another UA with which it has an established media session. The following example shows a SIP INFO message with DTMF content:

```
INFO sip:2143302100@172.17.2.33 SIP/2.0
Via: SIP/2.0/UDP 172.80.2.100:5060
From:  <sip:9724401003@172.80.2.100>;tag=43
To:  <sip:2143302100@172.17.2.33>;tag=9753.0207
```

```
Call-ID: 984072_15401962@172.80.2.100
CSeq: 25634 INFO
Supported: 100rel
Supported: timer
Content-Length: 26
Content-Type: application/dtmf-relay
Signal= 1
Duration= 160
```

This sample message shows a SIP INFO message received by the gateway with specifics about the DTMF tone to be generated. The combination of the "From", "To", and "Call-ID" headers identifies the call leg. The signal and duration headers specify the digit, in this case 1, and duration, 160 milliseconds in the example, for DTMF tone play.

Configuring SIP INFO Method for DTMF Tone Generation

You cannot configure, enable, or disable this feature. You can display SIP statistics, including SIP INFO method statistics, by using the **show sip-ua statistics** and **show sip-ua status** commands in privileged EXEC mode. See the following fields for SIP INFO method statistics:

- OkInfo 0/0, under SIP Response Statistics, Success, displays the number of successful responses to an INFO request.
- Info 0/0, under SIP Total Traffic Statistics, displays the number of INFO messages received and sent by the gateway.

Note To see sample output of these **show** commands, see "Configuring Passthrough on a Gateway that Connects to an MTP or Transcoder Gateway".

- To reset the counters for the **sip-ua statistics**command, use the **clear sip-ua statistics** command.

RTP-NTE Method (rtp-nte)

In-band RFC2833 NTE payload types and attributes are negotiated between the two ends at call setup using the Session Description Protocol (SDP) within the body section of the SIP message.

Feature benefits include the following:

- Reliable DTMF digit relay between Cisco VoIP gateways when low-bandwidth codecs are used
- Ability to communicate with SIP phone software that uses NTE packets to indicate DTMF digits

Reliable DTMF Relay

The SIP NTE DTMF relay feature provides reliable digit relay between Cisco VoIP gateways when a low-bandwidth codec is used. Using NTE to relay DTMF tones provides a standardized means of transporting DTMF tones in Real-Time Transport Protocol (RTP) packets according to section 3 of RFC 2833, *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals* , developed by the Internet Engineering Task Force (IETF) Audio/Video Transport (AVT) working group. RFC 2833 defines formats of NTE RTP packets used to transport DTMF digits, hookflash, and other telephony events between two peer endpoints.

DTMF tones are generated when a button on a touch-tone phone is pressed. When the tone is generated, it is compressed, transported to the other party, and decompressed. If a low-bandwidth codec, such as a G.729 or G.723 is used without a DTMF relay method, the tone may be distorted during compression and decompression.

With the SIP NTE DTMF relay feature, the endpoints perform per-call negotiation of the DTMF relay method. They also negotiate to determine the payload type value for the NTE RTP packets.

In a SIP call, the gateway forms a Session Description Protocol (SDP) message that indicates the following:

- If NTE will be used
- Which events will be sent using NTE
- NTE payload type value

The SIP NTE DTMF relay feature can relay hookflash events in the RTP stream using NTP packets.

Note The SIP NTE DTMF relay feature does not support hookflash generation for advanced features such as call waiting and conferencing.

SIP IP Phone Support

The SIP NTE DTMF relay feature adds SIP phone support. When SIP IP phones are running software that does not have the capability to generate DTMF tones, the phones use NTE packets to indicate DTMF digits. With the SIP NTE DTMF relay feature, Cisco VoIP gateways can communicate with SIP phones that use NTE packets to indicate DTMF digits. The Cisco VoIP gateways can relay the digits to other endpoints.

RFC 2833 DTMF MTP Passthrough

RFC 2833 DTMF MTP Passthrough is configured on a gateway that does not itself contain an MTP or transcoder but connects to another gateway that does. The RFC 2833 DTMF Media Termination Point (MTP) Passthrough feature passes DTMF tones transparently between Session Initiation Protocol (SIP) endpoints that require either transcoding or use of the RSVP Agent feature. (An RSVP agent is a Cisco IOS-based Resource Reservation Protocol [RSVP] proxy server that registers with the call manager--Cisco Unified CallManager or Cisco Unified CallManager Express--as a media-termination point or a transcoder device.)

The MTP or transcoding module on a gateway detects RFC 2833 (DTMF) packets from an IP endpoint. You can configure whether it should do either or both of the following:

- Generate and send an out-of-band signal event to the call manager
- Pass the packets through to the other IP endpoint (default)

You can configure this instruction from the call manager, from the gateway, or both. The gateway can itself contain a call manager with an MTP or transcoder, or it can connect to another gateway that contains a call manager with an MTP or transcoder. Configuration on the call manager takes precedence over configuration on the gateway.

You can specify that the gateway should relay DTMF tones between telephony interfaces and an IP network by using RTP with the Named Telephone Event (NTE) payload type using the **dtmf-relay rtp-nte** command.

Configure DTMF Relay for SIP Calls Using NTEs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *number* **voip**
4. **session protocol sipv2**
5. **dtmf-relay rtp-nte**
6. **rtp payload-type nte** *number* **comfort-noise** [**13** | **19**]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *number* **voip** **Example:** `Router(config)# dial-peer voice 10 voip`	Enters dial-peer VoIP configuration mode for the specified dial peer.
Step 4	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Specifies a session protocol for calls between local and remote routers using the packet network. The keyword is as follows: • **sipv2** --Dial peer uses the IETF SIP. Use this keyword with the SIP option.
Step 5	**dtmf-relay rtp-nte** **Example:** `Router(config-dial-peer)# dtmf-relay rtp-nte`	Specifies how an H.323 or SIP gateway relays DTMF tones between telephone interfaces and an IP network. The keyword is as follows: • **rtp-nte** --Forwards tones by using RTP with the NTE payload type.
Step 6	**rtp payload-type nte** *number* **comfort-noise** [**13** | **19**] **Example:**	Identifies the payload type of a RTP packet. Keywords and arguments are as follows: • **nte** *number* --Named telephone event (NTE). Range: 96 to 127. Default: 101.

	Command or Action	Purpose
	`Router(config-dial-peer)# rtp payload-type nte 100 comfort-noise 13`	• **comfort-noise** --RTP payload type of comfort noise. If you are connected to a gateway that complies with the RTP Payload for Comfort Noise July 2001 draft, use 13. If you are connected to an older Cisco gateway that uses DSPware before version 3.4.32, use 19.
Step 7	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

DTMF Relay for SIP Calls Using NTEs Examples

DTMF Relay using RTP-NTE

The following is an example of DTMF relay using RTP-NTE:

```
Router(config)# dial-peer voice 62 voip
Router(config-dial-peer)# session protocol sipv2
Router(config-dial-peer)# dtmf-relay rtp-nte
```

RTP Using Payload Type NTE

The following is an example of RTP Using Payload Type NTE with the default value of 101:

```
Router(config)# dial-peer voice 62 voip
Router(config-dial-peer)# rtp payload-type nte 101
```

SIP NOTIFY-Based Out-of-Band Method (sip-notify)

SCCP IP phones do not support in-band DTMF digits; they are capable of sending only out-of-band DTMF digits. To support SCCP devices, originating and terminating SIP gateways can use Cisco proprietary NOTIFY-based out-of-band DTMF relay. In addition, NOTIFY-based out-of-band DTMF relay can also be used by analog phones attached to analog voice ports (FXS) on the router.

NOTIFY-based out-of-band DTMF relay sends messages bidirectionally between the originating and terminating gateways for a DTMF event during a call. If multiple DTMF relay mechanisms are enabled on a SIP dial peer and are negotiated successfully, NOTIFY-based out-of-band DTMF relay takes precedence.

The originating gateway sends an Invite message with a SIP Call-Info header to indicate the use of NOTIFY-based out-of-band DTMF relay. The terminating gateway acknowledges the message with an 18x or 200 Response message, also using the Call-Info header. The Call-Info header for NOTIFY-based out-of-band relay appears as follows:

```
Call-Info: <sip: address>; method="NOTIFY;Event=telephone-event;Duration=msec"
```

Note Duration is the interval between NOTIFY messages sent for a single digit and is set by means of the **notify telephone-event** command.

First, the NOTIFY-based out-of-band DTMF relay mechanism is negotiated by the SIP Invite and 18x/200 Response messages. Then, when a DTMF event occurs, the gateway sends a SIP NOTIFY message for that event. In response, the gateway expects to receive a 200 OK message.

The NOTIFY-based out-of-band DTMF relay mechanism is similar to the DTMF message format described in RFC 2833. NOTIFY-based out-of-band DTMF relay consists of 4 bytes in a binary encoded format. The message format is shown in the figure below; field descriptions are listed in the table below.

Figure 85: Message Format of NOTIFY-Based Out-of-Band DTMF Relay

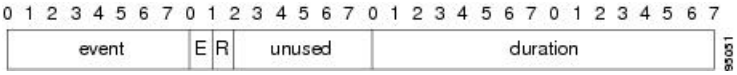


Table 51: Fields in NOTIFY-based out-of-band DTMF relay Message

Field	Description
event	The DTMF event that is between 0-9, A, B, C, D, #, * and flash.
E	E signifies the end bit. If E is set to a value of 1, the NOTIFY message contains the end of the DTMF event. Thus, the duration parameter in this final NOTIFY message measures the complete duration of the event.
R	Reserved.
unused	In RFC 2833, unused corresponds to the volume field, but is not used in NOTIFY-based out-of-band DTMF relay.
duration	Duration of this DTMF event, in milliseconds.

Sending NOTIFY Messages

As soon as the DTMF event is recognized, the gateway sends out an initial NOTIFY message for this event with the duration negotiated in the Invite's Call-Info header. For the initial NOTIFY message, the end bit is set to zero. Afterward, one of the following actions can occur:

- If the duration of the DTMF event is less than the negotiated duration, the originating gateway sends an end NOTIFY message for this event with the duration field containing the exact duration of the event and the end bit set to 1.
- If the duration of the DTMF event is greater than the negotiated duration, the originating gateway sends another NOTIFY message for this event after the initial timer runs out. The updated NOTIFY message has a duration of twice the negotiated duration. The end bit is set to 0 because the event is not yet over. If the event lasts beyond the duration specified in the first updated NOTIFY message, another updated NOTIFY message is sent with three times the negotiated duration.
- If the duration of the DTMF event is exactly the negotiated duration, either of the above two actions occurs, depending on whether the end of the DTMF event occurred before or after the timer ran out.

For example, if the negotiated duration is 600 ms, as soon as a DTMF event occurs, the initial NOTIFY message is sent with duration as 600 ms. Then a timer starts for this duration.

- If the DTMF event lasts only 300 ms, the timer stops and an end NOTIFY message is sent with the duration as 300 ms.
- If the DTMF event lasts longer than 600 ms (1000 ms), when the timer expires an updated NOTIFY message is sent with the duration as 1200 ms and the timer restarts. When the DTMF event ends, an end NOTIFY message is sent with the duration set to 1000 ms.

Every DTMF event corresponds to at least two NOTIFYs: an initial NOTIFY message and an end NOTIFY message. There might also be some update NOTIFYs involved, if the total duration of the event is greater than the negotiated max-duration interval. Because DTMF events generally last for less than 1000 ms, setting the duration using **notify telephone-event** command to more than 1000 ms reduces the total number of NOTIFY messages sent. The default value of **notify telephone-event** command is 2000 ms.

Receiving NOTIFY Messages

Once a NOTIFY message is received by the terminating gateway, the DTMF tone plays and a timer is set for the value in the duration field. Afterward, one of the following actions can occur:

- If an end NOTIFY message for a DTMF event is received, the tone stops.
- If an update is received, the timer is updated according to the duration field.
- If an update or end NOTIFY message is not received before the timer expires, the tone stops and all subsequent NOTIFY messages for the same DTMF event or DTMF digit are ignored until an end NOTIFY message is received.
- If a NOTIFY message for a different DTMF event is received before an end NOTIFY message for the current DTMF event is received (which is an unlikely case), the current tone stops and the new tone plays. This is an unlikely case because for every DTMF event there needs to be an end NOTIFY message, and unless this is successfully sent and a 200 OK is received, the gateway cannot send other NOTIFY messages.

Note In-band tones are not passed while NOTIFY-based out-of-band DTMF relay is used as the DTMF relay method.

Two commands allow you to enable or disable NOTIFY-based out-of-band DTMF relay on a dial peer. The functionality is advertised to the other end using Invite messages if it is enabled by the commands, and must be configured on both the originating and terminating SIP gateways. A third command allows you to verify DTMF relay status.

- **dtmf-relay (VoIP)**
- **notify telephone-event**
- **show sip-ua status**

Configuring SIP NOTIFY-Based Out-of-Band DTMF Relay

Note Cisco proprietary NOTIFY-based out-of-band DTMF relay adds support for devices that do not support in-band DTMF. This configuration must be done on both originating and terminating gateways. With this configuration, DTMF tones are forwarded by using SIP NOTIFY messages in SIP Invites or 18x or 200 Response messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay sip-notify**
5. **exit**
6. **sip-ua**
7. **notify telephone-event max-duration** *time*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the designated dial peer.
Step 4	**dtmf-relay sip-notify** **Example:** `Router(config-dial-peer)# dtmf-relay sip-notify`	Forwards DTMF tones using SIP NOTIFY messages.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

	Command or Action	Purpose
Step 6	**sip-ua** **Example:** `Router (config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 7	**notify telephone-event max-duration** *time* **Example:** `Router(config-sip-ua)# notify telephone-event max-duration 2000`	Sets the maximum time interval allowed between two consecutive NOTIFY messages for a single DTMF event. Keyword and argument are as follows: • **max-duration** *time* --Time, in ms, between consecutive NOTIFY messages for a single DTMF event. Range: 500 to 3000. Default: 2000.
Step 8	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

SIP NOTIFY-Based Out-of-Band DTMF Relay Example

```
Current configuration : 3394 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
memory-size iomem 15
ip subnet-zero
!
no ip domain lookup
!
voice service voip
 redirect ip2ip
sip
 redirect contact order best-match
ip dhcp pool vespa
 network 192.0.2.0 255.255.255.0
 option 150 ip 192.0.2.2
 default-router 192.0.2.3
!
voice call carrier capacity active
!
voice class codec 1
 codec preference 2 g711ulaw
!
no voice hpi capture buffer
no voice hpi capture destination
!
fax interface-type fax-mail
mta receive maximum-recipients 0
!
interface Ethernet0/0
 ip address 192.0.2.4 255.255.0.0
```

```
 half-duplex
!
interface FastEthernet0/0
 ip address 192.0.2.5 255.255.255.0
 speed auto
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id vespa2 ipaddr 192.0.2.6
!
router rip
 network 192.0.2.0
 network 209.165.201.0
!
ip default-gateway 192.0.2.9
ip classless
ip route 0.0.0.0 0.0.0.0 192.0.2.10
no ip http server
ip pim bidir-enable
!
tftp-server flash:SEPDEFAULT.cnf
tftp-server flash:P005B302.bin
call fallback active
!
call application global default.new
call rsvp-sync
!
voice-port 1/0
!
voice-port 1/1
!
mgcp profile default
!
dial-peer voice 1 pots
 destination-pattern 5100
 port 1/0
!
dial-peer voice 2 pots
 destination-pattern 9998
 port 1/1
!
dial-peer voice 123 voip
 destination-pattern [12]...
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
!
gateway
!
sip-ua
 retry invite 3
 retry register 3
 timers register 150
 registrar dns:myhost3.example.com expires 3600
 registrar ipv4:192.0.2.11 expires 3600 secondary
!
telephony-service
 max-dn 10
 max-conferences 4
!
ephone-dn 1
number 4001
!
ephone-dn 2
number 4002
```

```
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
login
line vty 5 15
 login
!
no scheduler allocate
end
```

SIP KPML-Based Out-of-Band Method (sip-kpml)

KPML support is required on SIP gateways for non-conferencing calls, and for interoperability between SIP products and SIP phones. If you configure KPML on the dial peer, the gateway sends INVITE messages with "kpml" in the Allow-Events header. Currently, all configured DTMF methods are recognized and sent in the outgoing INVITE. If you configure rtp-nte (RFC 2833), sip-notify, and sip-kpml, the outgoing INVITE contains a call-info header, an Allow-Events header with KPML, and an sdp with rtp-nte payload.

DTMF negotiation is performed based on the matching inbound dial-peer configuration. The gateway negotiates to either just cisco-rtp, just rtp-nte, rtp-nte + kpml, just kpml, or just sip-notify. If you configure more than one out-of-band DTMF method, preference goes from highest to lowest in the order they were configured. Whichever DTMF negotiation method you configure first takes precedence.

A gateway negotiates both rtp-nte and KPML if both are supported and advertised in the incoming INVITE. However, in this case, the gateway relies on the rtp-nte DTMF method to receive digits and a SUBSCRIBE for KPML is not initiated, however the gateway still accepts SUBSCRIBEs for KPML. This prevents double-digit reporting problems at the gateway.

The following example shows the INVITE and SUBSCRIBE sequence for KPML.

```
Sent:
INVITE sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>
Date: Fri, 01 Mar 2002 00:15:59 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Supported: 100rel,timer,resource-priority,replaces
Min-SE:  1800
Cisco-Guid: 1424162198-736104918-2148455531-3036263926
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Timestamp: 1014941759
Contact: <sip:172.18.193.251:5060>
Expires: 180
Allow-Events: kpml
, telephone-event
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221
v=0
o=CiscoSystemsSIP-GW-UserAgent 1438 8538 IN IP4 172.18.193.251
s=SIP Call
c=IN IP4 172.18.193.251
```

```
t=0 0
m=audio 17576 RTP/AVP 0 19
c=IN IP4 172.18.193.251
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20
//-1/xxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:34 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Timestamp: 1014941759
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
Require: 100rel
RSeq: 3482
Allow-Events: kpml
, telephone-event
Contact: <sip:8888@172.18.193.250:5060>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221
v=0
o=CiscoSystemsSIP-GW-UserAgent 9384 6237 IN IP4 172.18.193.250
s=SIP Call
c=IN IP4 172.18.193.250
t=0 0
m=audio 17468 RTP/AVP 0 19
c=IN IP4 172.18.193.250
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
a=ptime:20
//-1/xxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKC1ECC
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:38 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Timestamp: 1014941759
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
Allow-Events: kpml, telephone-event
Contact: <sip:8888@172.18.193.250:5060>
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 221
v=0
o=CiscoSystemsSIP-GW-UserAgent 9384 6237 IN IP4 172.18.193.250
s=SIP Call
c=IN IP4 172.18.193.250
t=0 0
m=audio 17468 RTP/AVP 0 19
c=IN IP4 172.18.193.250
a=rtpmap:0 PCMU/8000
a=rtpmap:19 CN/8000
```

```
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKEB8B
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 00:16:00 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
Max-Forwards: 70
CSeq: 101 ACK
Allow-Events: kpml, telephone-event
Content-Length: 0
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SUBSCRIBE sip:8888@172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKFF36
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 103 SUBSCRIBE
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:16:15 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Expires: 7200
Contact: <sip:172.18.193.251:5060>
Content-Type: application/kpml-request+xml
Content-Length: 327
<?xml version="1.0" encoding="UTF-8"?><kpml-request
xmlns="urn:ietf:params:xml:ns:kpml-request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:kpml-request kpml-request.xsd"
version="1.0"><pattern persist="persist"><regex
tag="dtmf">[x*#ABCD]</regex></pattern></kpml-request>
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SUBSCRIBE sip:172.18.193.251:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK5FE3
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 101 SUBSCRIBE
Max-Forwards: 70
Date: Fri, 01 Mar 2002 01:02:46 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Expires: 7200
Contact: <sip:172.18.193.250:5060>
Content-Type: application/kpml-request+xml
Content-Length: 327
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bKFF36
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:02:51 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 103 SUBSCRIBE
Content-Length: 0
Contact: <sip:172.18.193.250:5060>
Expires: 7200
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK5FE3
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Date: Fri, 01 Mar 2002 00:16:24 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 101 SUBSCRIBE
Content-Length: 0
Contact: <sip:172.18.193.251:5060>
Expires: 7200
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
NOTIFY sip:172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK101EA4
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 104 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:16:24 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Subscription-State: active
Contact: <sip:172.18.193.251:5060>
Content-Length: 0
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
NOTIFY sip:172.18.193.251:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK6111
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 102 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 01:02:51 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Subscription-State: active
Contact: <sip:172.18.193.250:5060>
Content-Length: 0
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.250:5060;branch=z9hG4bK6111
From: <sip:8888@172.18.193.250>;tag=39497C-2EA
To: <sip:172.18.193.251>;tag=EA330-F6
Date: Fri, 01 Mar 2002 00:16:32 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 102 NOTIFY
Content-Length: 0
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
NOTIFY sip:172.18.193.250:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK1117DE
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 105 NOTIFY
Max-Forwards: 70
Date: Fri, 01 Mar 2002 00:37:33 GMT
User-Agent: Cisco-SIPGateway/IOS-12.x
Event: kpml
Subscription-State: active
Contact: <sip:172.18.193.251:5060>
```

```
Content-Type: application/kpml-response+xml
Content-Length: 113
<?xml version="1.0" encoding="UTF-8"?><kpml-response version="1.0" code="200" text="OK"
digits="1" tag="dtmf"/>
/-1/xxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.18.193.251:5060;branch=z9hG4bK1117DE
From: <sip:172.18.193.251>;tag=EA330-F6
To: <sip:8888@172.18.193.250>;tag=39497C-2EA
Date: Fri, 01 Mar 2002 01:24:08 GMT
Call-ID: 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
CSeq: 105 NOTIFY
Content-Length: 0

...
```

SIP KPML-Based Out-of-Band DTMF Relay Example

```
router(config-dial-peer)# dtmf
router(config-dial-peer)# dtmf-relay ?
  cisco-rtp          Cisco Proprietary RTP
  h245-alphanumeric  DTMF Relay via H245 Alphanumeric IE
  h245-signal        DTMF Relay via H245 Signal IE
  rtp-nte            RTP Named Telephone Event RFC 2833
  sip-kpml           DTMF Relay via KPML over SIP SUBCRIBE/NOTIFY
  sip-notify         DTMF Relay via SIP NOTIFY messages
router(config-dial-peer)# dtmf-relay sip-kpml
router(config-dial-peer)# end
%SYS-5-CONFIG_I: Configured from console by console
router#sh run
Building configuration...Current configuration : 2430 bytes
!
version 12.4
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mahoney
!
boot-start-marker
boot-end-marker
!
logging buffered 5000000 debugging
!
no aaa new-model
!
resource policy
!
clock timezone EST 0
ip cef
ip name-server 192.0.2.21
ip name-server 192.0.2.22
!
voice-card 0
!
voice service voip
 sip
  min-se  90
  registrar server
!
```

```
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g729r8
 codec preference 3 g729br8
 codec preference 4 g711alaw
 codec preference 5 g726r16
 codec preference 6 g726r24
 codec preference 7 g726r32
 codec preference 8 g723ar53
 codec preference 9 g723ar63
!
!
voice register pool  1
 id ip 192.0.2.168 mask 0.0.0.0
 dtmf-relay rtp-nte
!
!
interface FastEthernet0/0
 ip address 192.0.2.1 255.255.255.0
 no ip proxy-arp
no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 192.0.2.200
ip route 0.0.0.0 0.0.0.0 192.0.2.1
ip route 0.0.0.0 0.0.0.0 192.0.2.225
!
ip http server
!
control-plane
!
voice-port 2/0
!
voice-port 2/1
!
voice-port 2/2
.
.
.
voice-port 2/22
!
voice-port 2/23
!
!
dial-peer voice 1 pots
 destination-pattern 8888
 port 2/1
!
dial-peer voice 9999 voip
 destination-pattern 9999
 session protocol sipv2
 session target ipv4:192.0.2.228
 dtmf-relay sip-kpml
 codec g711ulaw
!
dial-peer voice 5555555 voip
 destination-pattern 5555555
```

```
 session protocol sipv2
 session target ipv4:192.0.2.230
 codec g711ulaw
!
dial-peer voice 36 voip
 destination-pattern 36601
session protocol sipv2
 session target ipv4:192.0.2.235
 codec g711ulaw
!
dial-peer voice 444 voip
 destination-pattern 444
 session protocol sipv2
 session target ipv4:192.0.2.140
 codec g711ulaw
!
dial-peer voice 333 voip
 destination-pattern 333
 session protocol sipv2
 session target ipv4:192.0.2.200
!
!
sip-ua
 retry invite 3
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
end
```

Configuring SIP KPML-Based Out-of-Band DTMF Relay

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay sip-kpml**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode or any other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the designated dial peer.
Step 4	**dtmf-relay sip-kpml** **Example:** `Router(config-dial-peer)# dtmf-relay sip-kpml`	Forwards DTMF tones using SIP KPML messages.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Verifying SIP DTMF Support

To verify SIP DTMF support, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show running-config**
2. **show sip-ua retry**
3. **show sip-ua statistics**
4. **show sip-ua status**
5. **show sip-ua timers**
6. **show voip rtp connections**
7. **show sip-ua calls**

DETAILED STEPS

Step 1 **show running-config**

Use this command to show dial-peer configurations.

The following sample output shows that the **dtmf-relay sip-notify** command is configured in dial peer 123:

Example:

```
Router# show running-config
.
.
.
dial-peer voice 123 voip
```

```
 destination-pattern [12]...
 monitor probe icmp-ping
 session protocol sipv2
 session target ipv4:10.8.17.42
 dtmf-relay sip-notify
```

The following sample output shows that DTMF relay and NTE are configured on the dial peer.

Example:

```
Router# show running-config
!
dial-peer voice 1000 pots
 destination-pattern 4961234
 port 1/0/0
!
dial-peer voice 2000 voip
 application session
 destination-pattern 4965678
 session protocol sipv2
 session target ipv4:192.0.2.34
 dtmf-relay rtp-nte
! RTP payload type value = 101 (default)
!
dial-peer voice 3000 voip
 application session
 destination-pattern 2021010101
 session protocol sipv2
 session target ipv4:192.0.2.34
 dtmf-relay rtp-nte
 rtp payload-type nte 110
! RTP payload type value = 110 (user assigned)
!
```

Step 2 **show sip-ua retry**

Use this command to display SIP retry statistics.

Example:

```
Router# show sip-ua retry
SIP UA Retry Values
invite retry count = 6 response retry count = 1
bye retry count = 1 cancel retry count = 1
prack retry count = 10 comet retry count = 10
reliable 1xx count = 6 notify retry count = 10
```

Step 3 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics.

Tip To reset counters for the **show sip-ua statistics**display, use the **clear sip-ua statistics** command.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
Informational:
Trying 4/2, Ringing 2/1,
Forwarded 0/0, Queued 0/0,
SessionProgress 0/0
Success:
OkInvite 1/2, OkBye 0/1,
```

```
OkCancel 1/0, OkOptions 0/0,
OkPrack 2/0, OkPreconditionMet 0/0,
OkNotify 1/0, 202Accepted 0/1
Redirection (Inbound only):
MultipleChoice 0, MovedPermanently 0,
MovedTemporarily 0, SeeOther 0,
UseProxy 0, AlternateService 0
Client Error:
BadRequest 0/0, 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
RequestCancel 1/0, NotAcceptableMedia 0/0
Server Error:
InternalError 0/1, NotImplemented 0/0,
BadGateway 0/0, ServiceUnavail 0/0,
GatewayTimeout 0/0, BadSipVer 0/0,
PreCondFailure 0/0
Global Failure:
BusyEverywhere 0/0, Decline 0/0,
NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound) /* Traffic Statistics
Invite 3/2, Ack 3/2, Bye 1/0,
Cancel 0/1, Options 0/0,
Prack 0/2, Comet 0/0,
Notify 0/1, Refer 1/0
Retry Statistics        /* Retry Statistics
Invite 0, Bye 0, Cancel 0, Response 0,
Prack 0, Comet 0, Reliable1xx 0, Notify 0
```

Following is sample output verifying configuration of the SIP INFO Method for DTMF Tone Generation feature

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
Informational:
Trying 1/1, Ringing 0/0,
Forwarded 0/0, Queued 0/0,
SessionProgress 0/1
Success:
OkInvite 0/1, OkBye 1/0,
OkCancel 0/0, OkOptions 0/0,
OkPrack 0/0, OkPreconditionMet 0/0
OkSubscibe 0/0, OkNotify 0/0,
OkInfo 0/0, 202Accepted 0/0
Redirection (Inbound only):
MultipleChoice 0, MovedPermanently 0,
MovedTemporarily 0, SeeOther 0,
UseProxy 0, AlternateService 0
Client Error:
BadRequest 0/0, 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,
BadEvent 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 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0, Info 0/0
Retry Statistics
Invite 0, Bye 0, Cancel 0, Response 0, Notify 0
```

Step 4 **show sip-ua status**

Use this command to display status for the SIP user agent.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Session name line (s=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
```

The following sample output shows that the time interval between consecutive NOTIFY messages for a telephone event is the default of 2000 ms:

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
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 : 6
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: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
 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
```

The following sample output shows configuration of the SIP INFO Method for DTMF Tone Generation feature.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
 Version line (v=) required
 Owner line (o=) required
 Session name line (s=) required
 Timespec line (t=) required
 Media supported: audio image
 Network types supported: IN
 Address types supported: IP4
 Transport types supported: RTP/AVP udptl
```

Step 5 **show sip-ua timers**

Use this command to display the current settings for SIP user-agent timers.

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 300000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
```

Step 6 **show voip rtp connections**

Use this command to show local and remote Calling ID and IP address and port information.

Step 7 **show sip-ua calls**

Use this command to ensure the DTMF method is SIP-KPML.

The following sample output shows that the DTMF method isSIP-KPML.

Example:

```
router# show sip-ua calls
SIP UAC CALL INFO
Call 1
SIP Call ID                : 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
   State of the call       : STATE_ACTIVE (7)
   Substate of the call    : SUBSTATE_NONE (0)
```

```
   Calling Number          :
   Called Number           : 8888
   Bit Flags               : 0xD44018 0x100 0x0
   CC Call ID              : 6
   Source IP Address (Sig ): 192.0.2.1
   Destn SIP Req Addr:Port : 192.0.2.2:5060
   Destn SIP Resp Addr:Port: 192.0.2.3:5060
   Destination Name        : 192.0.2.4.250
   Number of Media Streams : 1
   Number of Active Streams: 1
   RTP Fork Object         : 0x0
   Media Mode              : flow-through
   Media Stream 1
     State of the stream      : STREAM_ACTIVE
     Stream Call ID           : 6
     Stream Type              : voice-only (0)
     Negotiated Codec         : g711ulaw (160 bytes)
 Codec Payload Type       : 0
     Negotiated Dtmf-relay    : sip-kpml
     Dtmf-relay Payload Type  : 0
     Media Source IP Addr:Port: 192.0.2.5:17576
     Media Dest IP Addr:Port  : 192.0.2.6:17468
     Orig Media Dest IP Addr:Port : 0.0.0.0:0
   Number of SIP User Agent Client(UAC) calls: 1
SIP UAS CALL INFO
   Number of SIP User Agent Server(UAS) calls: 0
```

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "Basic Troubleshooting Procedures" section.

- To enable debugging for RTP named event packets, use the **debug voip rtp** command.
- To enable KPML debugs, use the **debug kpml** command.
- To enable SIP debugs, use the **debug ccsip** command.
- Collect debugs while the call is being established and during digit presses.
- If an established call is not sending digits though KPML, use the **show sip-ua calls** command to ensure SIP-KPML is included in the negotiation process.

Additional References

The following sections provide references related to the SIP DTMF features.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
SIP commands	*Cisco IOS Voice Command Reference*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

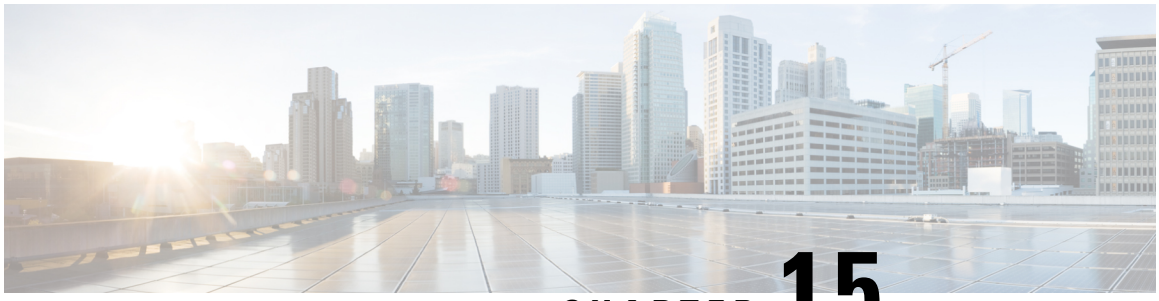
MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2833	*RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 15

Configuring SIP MWI Features

This module describes message-waiting indication (MWI) in a SIP-enabled network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP MWI

SIP MWI NOTIFY - QSIG MWI Translation Feature

- Ensure that you have a working SIP network with the following:
  - A voice-messaging system that provides a SIP MWI Notify message to the phone--including Cisco Unified Communications Manager (formerly known as Cisco CallManager), Release 5.0 or later or Cisco Unified Communications Manager Express (Cisco Unified CME, formerly known as Cisco CallManager Express) Release 4.0 or later.

- Voice messaging on Cisco Unity 4.0.1 or later releases (colocated or integrated with the Cisco Unified Communications Manager) or an ISDN Q-signaling (QSIG) PBX.

- Connect gateway and Cisco routers directly to a PBX.
- Ensure that phones connected to PBXs support MWI notification.

SIP Audible Message-Waiting Indicator for FXS Phones Feature

- The MWI tone is generated by the voice-mail server. Be sure that you understand how to configure MWI service on a voice-mail server (such as Cisco Unity).

Restrictions for SIP MWI

SIP MWI NOTIFY - QSIG MWI Translation Feature

- Visual MWI for phones is a functionality of the phone itself and is not addressed in this document.
- The feature supports only SIP unsolicited notify and does not support SIP subscribe notify.
- This feature is not supported in trunk groups in ISDN circuits. In this scenario, trunk groups disable the SIP MWI feature.

SIP Audible Message-Waiting Indicator for FXS Phones Feature

- The SIP Audible Message-Waiting Indicator for FXS Phones feature does not provide the following functionality:
  - Security or authentication services
  - Call redirection to the voice-mail server when the line is busy or there is no answer
  - Instructions on accessing the voice-mail server or retrieving voice messages

Information About SIP MWI

The SIP Audible Message-Waiting Indicator for FXS Phones feature enables an FXS port on a voice gateway to receive audible MWI in a SIP-enabled network. The FXS port on a voice gateway is an RJ-11 connector that allows connections to basic telephone service equipment.

This feature provides the following benefits:

- Message waiting is now indicated to FXS phone users through an audible tone, replicating the functionality users have with traditional telephone systems.
- By means of the Cisco IOS command-line interface, you can enable or disable MWI under the voice port and configure one voice-mail server per user agent (UA) or voice gateway.

To configure SIP MWI support, you should understand the following concepts:

SUBSCRIBE NOTIFY MWI

MWI is a common feature of telephone networks and uses an audible indication (such as a special dial tone) that a message is waiting. The IETF draft A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP) draft-ietf-sipping-mwi-03.txt defines MWI as "a SIP event package carrying message waiting status and message summaries from a messaging system to an interested user agent."

In Cisco SIP networks, the event notification mechanisms used to carry message waiting status are the SUBSCRIBE and NOTIFY methods. The SUBSCRIBE method requests notification of an event. The NOTIFY method provides notification that an event requested by an earlier SUBSCRIBE method has occurred.

Note For information on the SUBSCRIBE and NOTIFY methods, see the "Configuring Additional SIP Application Support" chapter of the *Cisco IOS SIP Configuration Guide* .

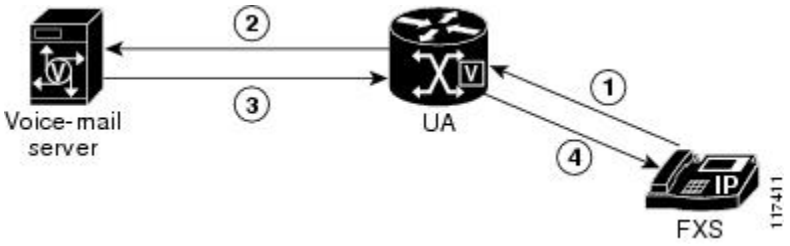
In this feature, a UA (on behalf of the analog FXS phone) subscribes to a voice-mail server to request notification of mailbox status. When the mailbox status changes, the voice-mail server notifies the UA. The UA then indicates that there is a change in mailbox status by providing an MWI tone when the user takes the phone off-hook.

The frequency and cadence of the MWI tone may vary from country to country. For North America, it is defined in GR-506. After you configure the **cp tone** command under your voice port, Cisco IOS software chooses the correct MWI tone accordingly.

Each voice port has its own subscription and notification process. If there are multiple dial peers associated with an FXS voice port, multiple subscriptions are sent to the voice-mail server. If the voice port does not have MWI enabled, the voice gateway returns a 481 Call Leg/Transaction Does Not Exist message to the voice-mail server.

The figure below shows the basic MWI subscription and notification flow.

Figure 86: MWI Notification Flow



1. The user enables the MWI service for the FXS phone by configuring the voice gateway.
2. The UA sends a subscription request to the server on the user's behalf.
3. The voice-mail server notifies the UA when there is a change in voice-mail status.
4. The UA notifies the phone user with an audible tone.

Unsolicited MWI

In addition to the MWI status forwarded by using the SUBSCRIBE and NOTIFY methods, unsolicited MWI notify is also supported. With unsolicited MWI, MWI service is initially configured on the voice-mail server.

The UA does not need to subscribe to the voice-mail server to receive MWI service. If configured for unsolicited MWI, the voice-mail server automatically sends a SIP notification message to the UA if the mailbox status changes.

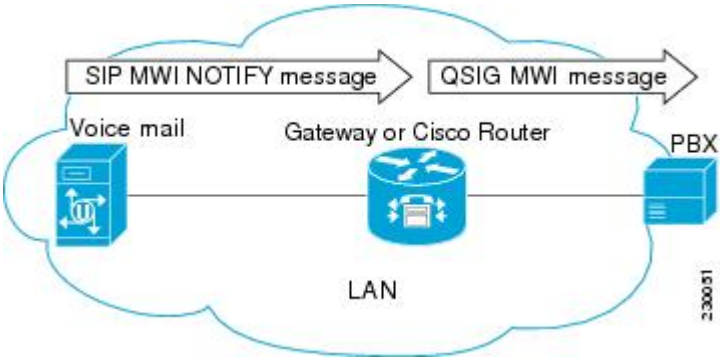
SIP MWI NOTIFY - QSIG MWI Translation

In Cisco IOS Release 12.4(11)T, the SIP MWI NOTIFY - QSIG Translation feature enhances MWI functionality to include SIP-MWI-NOTIFY-to-QSIG-MWI translation between Cisco gateways or routers over a LAN or WAN and extends message waiting indicator (MWI) functionality for SIP MWI and QSIG MWI interoperation to enable sending MWI over QSIG from a Cisco IOS SIP gateway to a PBX.

When the SIP Unsolicited NOTIFY is received from voice mail, the Cisco router translates this event to activate QSIG MWI to the PBX via PSTN. The PBX will switch the MWI lamp either on or off on the corresponding IP phone as appropriate.

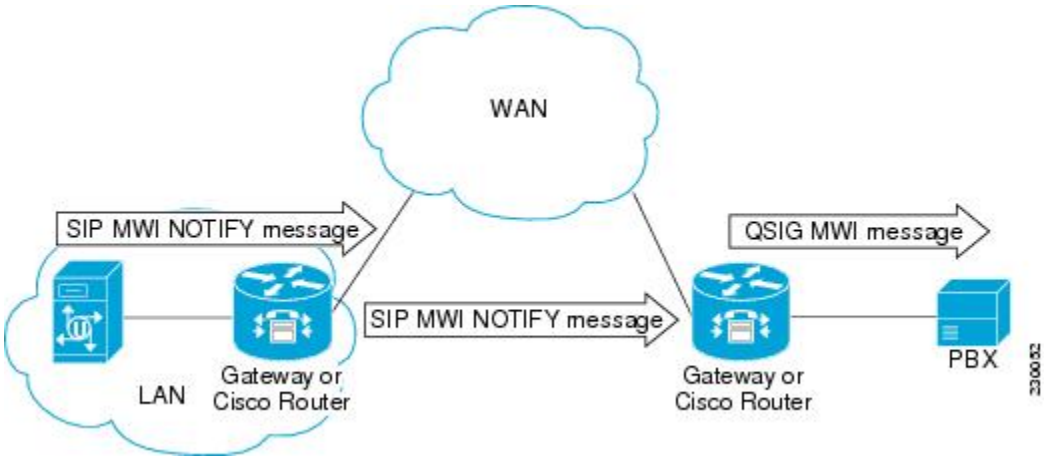
This feature supports only Unsolicited NOTIFY. Subscribe NOTIFY is not supported by this feature.

In the figure below, the Cisco router receives the SIP Unsolicited NOTIFY, performs the protocol translation, and initiates the QSIG MWI call to the PBX, where it is routed to the appropriate phone.



Whether the SIP Unsolicited NOTIFY is received via LAN or WAN does not matter as long as the PBX is connected to the gateway or Cisco router, and not to the remote voice mail server.

In the figure below, a voice mail system, such as Cisco Unity, and Unified CME are connected to the same LAN and a remote Unified CME is connected across the WAN. In this scenario, the protocol translation is performed at the remote Unified CME router and the QSIG MWI message is sent to the PBX.



How to Configure SIP MWI

This section contains the following procedures for configuring the SIP Audible MWI for FXS Phones feature:

Note For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring SIP MWI NOTIFY - QSIG MWI Translation

This section contains information for configuring SIP MWI NOTIFY - QSIG MWI Translation on a gateway.

Note All configuration for this feature is done on the gateway or Cisco router.

Configuring the Gateway

To configure SIP MWI NOTIFY - QSIG MWI Translation on a gateway, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice-port** *slot* / *port*
4. **mwi**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice-port** *slot* / *port* **Example:** `Router(config)# voice-port 2/2`	Enters voice-port configuration mode for the specified PRI or BRI voice port.
Step 4	**mwi** **Example:** `Router (config-voiceport)# mwi`	Enables MWI on this voice port. **Note** If the voice port is not configured for MWI, the gateway returns a 481 Call Leg/Transaction Does Not Exist message to the voice-mail server. If multiple dial peers are associated with the same voice port, multiple subscriptions are sent to the voice-mail server.
Step 5	**exit** **Example:** `Router(config-dial-peer-voice)# exit`	Exits the current configuration mode.

Configuring Voice-Mail Server Settings on the UA

Note This configuration initiates the capability of a UA or voice gateway to indicate voice-mail status changes. One voice-mail server is configured per voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **mwi-server** {**ipv4:** *destination-address* | **dns :** *host-name*} [**expires***seconds*] [**port***port*] [**transport** {**tcp** | **udp**}] [**unsolicited**]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**mwi-server** {**ipv4:** *destination-address* | **dns :** *host-name*} [**expires***seconds*] [**port***port*] [**transport** {**tcp** | **udp**}] [**unsolicited**] **Example:** `Router(config-sip-ua)# mwi-server dns:test.example.com expires 86000 port 5060 transport udp unsolicited`	Configures voice-mail server settings on a voice gateway or UA. Keywords and arguments are as follows: • **ipv4:** *destination-address* --IP address of the voice-mail server. • **dns:** *host-name* --Host device housing the domain name server that resolves the name of the voice-mail server. The argument should contain the complete hostname to be associated with the target address; for example, **dns:test.example.com**. • **expires** *seconds* --Subscription expiration time, in seconds. Range is from 1 to 999999. Default is 3600. • **port** *port* --Port number on the voice-mail server. Default is 5060. • **transport** --Transport protocol to the voice-mail server. Valid values are tcp and udp. Default is UDP. • **unsolicited** --Requires the voice-mail server to send a SIP notification message to the voice gateway or UA if the mailbox status changes. Removes the requirement that the voice gateway subscribe for MWI service.
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring the Voice-Mail Server for Unsolicited

To configure the Cisco Unity voice-mail server to be unsolicited, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **mwi-server ipv4:** x.x.x.x **unsolicited**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP-user-agent configuration mode.
Step 4	**mwi-server ipv4:** x.x.x.x **unsolicited** **Example:** `Router (config-sip-ua)# mwi-server ipv4:192.0.10.150 unsolicited`	Configures the specified voice-mail (MWI) server to be unsolicited. (That is, requires the server to send a SIP notification message to the voice gateway or user agent if the mailbox status changes. Removes the requirement that the voice gateway subscribe for MWI service.)
Step 5	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current configuration mode.

Enabling MWI Under an FXS Voice Port

To enable MWI under the specified FXS voice port, perform the following steps.

Note If the voice port does not have MWI enabled, the voice gateway returns a 481 Call Leg/Transaction Does Not Exist message to the voice-mail server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice-port** *port*
4. **cptone** *locale*
5. **mwi**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice-port** *port* **Example:** `Router(config)# voice-port 2/2`	Enters voice-port configuration mode. To find the *port* argument for your router, see the Cisco IOS Voice Command Reference, Release 12.3T.
Step 4	**cptone** *locale* **Example:** `Router(config-voiceport)# cptone us`	Specifies a regional analog voice-interface-related tone, ring, and cadence setting for a specified FXS voice port.
Step 5	**mwi** **Example:** `Router(config-voiceport)# mwi`	Enables MWI for a specified FXS voice port.
Step 6	**exit** **Example:** `Router(config-voiceport)# exit`	Exits the current mode.

Verifying MWI Settings

SUMMARY STEPS

1. **show sip-ua mwi**

DETAILED STEPS

show sip-ua mwi

Use this command to display SIP MWI settings from the voice-mail server. The command displays endpoint status as OFF if a message is deleted or if no message is waiting. The endpoint status changes to ON when a message is waiting.

The following sample output shows endpoint status as OFF if a message is deleted or if no message is waiting. The endpoint status changes to ON when a message is waiting.

Example:

```
Router#
show sip-ua mwi
MWI type: 2
MWI server: dns:unity-vm.example1.com
MWI expires: 60
MWI port: 5060
MWI transport type: UDP
MWI unsolicited
MWI server IP address:
C801011E
0
0
0
0
0
0
0
MWI ipaddr cnt 1:
MWI ipaddr idx 0:
MWI server: 192.168.1.30, port 5060, transport 1
MWI server dns lookup retry cnt: 0
endpoint 8000  mwi status ON
endpoint 8000  mwi status ON
endpoint 8001  mwi status OFF
```

Configuring VMWI on analog phones connected to FXS

There are two types of visual message waiting indicator (VMWI) features: Frequency-shift Keying (FSK) and DC voltage. The message-waiting lamp can be enabled to flash on an analog phone that requires an FSK message to activate a visual indicator. The DC Voltage VMWI feature is used to flash the message-waiting lamp on an analog phone which requires DC voltage instead of an FSK message. For all other applications, such as MGCP, FSK VMWI is used even if the voice gateway is configured for DC voltage VMWI. The configuration for DC voltage VMWI is supported only for Foreign Exchange Station (FXS) ports on the Cisco VG224 analog voice gateway with analog device version V1.3 and V2.1.

The Cisco VG224 can only support 12 Ringer Equivalency Number (REN) for ringing 24 onboard analog FXS voice ports. To support ringing and DC Voltage VMWI for 24 analog voice ports, stagger-ringing logic is used to maximize the limited REN resource. When a system runs out of REN because too many voice ports are being rung, the MWI lamp temporarily turns off to free up REN to ring the voice ports.

To enable MWI under the specified FXS voice port, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice-port** *port*
4. **mwi**
5. Do one of the following:
   - **vmwi dc-voltage**
   -
   -
   - **vmwi fsk**
6. **exit**
7. **sip-ua**
8. **mwi-server** {**ipv4**:*destination-address* | **dns**:*host-name*} [**unsolicited**]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice-port** *port* **Example:** `Router(config)# voice-port 2/0`	Enters voice-port configuration mode. • *port* --Syntax is platform-dependent. Type **?** to determine.
Step 4	**mwi** **Example:** `Router(config-voiceport)# mwi`	Enables MWI for a specified voice port.

	Command or Action	Purpose
Step 5	Do one of the following: • **vmwi dc-voltage** • • • **vmwi fsk** **Example:** `Router(config-voiceport)# vmwi dc-voltage`	(Optional) Enables DC voltage or FSK VMWI on a Cisco VG224 onboard analog FXS voice port. You do not need to perform this step for the Cisco VG202 and Cisco VG204. They support FSK only. VMWI is configured automatically when MWI is configured on the voice port. This step is required for the VG224. If an FSK phone is connected to the voice port, use the **fsk** keyword. If a DC voltage phone is connected to the voice port, use the **dc-voltage** keyword.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits to the next highest mode in the configuration mode hierarchy.
Step 7	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters Session Initiation Protocol user agent configuration mode for configuring the user agent.
Step 8	**mwi-server** {**ipv4**:*destination-address* | **dns**:*host-name*} [**unsolicited**] **Example:** `Router(config-sip-ua)# mwi-server ipv4:1.5.49.200` **Example:** `or` **Example:** `Router(config-sip-ua)# mwi-server dns:server.yourcompany.com unsolicited`	Specifies voice-mail server settings on a voice gateway or user agent (ua). **Note** The **sip-server** and **mwi expires commands**under the telephony-service configuration mode have been migrated to **mwi-server** to support DNS format of the Session Initiation Protocol (SIP) server.
Step 9	**end** **Example:** `Router(config-voiceport)# end`	Exits voice-port configuration mode and returns to privileged EXEC mode.

What to do next

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "Verifying and Troubleshooting SIP Features" chapter in the *Cisco IOS SIP Configuration Guide*.

- Use the **debug ccsip messages** command for debugging purposes.
- Use the debug vpm all command for showing the VMWI state of a voice-port

Following is sample output for this command:

Sample Output for the debug ccsip messages Command

The following sample output is from the perspective of a SIP UA acting on the behalf of an analog FXS phone. The output shows that when the phone connected to the UA is called and the line is busy, the caller leaves a message. The UA, connected to the voice-mail server, receives notification and provides a tone to the user. The user listens to the message and deletes it.

```
Router# debug ccsip messages
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
INVITE sip:78002@csps-release.example1.com:5060 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK24E9
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Supported: 100rel,timer
Min-SE:  1800
Cisco-Guid: 3659524871-1844515286-2148452871-566800187
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Remote-Party-ID: "SIPMWI-1" <sip:78001@192.168.1.174>;party=calling;screen=no;privacy=off
Timestamp: 1022206059
Contact: <sip:78001@192.168.1.174:5060>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 234
v=0
o=CiscoSystemsSIP-GW-UserAgent 5421 615 IN IP4 192.168.1.174
s=SIP Call
c=IN IP4 192.168.1.174
t=0 0
m=audio 16818 RTP/AVP 18 19
c=IN IP4 192.168.1.174
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
a=ptime:20
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
```

```
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK24E9
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>
CSeq: 101 INVITE
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK24E9
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=5ea400de-695763f1
CSeq: 101 INVITE
Proxy-Authenticate: DIGEST realm="example.com", nonce="40871b34", qop="auth", algorithm=MD5
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:78002@csps-release.example1.com:5060 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK24E9
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=5ea400de-695763f1
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Max-Forwards: 70
CSeq: 101 ACK
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
INVITE sip:78002@csps-release.example1.com:5060 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK612
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Supported: 100rel,timer
Min-SE:  1800
Cisco-Guid: 3659524871-1844515286-2148452871-566800187
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq: 102 INVITE
Max-Forwards: 70
Remote-Party-ID: "SIPMWI-1" <sip:78001@192.168.1.174>;party=calling;screen=no;privacy=off
Timestamp: 1022206059
Contact: <sip:78001@192.168.1.174:5060>
Expires: 180
Allow-Events: telephone-event
Proxy-Authorization: Digest
username="user1",realm="example.com",uri="sip:192.168.1.37",response="df92654ce55d734
6398013442919e7fc",nonce="40871b34",cnonce="2AEBD5CD",qop=auth,algorithm=MD5,nc=00000001
Content-Type: application/sdp
Content-Length: 234
v=0
o=CiscoSystemsSIP-GW-UserAgent 5421 615 IN IP4 192.168.1.174
s=SIP Call
c=IN IP4 192.168.1.174
t=0 0
m=audio 16818 RTP/AVP 18 19
c=IN IP4 192.168.1.174
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
```

```
a=rtpmap:19 CN/8000
a=ptime:20
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK612
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>
CSeq: 102 INVITE
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
INVITE sip:78002@192.168.1.174:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.37:5060;branch=474b6083-19c218c7-16e9de49-93b83d71-1
Record-Route:
<sip:78001.474b6083-19c218c7-16e9de49-93b83d71@192.168.1.174:5060;maddr=192.168.1.37>
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK612
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Supported: 100rel,timer
Min-SE: 1800
Cisco-Guid: 3659524871-1844515286-2148452871-566800187
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO,
UPDATE, REGISTER
CSeq: 102 INVITE
Max-Forwards: 69
Remote-Party-ID: "SIPMWI-1" <sip:78001@192.168.1.174>;party=calling;screen=no;privacy=off
Timestamp: 1022206059
Contact: <sip:78001@192.168.1.174:5060>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Length: 234
v=0
o=CiscoSystemsSIP-GW-UserAgent 5421 615 IN IP4 192.168.1.174
s=SIP Call
c=IN IP4 192.168.1.174
t=0 0
m=audio 16818 RTP/AVP 18 19
c=IN IP4 192.168.1.174
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:19 CN/8000
a=ptime:20
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.37:5060;branch=474b6083-19c218c7-16e9de49-93b83d71-1,SIP/2.0/UDP
 192.168.1.174:5060;re
ceived=192.168.1.174;branch=z9hG4bK612
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A843C-187B
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Timestamp: 1022206059
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 102 INVITE
Allow-Events: telephone-event
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
```

```
Sent:
SIP/2.0 486 Busy here
Via: SIP/2.0/UDP 192.168.1.37:5060;branch=474b6083-19c218c7-16e9de49-93b83d71-1,SIP/2.0/UDP
 192.168.1.174:5060;re
ceived=192.168.1.174;branch=z9hG4bK612
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A843C-187B
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Timestamp: 1022206059
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 102 INVITE
Allow-Events: telephone-event
Reason: Q.850;cause=17
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
ACK sip:78002@192.168.1.174:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.37:5060;branch=474b6083-19c218c7-16e9de49-93b83d71-1
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A843C-187B
CSeq: 102 ACK
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 180 Ringing
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK612
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
CSeq: 102 INVITE
Content-Length: 0
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK612
Record-Route:
<sip:7200@example1.com:5060;maddr=192.168.1.37>,<sip:78002@csps-release.example1.com:5060;maddr=192.168.1.37>
Contact: sip:7200@192.168.1.30:5060
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
CSeq: 102 INVITE
Content-Length: 166
Content-Type: application/sdp
v=0
o=192.168.1.30 7542610 7542610 IN IP4 192.168.1.30
s=No Subject
c=IN IP4 192.168.1.30
t=0 0
m=audio 22840 RTP/AVP 18
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
00:11:29: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
ACK sip:78002@csps-release.example1.com:5060;maddr=192.168.1.37 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK10EF
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
Route: <sip:7200@example1.com:5060;maddr=192.168.1.37>,<sip:7200@192.168.1.30:5060>
Max-Forwards: 70
```

```
CSeq: 102 ACK
Proxy-Authorization: Digest
username="user1",realm="example.com",uri="sip:192.168.1.37",response="631ff1eec9e21b0
2fcbdbe932c9f7b5b",nonce="40871b34",cnonce="81C16CF6",qop=auth,algorithm=MD5,nc=00000002
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
REGISTER sip:csps-release.example1.com:5060 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK171F
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
Date: Fri, 24 May 2002 02:07:48 GMT
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 70
Timestamp: 1022206068
CSeq: 14 REGISTER
Contact: <sip:78002@192.168.1.174:5060>
Expires: 60
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK171F
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
CSeq: 14 REGISTER
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK171F
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
CSeq: 14 REGISTER
WWW-Authenticate: DIGEST realm="example.com", nonce="40871b3d", qop="auth", algorithm=MD5
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
REGISTER sip:csps-release.example1.com:5060 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK21B5
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
Date: Fri, 24 May 2002 02:07:48 GMT
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 70
Timestamp: 1022206068
CSeq: 15 REGISTER
Contact: <sip:78002@192.168.1.174:5060>
Expires: 60
Authorization: Digest
username="user2",realm="example.com",uri="sip:192.168.1.37",response="134885a71dd9690370196
089e445e955",nonce="40871b3d",cnonce="7446932B",qop=auth,algorithm=MD5,nc=00000001
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK21B5
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
```

```
CSeq: 15 REGISTER
Content-Length: 0
00:11:38: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK21B5
Call-ID: 6CD62112-6DF011D6-8006CA07-21C8AF3B
From: "user2" <sip:78002@192.168.1.174>;tag=AA7F4-1F83
To: <sip:78002@csps-release.example1.com>
CSeq: 15 REGISTER
Contact: <sip:78002@192.168.1.174:5060>;expires=60
Content-Length: 0
00:11:44: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Sent:
BYE sip:78002@csps-release.example1.com:5060;maddr=192.168.1.37 SIP/2.0
Via: SIP/2.0/UDP  192.168.1.174:5060;branch=z9hG4bK79A
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
Date: Fri, 24 May 2002 02:07:39 GMT
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
User-Agent: Cisco-SIPGateway/IOS-12.x
Max-Forwards: 70
Route: <sip:7200@example1.com:5060;maddr=192.168.1.37>,<sip:7200@192.168.1.30:5060>
Timestamp: 1022206074
CSeq: 103 BYE
Reason: Q.850;cause=16
Proxy-Authorization: Digest
username="user1",realm="example.com",uri="sip:192.168.1.37",response="dffc15fe72d26b9
3d78162852ae1a341",nonce="40871b34",cnonce="AF9FD85E",qop=auth,algorithm=MD5,nc=00000003
Content-Length: 0
00:11:44: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK79A
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
CSeq: 103 BYE
Content-Length: 0
00:11:44: //-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
From: "SIPMWI-1" <sip:78001@192.168.1.174>;tag=A842C-2612
To: <sip:78002@csps-release.example1.com>;tag=A59035E8274E4600A8F3D15C3DAB9631
Via: SIP/2.0/UDP 192.168.1.174:5060;received=192.168.1.174;branch=z9hG4bK79A
Call-ID: DBAC09D2-6DF111D6-8011CA07-21C8AF3B@192.168.1.174
CSeq: 103 BYE
Content-Length: 0
```

Sample relevant output for the debug vpm all command

```
Process vmwi. vmwi state: OFF
The phone is not onhook (1). Delay the vmwi processing.
Process dc-voltage vmwi. State: OFF
*Mar  2 02:33:34.841: [2/0] c2400_dc_volt_mwi: on=0
The phone is not onhook (1). Delay the vmwi processing.
Process vmwi. vmwi state: ON
```

Configuration Examples for SIP MWI

The following example shows that SIP MWI is configured on the gateway.

```
Router# show running-config
Building configuration...
Current configuration : 14146 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service dhcp
!
boot-start-marker
boot system flash:c2430-is-mz.mwi_dns
boot-end-marker
!
card type e1 1
logging buffered 9000000 debugging
!
username all
network-clock-participate E1 1/0
network-clock-participate E1 1/1
no aaa new-model
no ip subnet-zero
!
ip domain name example1.com
ip name-server 192.168.1.1
ip dhcp excluded-address 172.16.224.97
!
isdn switch-type primary-qsig
!
trunk group  Incoming
!
voice-card 0
!
voice service voip
 fax protocol t38 ls-redundancy 0 hs-redundancy 0 fallback none
 h323
 sip
!
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 2 g729r8
 codec preference 3 g726r32
!
voice hpi capture buffer 100000
voice hpi capture destination flash:t1.dat
!
voice translation-rule 1
 rule 1 /.*/ /8005550100/
!
voice translation-profile Out
 translate calling 1
!
controller E1 1/0
 linecode ami
 pri-group timeslots 1-31
!
controller E1 1/1
```

```
 linecode ami
 pri-group timeslots 1-10,16
!
interface FastEthernet0/0
 ip address 192.168.1.172 255.255.255.0
 no ip mroute-cache
 duplex half
 speed auto
!
interface FastEthernet0/1
 ip address 10.2.141.19 255.255.0.0
 no ip mroute-cache
 duplex auto
 speed auto
!
ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.2
!
ip rtcp report interval 30000
!
control-plane
!
! Enable MWI on voice ports 2/0 and 2/1.
!
voice-port 2/0
 mwi
 timeouts ringing 30
 station-id name SIPUser1
 station-id number 8000
 caller-id enable
!
voice-port 2/1
 mwi
 timeouts ringing 30
 station-id name SIPUser2
 station-id number 8001
 caller-id enable
!
dial-peer cor custom
!
! Configure dial peers.
!
dial-peer voice 1 pots
 preference 5
 destination-pattern 8000
 port 2/0
!
dial-peer voice 2 pots
 preference 5
 destination-pattern 8001
 port 2/1
!
dial-peer voice 3 voip
 destination-pattern .T
 voice-class codec 1
 session protocol sipv2
 session target sip-server
 dtmf-relay rtp-nte
!
dial-peer voice 7 pots
 trunkgroup Incoming
 destination-pattern 789...
!
```

```
dial-peer voice 8 pots
 trunkgroup Incoming
 destination-pattern 789...
!
dial-peer voice 22 voip
 destination-pattern 7232
 session protocol sipv2
 session target sip-server
 dtmf-relay rtp-nte
 codec g711ulaw
!
gateway
 timer receive-rtcp 5
 timer receive-rtp 1200
!
! Configure the voice-mail server settings on the gateway with the mwi-server command.
!
sip-ua
 authentication username user1 password password1 realm example.com
 mwi-server dns:test.example.com expires 60 port 5060 transport udp unsolicited
 registrar dns:csps-release.test.example.com expires 3600
 sip-server dns:csps-release.test.example.com
!
telephony-service
 max-dn 100
 max-conferences 4
!
ephone-dn  1
!
line con 0
 exec-timeout 0 0
 password 7 password2
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 password 7 password3
 login
 transport preferred all
 transport input all
 transport output all
!
end
```

Configuration Example for SIP MWI NOTIFY - QSIG MWI Translation

The following example shows a sample configuration of the SIP MWI NOTIFY - QSIG MWI Translation feature on a SIP gateway.

```
dial-peer voice 1000 voip
 destination-pattern .T
 session protocol sipv2
 session target ipv4:10.120.70.10
 incoming called-number .T
 dtmf-relay rtp-nte
!
```

```
sip-ua
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
!
end
```

Configuration Example for SIP VMWI

```
Router# show running-config
Building configuration...
!
sip-ua
 mwi-server ipv4:9.13.40.83 expires 3600 port 7012 transport udp unsolicited
!
voice-port 2/0
 vmwi dc-voltage
 mwi
!
```

Additional References

General SIP References

- "Basic SIP Configuration" chapter--Describes underlying SIP technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance.

References Mentioned in This Chapter (listed alphabetically)

- RFC 3842 , "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)" at http://www.ietf.org/rfc/rfc3842.txt
- *Cisco IOS Voice Command Reference*
- *Cisco IOS Voice Configuration Library*

Feature Information for SIP MWI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 52: Feature Information for Configuring SIP MWI Features

Feature Name	Releases	Feature Information
SIP Audible Message-Waiting Indicator for FXS Phones	12.3(8)T	This feature enables an FXS port on a voice gateway to receive audible MWI in a SIP-enabled network.
SIP MWI NOTIFY - QSIG MWI Translation	12.4(11)T	This feature was introduced. This feature is used to configure SIP MWI NOTIFY - QSIG MWI Translation on a gateway.
VMWI on analog phones connected to FXS	15.1(2)T	This feature introduces support fpor VMWI on analog phones connected to FXS.

CHAPTER 16

Configuring SIP QoS Features

This chapter discusses the following features that affect quality of service (QoS) in SIP networks:

- Enhanced Codec Support for SIP Using Dynamic Payloads
- Measurement-Based Call Admission Control for SIP
- SIP Gateway Support of RSVP
- SIP Gateway Support of 'tel' URL
- SIP: Hold Timer Support
- SIP Media Inactivity Timer
- SIP Stack Portability

Note This feature is described in "Configuring SIP Message, Timer, and Response Features".

Feature History for Enhanced Codec Support for SIP Using Dynamic Payloads

Release	Modification
12.2(11)T	This feature was introduced.

Feature History for Measurement-Based Call Admission Control for SIP

Release	Modification
12.2(15)T	This feature was introduced.

Feature History for SIP Gateway Support of RSVP

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB1	This feature was implemented on an additional platform.

Release	Modification
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Feature History for SIP Gateway Support of 'tel' URL

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(2)XB1	This feature was implemented on an additional platform.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Feature History for SIP: Hold Timer Support

Release	Modification
12.3(13)	This feature was introduced.

Feature History for SIP Media Inactivity Timer

Release	Modification
12.2(2)XB	This feature was introduced.
12.2(8)T	This feature was integrated into this release.
12.2(11)T	This feature was implemented on additional platforms.

Finding Support Information for Platforms and Cisco Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

- Prerequisites for SIP QoS, on page 603
- Restrictions for SIP QoS, on page 604
- Information About SIP QoS, on page 604
- How to Configure SIP QoS Features, on page 617
- Configuration Examples for SIP QoS Features, on page 649
- Additional References, on page 658

Prerequisites for SIP QoS

Measurement-Based Call Admission Control for SIP Feature

- By default, gateways support reliable provisional responses. That is, no additional configuration tasks are necessary to enable reliable provisional responses.

Note For information on configuring reliable provisional responses including enabling the feature again if it was disabled, see SIP Gateway Support of RSVP and TEL URL.

- Configure a basic VoIP network.
- Enable Service Assurance Agent (SAA) Responder on the originating and terminating gateway.

Note For information on configuring Service Assurance Agent, see Network Monitoring Using Cisco Service Assurance Agent.

Note For information about configuring VoIP, see Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.2. For information about configuring reliable provisional responses including reenabling the feature if it was disabled, see SIP Gateway Support of RSVP and TEL URL. For information about configuring Service Assurance Agent, see "Network Monitoring Using Cisco Service Assurance Agent".

SIP Gateway Support of RSVP and SIP Gateway Support of 'tel' URL Features

- Enable RSVP on the appropriate gateway interfaces by using the **ip rsvp bandwidth** command.

Note For details on the command, see the *Cisco IOS Quality of Service Solutions Command Reference, Release 12.3*.

- Enable weighted fair queuing (WFQ) on these interfaces by using the **fair-queue** command. This ensures that the voice packets get priority over the interface.

Note For details on the command, see the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3. For an example, see "SIP Gateway Support of RSVP and TEL URL Example".

- Set the desired and acceptable quality of service (QoS) levels in your dial peers by using the **req-qos** and **acc-qos** dial-peer configuration commands.

Bandwidth reservation is not attempted unless the desired QoS for the associated dial peer is set to **controlled-load** or **guaranteed-delay**. If the desired QoS level is set to the default of **best-effort**, bandwidth reservation is *not* attempted. With the **req-qos**command, synchronized RSVP is attempted for a SIP call as long as the desired (requested) QoS for the associated dial peer is set to **controlled-load** or **guaranteed-delay**.

Note For details on the commands, see the *Cisco IOS Voice Command Referenc*e, Release 12.3. For an example, see "SIP Gateway Support of RSVP and TEL URL Example".

Restrictions for SIP QoS

Enhanced Codec Support for SIP Using Dynamic Payloads Feature

Measurement-Based Call Admission Control for SIP Feature

- When detecting network congestion, the PSTN fallback feature does not affect an existing call; it affects only subsequent calls.
- Only a single calculated planning impairment factor (ICPIF) delay or loss value is allowed per system.
- A small additional call setup delay can be expected for the first call to a new IP destination.
- The Service Assurance Agent Responder feature, a network congestion analysis mechanism, cannot be configured for non-Cisco devices.

SIP Gateway Support of RSVP and SIP Gateway Support of 'tel' URL Features

- Bandwidth reservation (QoS) is not supported for Session Description Protocol (SDP changes between 183 Session Progress/180 Alerting and 200 OK responses.
- Bandwidth reservation (QoS) is not attempted if the desired QoS level is set to the default of **best-effort.**The desired QoS for the associated dial peer must be set to **controlled-load** or **guaranteed-delay**.
- Distributed Call Signaling (DCS) headers and extensions are not supported.
- SIP gateways do not support codecs other than those listed in the SIP codec table listed in "Additional Codec Support". When an unsupported codec is selected during configuration of the dial peers, the action taken depends on the selected gateway:
  - If on the originating gateway, an appropriate SIP debug trace is presented, indicating the failure to originate the SIP call leg.
  - If on the terminating gateway, an appropriate SIP response (4xx) with a warning indicating incompatible media types is sent.

Information About SIP QoS

To configure SIP QoS features, you should understand the following concepts:

Enhanced Codec Support for SIP Using Dynamic Payloads

The Enhanced Codec Support for SIP Using Dynamic Payloads feature enhances codec selection and payload negotiation between originating and terminating SIP gateways.

This feature offers the following benefits:

- Expanded dynamic payload support on Cisco IOS gateways, resulting in enhanced bandwidth control
- Expanded ability to advertise and negotiate all codecs available on a given platform
- Expanded interoperability and interconnectivity between gateways, applications, and services in the network

The feature provides the SIP enhancements described in the following sections:

Additional Codec Support

Codecs are a digital signal processor (DSP) software algorithm used to compress or decompress speech or audio signals. Previous implementations of the SIP stack on Cisco IOS gateways supported only a subset of the available codecs for each platform.

Support for codecs varies on different platforms. See the table below for a listing of SIP codec support by platform. Use the codec ? command to determine the codecs available on a specific platform.

Table 53: SIP Codec Support by Platform and Cisco IOS Release

Codec	Cisco 2600 Series, Cisco 3620, Cisco 3640, Cisco 3660	Cisco 7200 Series	Cisco AS5300	Cisco AS5350, Cisco AS5400, Cisco AS5850
Clear-channel	Yes	No	Yes	Yes
G711alaw	Yes	Yes	Yes	Yes
G711ulaw	Yes	Yes	Yes	Yes
G723ar53	Yes	Yes	Yes	Yes
G723ar63	Yes	Yes	Yes	Yes
G723r53	Yes	Yes	Yes	Yes
G723r63	Yes	Yes	Yes	Yes
G726r16	Yes	Yes	Yes	Yes
G726r24	Yes	Yes	Yes	Yes
G726r32	Yes	Yes	Yes	Yes
G728	Yes	Yes	Yes	No
G729br8	Yes	Yes	Yes	Yes
G729r8	Yes	Yes	Yes	Yes
GSM-EFR	Yes	No	Yes	No

Codec	Cisco 2600 Series, Cisco 3620, Cisco 3640, Cisco 3660	Cisco 7200 Series	Cisco AS5300	Cisco AS5350, Cisco AS5400, Cisco AS5850
GSM-FR	Yes	No/No	Yes	Yes

Payload Type Selection

Payload types define the content and format of Real-Time Transport Protocol (RTP) packets and the resulting stream of data generated by the RTP flow. The payload type defines the codec in use and is identified in the payload type field of the header of each RTP packet. There are two mechanisms for specifying payload type, static and dynamic.

Static payload types are assigned to specific RTP formats by RFC 1890 and these mappings are registered with the Internet Assigned Numbers Authority (IANA). Although not required, static payload types can also be mapped to RTP encodings using the rtpmap attribute. The following SIP-supported codecs have static payload values defined by the IANA:

- G711ulaw
- G711alaw
- G723r63
- G726r32
- G728
- G729r8
- GSM-FR

Dynamic payload values are used for codecs that do not have static payload values defined. Dynamic payload types do not have fixed mappings, and must be mapped to RTP encodings within the Session Description Protocol (SDP) itself using the a=rtpmap: line. The feature allows dynamic payload values to be used for the following codecs with no static payload values defined:

- Clear-channel
- G726r16
- G726r24
- GSM-EFR

Of the four codecs listed that allow dynamic payload values to be assigned, only the payload type for the clear-channel codec can be configured using the command-line interface (CLI). The remaining G.726r16, G.726r24 and GSM-EFR codecs are selected on a per-call basis by the SIP subsystem. The dynamic payload range is assigned by the IANA, with values from 96 to 127. The SIP subsystem looks for and uses the first value in the range that is both available and not reserved for Cisco IOS applications. Once a dynamic payload value is picked for a particular payload type, it cannot be used for other payload types. Of the 32 available IANA values, those reserved for special Cisco IOS applications are listed in the table below. To configure dynamic payload values for the payload types listed in the table, use the rtp payload-type command; otherwise the default values for the payload types are used.

Table 54: Default Dynamic Payload Values

Dynamic Payload Type	Default Dynamic Payload Value	Supported by SIP
Cisco-rtp-dtmf-relay	121	Yes
Named Signal Event	100	Yes
Named Telephony Event	101	Yes
Cisco-cas-payload	123	No
Cisco-clear-channel	125	No
Cisco-codec-fax-ack	97	No
Cisco-codec-fax-ind	96	No
Cisco-fax-relay	122	No
Cisco-pcm-switch-over-alaw	127	No
Cisco-pcm-switch-over-ulaw	126	No

Note After a dynamic payload value has been assigned from the reserved range, it cannot be used for any other payload types.

Advertising Codec Capabilities

The dynamic payload value selected by the SIP subsystem is advertised in the outgoing SIP INVITE request. The Enhanced Codec Support for SIP Using Dynamic Payloads feature supports dynamic payloads by expanding the SIP subsystem ability to advertise and negotiate available codecs. SIP uses the connection, media, and attribute fields of the SDP message during connection negotiation.

The feature supports the following Internet Engineering Task Force (IETF) drafts:

- draft-ietf-avt-rtp-mime-06.txt, MIME Type Registration of RTP Payload Formats (further developed and later published as RFC 3555) .
- draft-ietf-avt-profile-new-12.txt, RTP Profile for Audio and Video Conferences with Minimal Control (further developed and later published as RFC 3551) .

The following sample SIP INVITE message shows the payload value and codec selection resulting from the payload negotiation process. The media m= field includes the added payload value. The attribute a= field includes the selected codec. In this outgoing INVITE message, the first available dynamic payload value of 115 is selected by the SIP subsystem for a GSM-EFR codec.

```
INVITE sip:36602@172.18.193.120:5060;user=phone SIP/2.0
          Via: SIP/2.0/UDP 172.18.193.98:5060
          From: "36601" <sip:36601@172.18.193.98>
          To: <sip:36602@172.18.193.120;user=phone>
          Date: Mon, 01 Mar 1993 00:05:14 GMT
          Call-ID: 4326879A-14EF11CC-80069792-19DC655A@172.18.193.98
```

```
          Cisco-Guid: 1092278192-351211980-2147784594-433874266
          User-Agent: Cisco-SIPGateway/IOS-12.x
          CSeq: 101 INVITE
          Max-Forwards: 6
          Timestamp: 730944314
          Contact: <sip:36601@172.18.193.98:5060;user=phone>
          Expires: 180
          Content-Type: application/sdp
Content-Length: 228
          v=0
          o=CiscoSystemsSIP-GW-UserAgent 6973 8772 IN IP4 172.18.193.98
          s=SIP Call
          c=IN IP4 172.18.193.98
          t=0 0
          m=audio 17928 RTP/AVP 18 115
          a=rtpmap:18 G729/8000
          a=rtpmap:115 GSM-EFR/8000
```

G723 Codec Versions

In addition to the previously supported G.723r63 version of the G.723 codec, the feature supports the following versions:

- G723r53, where the number 53 indicates the bit rate of 5.3 kbps
- G723ar53, where the letter a indicates support for Annex A, which specifies voice activity detection (VAD)
- G723ar63, where the number 63 indicates a bit rate of 6.3 kbps

A static payload value of 4 is used for all versions of the G.723 codec.

Expanded codec support allows the originating and terminating gateways to advertise and negotiate additional codec capabilities. Cisco implements support for multiple G.723 codec versions by using a=fmtp and a=rtpmap attributes in the SDP body of outgoing INVITE requests to define the G.723 codec version. For the G.723 codec, the value of a=fmtp is 4 (the IANA assigned static value), and the annexa value is either yes or no. The default for annexa is yes.

The table below lists the possible codec configurations, that, taken together with Annex A support at the remote end, result in selecting the negotiated codec.

Table 55: G723 Codecs

Configured Codec(s)	Remote End Supports Annex A	Negotiated Codec
G723r63	annexa = no or no fmtp line	G723r63
G723r53	annexa = no or no fmtp line	G7223r53
G723r53 and G723r63	annexa = no or no fmtp line	G723r63
G723ar63	annexa=yes or no fmtp line	G723ar63
G723ar53	annexa=yes or no fmtp line	G723ar53
G723ar53 and G723ar63	annexa=yes or no fmtp line	G723ar63
G723ar53 and G723r53	annexa=yes or no fmtp line	G723ar53

Configured Codec(s)	Remote End Supports Annex A	Negotiated Codec
G723ar63 and G723r63	annexa=yes or no fmtp line	G723ar63
G723ar63 and G723r53	annexa=yes or no fmtp line	G723ar63
G723ar53 and G723r63	annexa=yes or no fmtp line	G723ar63
G723ar53, G723r53, G723ar63, and G723r63	annexa = no or no fmtp line	G723ar63

The following partial SDP body shows the media m= field and attribute a= field for a gateway with G.723 codecs and Annex A specified.

```
m=audio 62986 RTP/AVP 4
a=rtpmap:4 G723/8000
a=fmtp:4 annexa=yes
```

G729 Codec Versions

The feature supports the following versions of G.729 codecs:

- G729r8, where r8 indicates the bit rate of 8 kbps
- G729br8, where b indicates support for Annex B, which specifies VAD, Discontinuity Transmission (DTX), and Comfort Noise generation (CNG).

A static payload value of 18 is used for all versions of the G.729 codec.

Cisco implements support for multiple G.729 codec versions by using a=fmtp and a=rtpmap attributes in the SDP body of outgoing INVITE requests. For the G.729 codec, the value of a=fmtp is 18 (the IANA assigned static value), and the annexb value is either yes or no. The default for annexb is yes.

The table below lists the possible codec configuration that, taken together with Annex B support at the remote end, result in selecting the negotiated codec.

Table 56: G729 Codecs

Configured Codec(s)	Remote End Supports Annex B	Negotiated Codec
G729r8	annexb= no or no fmtp line	G729r8
G729br8	annexb = yes or no fmtp line	G729br8
G729r8 and G729br8	annexb= yes or no fmtp line	G729br8
G729r8 and G729br8	no fmtp line	G729br8
G729r8 and G729br8	annexb=no or no fmtp line	G729r8
G729r8 and G729br8	annexb=yes	G729br8

The following partial SDP body shows the media m= field and attribute a= field for a gateway with G.729 codecs and Annex B specified:

```
m=audio 17928 RTP/AVP 18
```

```
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
```

Measurement-Based Call Admission Control for SIP

The Measurement-Based Call Admission Control for SIP feature implements support within SIP to monitor IP network capacity and reject or redirect calls based on congestion detection.

Feature benefits include the following:

- PSTN fallback
  - Automatically routes a call to an alternate destination when the data network is congested at the time of call setup, thereby enabling higher call completion rates.
  - Enables the service provider to give a reasonable guarantee about the quality of the conversation to VoIP users at the time of call admission.
  - PSTN fallback provides network congestion measurement, including delay, jitter, and packet loss information for the configured IP addresses.
  - A new call need not wait for probe results before being admitted, thereby minimizing delays.
- Call admission control
  - Configurable call treatment allows the Internet service provider (ISP) the flexibility to configure how the call will be treated when local resources to process the call are not available.
  - Resource unavailable signaling allows you to automatically busy out channels when local resources are not available to handle the call.
  - User-selected thresholds allow you the flexibility to configure thresholds to determine resource availability.

The Measurement-Based Call Admission Control for SIP feature does the following:

- Verifies that adequate resources are available to carry a successful VoIP session.
- Implements a mechanism to prevent calls arriving from the IP network from entering the gateway when required resources are not available to process the call.
- Supports measurement-based call admission control (CAC) processes.

Before the CAC feature was developed, gateways did not have a mechanism to check for IP network congestion and resource unavailability. Although quality of service (QoS) mechanisms provide a level of low latency and guaranteed delivery that is required for voice traffic, CAC mechanisms are intended to extend the capabilities of QoS to protect voice traffic from being negatively affected by other voice traffic. CAC is used to gracefully deny network access under congestion conditions and provide alternative call rerouting to prevent dropped or delayed calls. There are a variety of CAC mechanisms, including the following:

- Measurement-based CAC, which uses probes to look ahead into the packet network to gauge the state of the network to determine whether to allow a new call.
- Resource-based CAC, which calculates resources needed for the call, determines their availability, and reserves those resources.

The Cisco IOS VoiceXML feature provides an alternative to Resource Reservation Protocol (RSVP) for VoIP service providers that do not deploy RSVP.

The new feature implements measurement-based CAC using the mechanisms described in the following sections:

Service Assurance Agents

Service Assurance Agents (SAA) is a generic network management feature that provides a mechanism for network congestion analysis. SAA determine latency, delay, and jitter and provides real-time ICPIF calculations before establishing a call across an IP infrastructure. The SAA Responder feature uses SAA probes to traverse the network to a given IP destination and measure the loss and delay characteristics of the network along the path traveled. These values are returned to the outgoing gateway to use in making a decision on the condition of the network and its ability to carry a call. Threshold values for rejecting a call are configured at the outgoing gateway (see "PSTN_Fallback").

Each probe consists of multiple packets, a configurable parameter of this feature. SAA packets emulate voice packets and receive the same priority as voice throughout the entire network. The delay, loss, and ICPIF values entered into the cache for the IP destination are averaged from all the responses. If the call uses G.729 and G.711 codecs, the probe packet sizes mimic those of a voice packet for that codec. Other codecs use G.711-like probes. In Cisco IOS software releases later than Release 12.1(3)T, other codec choices may also be supported with their own specific probes.

The IP precedence of the probe packets can also be configured to simulate the priority of a voice packet more closely. This parameter should be set equal to the IP precedence used for other voice media packets in the network.

SAA probes used for CAC go out randomly on ports selected from within the top end of the audio User Datagram Protocol (UDP) defined port range (16384 to 32767). Probes use a packet size based on the codec the call will use. IP precedence can be set if desired, and a full Realtime Transport Protocol (RTP), UDP, or IP header is used just as a real voice packet would carry. The SAA Responder feature was called Response Time Reporter (RTR) in earlier releases of Cisco IOS software.

The SAA Responder feature can not be configured for non-Cisco devices. For a complete description of SAA configuration, see the Cisco IOS Configuration Fundamentals and Network Management Configuration Guide, Release 12.3.

Calculated Planning Impairment Factor

The Cisco IOS VoiceXML feature supports the determination of ICPIF, as specified by International Telecommunications Union (ITU) standard G.113. The SIP subsystem calculates an impairment factor for network conditions to a particular IP address. ICPIF checks for end-to-end resource availability by calculating a Total Impairment Value, which is a function of codecs used and loss or delay of packets. You can configure router resources to make call admission decisions, using either the ICPIF threshold, or by setting delay and loss thresholds.

Configurable ICPIF values that represent the ITU specification for quality of voice as described in G.113 are the following:

- 5--Very good
- 10--Good
- 20--Adequate
- 30--Limiting case
- 45--Exceptional limiting case
- 55--Customers likely to react strongly

The default value is 20. SAA probe delay and loss information is used in calculating an ICPIF value, which is then used as a threshold for CAC decisions. You can base such decisions on either the ITU interpretation described or on the requirements of an individual customer network.

PSTN Fallback

The Cisco IOS VoiceXML feature supports PSTN Fallback, which monitors congestion in the IP network and either redirects calls to the public switched telephone network (PSTN) or rejects calls based on network congestion. Calls can be rerouted to an alternate IP destination or to the PSTN if the IP network is found unsuitable for voice traffic at that time. You can define congestion thresholds based on the configured network. This functionality allows the service provider to give a reasonable guarantee about the quality of the conversation to VoIP users at the time of call admission.

Note PSTN Fallback does not provide assurances that a VoIP call that proceeds over the IP network is protected from the effects of congestion. This is the function of the other QoS mechanisms, such as IP Real-Time Transport Protocol (RTP) priority or low latency queueing (LLQ).

PSTN Fallback includes the following capabilities:

- Provides the ability to define the congestion thresholds based on the network.
  - Defines a threshold based on ICPIF, which is derived as part of ITU G.113 (see "Service Assurance Agents").
  - Defines a threshold based solely on packet delay and loss measurements.
- Uses SAA probes to provide packet delay, jitter, and loss information for the relevant IP addresses. Based on the packet loss, delay, and jitter encountered by these probes, an ICPIF or delay or loss value is calculated.
- Supports calls of any codec. Only G.729 and G.711 have accurately simulated probes. Calls of all other codecs are emulated by a G.711 probe.

The call fallback subsystem has a network traffic cache that maintains the ICPIF or delay or loss values for various destinations. This capability helps performance, because each new call to a well-known destination need not wait for a probe to be admitted because the value is usually cached from a previous call.

Once the ICPIF or delay or loss value is calculated, they are stored in a fallback cache where they remain until the cache ages out or overflows. Until an entry ages out, probes are sent periodically for that particular destination. This time interval is configurable.

Media Information for Fallback Services

SIP reliable provisional responses ensure that media information is exchanged and that resource and network checks can take place prior to connecting the call. The following SIP methods have been implemented to support fallback services:

- INVITE with Session Description Protocol (SDP) body. The PSTN Fallback feature provides support for a new attribute line, a=rtr, in the SDP message body. The rtr attribute enables support for invoking fallback services. The INVITE message with SDP body provides media connection information, including IP address and negotiated codec.
- Provisional Acknowledgment (PRACK). PRACK allows reliable exchanges of SIP provisional responses between SIP endpoints. When the INVITE message has no SDP body, that is, no delayed media, the

terminating gateway sends media information in the 183 Session Progress message and expects the SDP from the originating gateway in the PRACK message.

- Conditions Met (COMET), which indicates if the preconditions for a given call have been met.

Call Admission Thresholds

User-selected thresholds allow you to configure call admission thresholds for local resources and end-to-end memory and CPU resources. You can configure two thresholds, high and low, for each global or interface-related resource. The specified call treatment is triggered when the current value of a resource goes beyond the configured high, and remains in effect until the current resource value falls below the configured low.

Call Treatment Options

You can select how the call should be treated when local resources are not available to handle the call. For example, when the current resource value for any one of the configured triggers for call threshold exceeds the configured threshold, you have the following the call treatment choices:

- TDM hairpinning--Hairpins the calls through the POTS dial peer.
- Reject--Disconnects the call.
- Play message or tone--Plays a configured message or tone to the user.

Resource Unavailable Signaling

The Resource Unavailable Signaling feature supports autobusyout capability, which busies out channels when local resources are not available to handle the call. Autobusyout is supported on both channel-associated signaling (CAS) and primary rate interface (PRI) channels:

- CAS--Uses busyout to signal local resources are unavailable.
- PRI--Uses either service messages or disconnect with correct cause-code to signal resources are unavailable.

SIP Gateway Support of RSVP and TEL URL

The SIP Gateway Support of RSVP and TEL URL feature provides the following SIP enhancements:

This section describes the SIP Gateway Support of RSVP and the SIP Gateway Support of ‘tel’ URL features. SIP gateways can enable resource reservation using Resource Reservation Protocol (RSVP). Resource reservation on SIP gateways synchronizes RSVP procedures with SIP call establishment procedures, ensuring that the required quality of service (QoS) for a call is maintained across the IP network.

Feature benefits include the following:

- SIP Gateway Support of RSVP and TEL URL enables Quality of Service (QoS), ensuring certain bandwidth reservations for specific calls. The bandwidth reservation can be **best-effort**, in which case the call is completed even if the reservation is not supported by both sides or cannot be established. Or the bandwidth reservation can be *required* , and the call is not set up if the bandwidth reservation is not performed successfully.

- With the reliable provisional response features, you can ensure that media information is exchanged and resource reservation takes place before connecting a call.
- Gateways now accept TEL calls sent through the Internet, which provides interoperability with other equipment that uses TEL URL. The TEL URL feature also gives service providers a way to differentiate services based on the type of call, allowing for deployment of specific services.

RSVP

Before this feature was implemented, SIP applications over IP networks functioned as best-effort services--their media packets were delivered with no performance guarantees. However, SIP gateway support of RSVP and TEL URL ensures quality of service (QoS) by coordinating SIP call signaling and RSVP resource management. This feature reserves sufficient network-layer resources to guarantee bandwidth and bounds on packet loss, delay, and jitter; thus ensuring that the called party's phone rings only after bandwidth required for the call has been successfully reserved.

Additionally, appropriate changes to the resources reserved for the SIP call are made when mid-call INVITE messages, requiring media change (such as a change of codec) are requested.

Synchronization with Cisco IOS QoS

A QoS module is provided that acts as a broker between the VoIP service-provider interfaces (SPIs) and the Cisco IOS RSVP subsystem. The QoS module enables the VoIP SPIs to initiate resource reservation, modify parameters of an existing reservation, and clean up the reserved resources. The QoS module then communicates the results of the operation to the RSVP subsystem.

The conditions for SIP calls using QoS are as shown in the table below.

Table 57: Conditions for SIP Calls Using QoS

SIP Call Setup	Result
Bandwidth reservation (QoS) is attempted when:	The desired (requested) QoS for the associated dial peer is set to **controlled-load** or **guaranteed-delay**.
Bandwidth reservation (QoS) is not attempted when:	The desired QoS level is set to the default of **best-effort**.
If bandwidth reservation (QoS) is attempted but fails, the acceptable QoS for the dial peer determines the outcome of the call: --	The call proceeds without any bandwidth reservation in place if the acceptable QoS is configured with **best-effort**.
	The call is released if the acceptable QoS on either gateway is configured with **controlled-load** or **guaranteed-delay**.

The desired QoS and acceptable QoS are configured through Cisco IOS software by using the **req-qos** and **acc-qos** dial-peer configuration commands, respectively.

TEL URL Format in SIP Messages

The SIP Gateway Support of RSVP and TEL URL feature also supports Telephone Uniform Resource Locators or TEL URL. Currently SIP gateways support URLs in the SIP format. SIP URLs are used in SIP messages to indicate the originator, recipient, and destination of the SIP request. However, SIP gateways may also encounter URLs in other formats, such as TEL URLs. TEL URLs describe voice call connections. They also

enable the gateway to accept TEL calls sent through the Internet, and to generate TEL URLs in the request line of outgoing INVITEs requests.

SIP and TEL URL Examples

SIP URL

A SIP URL identifies a user's address and appears similar to an email address: user@host where *user* is the telephone number and *host* is either a domain name or a numeric network address. For example, the request line of an outgoing INVITE request might appear as:

```
INVITE sip:5550100@
example
.com; user=phone.
```

The *user=phone* parameter distinguishes that the user address is a telephone number rather than a username.

TEL URL

A TEL URL takes the basic form of *tel:telephone subscriber number* , where *tel* requests the local entity to place a voice call, and *telephone subscriber number* is the number to receive the call. For example: tel:+555-0100

For more detailed information on the structure of TEL URL, see RFC 2806, *URLs for Telephone Calls.*

Reliability of SIP Provisional Responses

SIP reliable provisional responses ensure that media information is exchanged and resource reservation can take place prior to connecting the call. Provisional acknowledgement (PRACK) and conditions met (COMET) are two methods that have been implemented.

PRACK allows reliable exchanges of SIP provisional responses between SIP endpoints. COMET indicates if the preconditions for a given call or session have been met.

SIP Hold Timer Support

The SIP: Hold Timer Support feature provides the ability to terminate a call that has been placed on hold in excess of a configurable time period, and to thereby free up trunk resources.

Feature benefits include the following:

- Improved trunk resource utilization
- Improved network monitoring and management capability

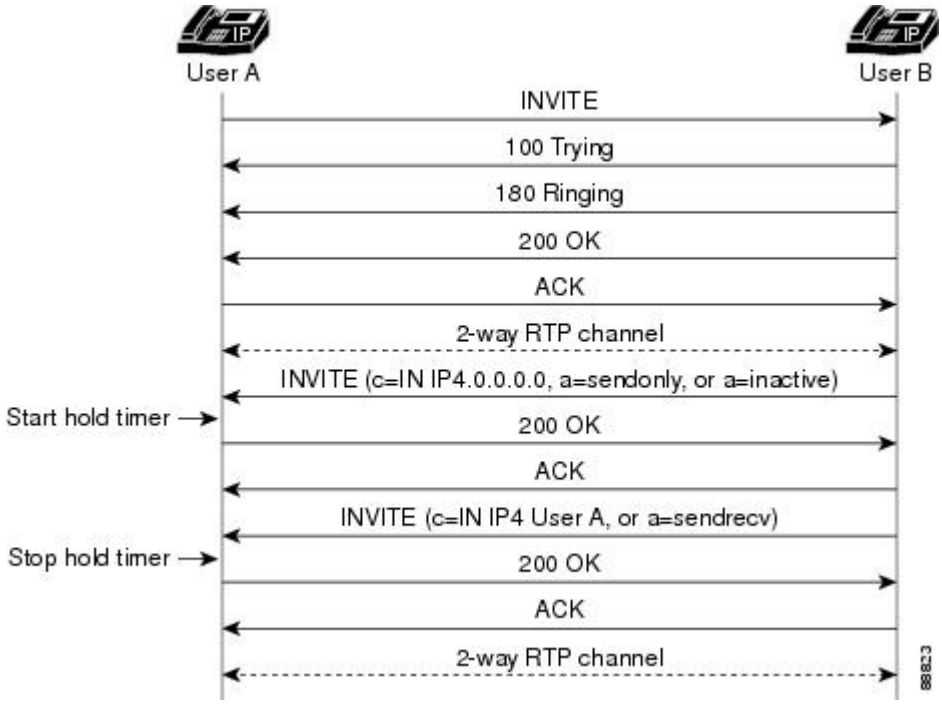
The SIP: Hold Timer Support feature provides a new configurable hold timer that allows you to specify a maximum hold time of up to 2880 minutes. Prior to this feature, there was no mechanism to automatically disconnect a call that had been on hold for a set period of time. When the SIP call hold process occurs in response to ISDN Suspend and Resume messages, a media inactivity timer allows a gateway to monitor and disconnect a VoIP call if no Real-Time Control Protocol (RTCP) packets are received within a configurable time period. This timer is deactivated when a call is placed on hold and no media packets are sent. As a result, a call is potentially allowed to stay on hold indefinitely.

Note For information on the media inactivity timer, see SIP Media Inactivity Timer *and* SIP: ISDN Suspend/Resume Support.

The SIP: Hold Timer Support feature resolves this problem by allowing you to configure a gateway to disconnect a held call when the hold timer is exceeded. The hold timer is activated when a gateway receives a call hold request from the other endpoint, for example, a SIP phone. SIP gateways receive notice of a call hold when the originating gateway sends a re-INVITE to the terminating gateway containing one of the following Session Description Protocol (SDP) lines: a connection IP address set to 0.0.0.0 (c=0.0.0.0), or the attribute field set to send only (a=sendonly) or to inactive (a=inactive). When the SIP phone or user-agent client cancels the hold, the originating gateway takes the call off hold by sending a re-INVITE with the attribute field (a=) set to sendrec or with the connection field (c=) set back to the actual IP address of the remote SIP entity, in place of 0.0.0.0.

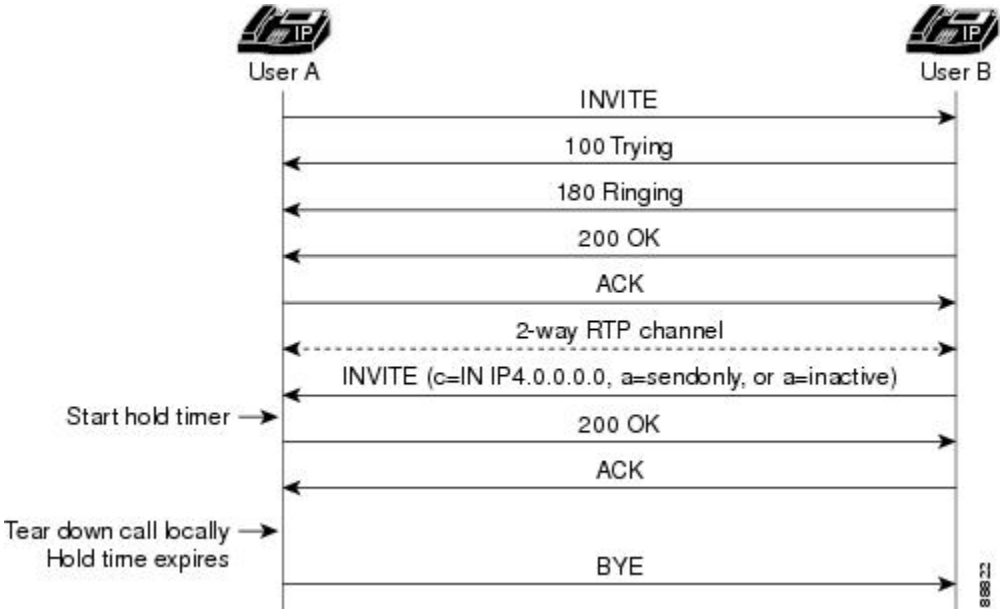
The following call flows show gateway behavior upon receiving a call hold request from a SIP endpoint. In the figure below, the originating gateway sends an INVITE with an indication to place a call on hold (c=IN IP4.0.0.0.0, a=sendonly, or a=inactive in the SDP), which starts the hold timer. When the gateway on hold receives a re-INVITE with the indication to resume the call (c=IN IP4 User A or a=sendrecv), it stops the hold timer, sends a 200 OK, and resumes the call.

Figure 87: Start and Stop Hold Time



In the figure below, the hold timer expires, the gateway on hold tears down the call and sends a BYE request to the other end.

Figure 88: Hold Timer Expiration



SIP Media Inactivity Timer

The SIP Media Inactivity Timer feature enables Cisco gateways to monitor and disconnect VoIP calls if no Real-Time Control Protocol (RTCP) packets are received within a configurable time period.

When RTCP reports are not received by a Cisco gateway, the SIP Media Inactivity Timer feature releases the hung session and its network resources in an orderly manner. These network resources include the gateway digital signal processor (DSP) and time-division multiplexing (TDM) channel resources that are utilized by the hung sessions. Because call signaling is sent to tear down the call, any stateful SIP proxies involved in the call are also notified to clear the state that they have associated with the hung session. The call is also cleared back through the TDM port so that any attached TDM switching equipment also clears its resources.

Feature benefits include the following:

- Provides a mechanism for detecting and freeing hung network resources when no RTCP packets are received by the gateway.

How to Configure SIP QoS Features

For help with a procedure, see the verification and troubleshooting sections listed above. Before you perform a procedure, familiarize yourself with the following information:

Configuring Enhanced Codec Support for SIP Using Dynamic Payloads

Note This procedure is optional and selects a dynamic payload value from the IANA defined range of 96 to 127.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voip** number
4. **rtp payload-type** *type number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voip** number **Example:** `Router(config)# dial-peer voip 110`	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 4	**rtp payload-type** *type number* **Example:** `Router(config-dial-peer)# rtp payload-type nte 125`	Identifies the payload type of a RTP packet. Arguments are as follows: • *type number* --Payload type. Valid values are the following: • cisco-cas-payload--Cisco CAS RTP payload • **cisco-clear-channel**--Cisco clear-channel RTP payload • **cisco-codec-fax-ack** --Cisco codec fax acknowledge • **cisco-codec-fax-ind**--Cisco codec fax indication • **cisco-fax relay**--Cisco fax relay • **cisco-pcm-switch-over-alaw**--Cisco RTP PCM codec switch over indication (a-law) • **cisco-pcm-switch-over-ulaw**--Cisco RTP PCM codec switch over indication (u-law) • **cisco-rtp-dtmf-relay**--Cisco RTP DTMF relay • **nte**--Named telephone event • **nse**--Named signaling event • *number* --Payload identity. Range: 96 to 127. Default: 101.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Measurement-Based Call Admission Control for SIP

Configure SAA Responder

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rtr responder**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**rtr responder** **Example:** `Router(config)# rtr responder`	Enables SAA Responder functionality on a device.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configure PSTN Fallback

Note PSTN fallback configuration applies to both inbound and outbound gateways. In most networks, gateways generate calls to each other, so that every gateway is both an outgoing gateway and a terminating gateway.

- Configure the destination node, which is often but not necessarily the terminating gateway, with the SAA Responder feature.
- PSTN fallback configuration is done at the global level and therefore applies to all calls attempted by the gateway. You cannot selectively apply PSTN fallback only to calls initiated by specific PSTN or PBX interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call fallback active**
4. **call fallback cache size** *number*
5. **call fallback instantaneous-value-weight** *weight*
6. **call fallback jitter-probe num-packets** *number-of-packets*
7. **call fallback jitter-probe precedence** *precedence-value*
8. **call fallback jitter-probe priority-queue**
9. **call fallback threshold delay** *delay-value* loss *loss-value*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call fallback active** **Example:** `Router(config)# call fallback active`	Enables a call request to fall back to alternate dial peers in case of network congestion.
Step 4	**call fallback cache size** *number* **Example:**	Specifies the call-fallback cache size for network traffic probe entries. The argument is as follows:

	Command or Action	Purpose
	`Router(config)# call fallback cache size 128`	• **number** --Cache size, in number of entries. Range: 1 to 256. Default: 128.
Step 5	**call fallback instantaneous-value-weight** *weight* **Example:** `Router(config)# call fallback instantaneous-value-weight 50`	Configures the call-fallback subsystem to determine an average value based on the last two probes registered in the cache for call requests. This command allows the call-fallback subsystem to recover gradually from network congestion conditions. The argument is as follows: • *weight* --By percent, when a new probe is received, how much to rely upon the new probe as opposed to the previous cache entry. The configured weight applies to the new probe first. Range: 0 to 100. Default: 66.
Step 6	**call fallback jitter-probe num-packets** *number-of-packets* **Example:** `Router(config)# call fallback jitter-probe num-packets 15`	Specifies the number of packets in a jitter probe used to determine network conditions. The argument is as follows: • *number-of-packets* --Number of packets. Range: 2 to 50. Default: 15.
Step 7	**call fallback jitter-probe precedence** *precedence-value* **Example:** `Router(config)# call fallback jitter-probe precedence 2`	Specifies the priority of the jitter-probe transmission by setting the IP precedence of IP packets. The argument is as follows: • *precedence-value* --Jitter-probe precedence. Range: 0 to 6. Default: 2.
Step 8	**call fallback jitter-probe priority-queue** **Example:** `Router(config)# call fallback jitter-probe priority-queue`	Assigns a priority queue for jitter-probe transmissions. You must set IP priority queueing for UDP voice ports 16384 to 32767.
Step 9	**call fallback threshold delay** *delay-value* loss *loss-value* **Example:** `Router(config)# call fallback threshold delay 36000 loss 50`	Configures the call-fallback threshold to use only specified packet delay and loss values. Arguments are as follows: • *delay-value* --Delay value, in ms. Range: 1 to 2147483647. No default. • *loss-value* --Loss value, as a percentage. Range: 0 to 100. No default.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configure Resource-Availability Check

To enable resource-availability checking, perform one of the following tasks:

Configuring Global Resources

To configure resource availability checking for global resources, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call threshold global** *trigger-name* low value high value [bus**y**out | treatment]
4. **call treatment** {**on** | **action** *action* [*value*] | **cause-code** *cause-code* | **isdn-reject** *value*}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call threshold global** *trigger-name* low value high value [bus**y**out | treatment] **Example:** `Router(config)# call threshold global total-calls low 5 high 1000 busyout`	Enables a trigger and define associated parameters to allow or disallow new calls on the router. Action is enabled when the trigger value exceeds the value specified by the **high** keyword and is disabled when it drops below the value specified by the low keyword. Keywords and arguments are as follows: • *trigger-name* --Global resources on the gateway to be used as call admission or utilization triggers. Valid values are the following: • cpu-5sec--CPU utilization in the last 5 seconds • cpu-avg--Average CPU utilization • io-mem--IO memory utilization • proc-mem--Processor memory utilization • total-calls--Total number of calls • total-mem--Total memory utilization • **low** *value* --Low threshold. Range: 1 to 100 percent for utilization triggers and 1 to 10000 for total-calls. • **high** *value* --High threshold: Range: 1 to 100 percent for utilization triggers and 1 to 10000 for total-calls.

	Command or Action	Purpose
		• **busyout** --Busy out the T1 or E1 channels if the resource is not available • **treatment** --Apply call treatment from the session application if the resource is not available
Step 4	**call treatment** {**on** | **action** *action* [*value*] | **cause-code** *cause-code* | **isdn-reject** *value*} **Example:** `Router(config)# call treatment action cause-code 17`	Specifies how calls should be processed when local resources are unavailable. Keywords and arguments are as follows: • **on** --Enable call treatment from the default session application • **action** --Action to be taken when call treatment is triggered. Valid values are as follows: • hairpin--Hairpinning action • playmsg--The gateway plays the selected message. The optional value argument specifies the audio file to play in URL format. • reject--The call should be disconnected and the ISDN cause code passed. • **cause-code** --Reason for disconnection to the caller. Valid values are as follows: • busy--Gateway is busy. • no-QoS--Gateway cannot provide quality of service (QOS). • no-resource--Gateway has no resources available. • **isdn-reject** *value* --For ISDN interfaces only, the ISDN reject cause code. Range: 34 to 47 (ISDN cause code for rejection).
Step 5	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configuring Interface Resources

To configure resource availability checking for interface resources, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call threshold interface** *interface-name interface-number* int-calls low value high value
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**call threshold interface** *interface-name interface-number* int-calls low value high value **Example:** `Router(config)# call threshold interface ethernet 0 int-calls low 5 high 2500`	Specifies threshold values for total numbers of voice calls placed through a particular interface. Use it also to allow or disallow admission for new calls on the router. Keywords and arguments are as follows: • *interface-name* --Interface used in making call admission decisions. Types of interfaces and their numbers depend upon the configured interfaces. • *interface-number* --Number of calls through the interface that triggers a call admission decision. • **int-calls** --Use the number of calls through the interface as a threshold. • **low** *value* --Value of low threshold, in percent. Range: 1 to 100 for the utilization triggers and 1 to 10000 calls for int-calls. • **high** *value* --Value of high threshold, in percent. Range: 1 to 100 for the utilization triggers and 1 to 10000 calls for int-calls.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configure SIP Reliable Provisional Response

Note By default, gateways support reliable provisional responses. That is, no additional configuration tasks are necessary to enable reliable provisional responses. This task enables reliable provisional response if it was disabled using the **no rel1xx** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **rel1xx** {**supported** *value* | **require** *value* | **disable**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode.
Step 4	**sip** **Example:** `Router (config-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**rel1xx** {**supported** *value* | **require** *value* | **disable**} **Example:** `Router(config-srv-sip)# rel1xx supported 100rel` **Example:** **Example:** `Router(config-srv-sip)# rel1xx require 100rel` **Example:** **Example:** `Router(config-srv-sip)# rel1xx disable`	Enables all SIP provisional responses (other than 100 Trying) to be sent reliably to the remote SIP endpoint. • **supported** *value* --Use provisional responses and you set the value; for instance, 100rel. The value argument may have any value, as long as both the user-agent client (UAC) and user-agent server (UAS) configure it the same. Default value is supported with the 100rel value. • **require** *value* --Use provisional responses and you set the value; for instance, 100rel. The value argument may have any value, as long as both the UAC and UAS configure it the same. • disable--Disables the use of reliable provisional responses.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-srv-sip)# exit`	Exits the current mode.

Configuring SIP Gateway Support of RSVP and TEL URL

This section contains the following procedures (you must perform them in the order listed):

Configure SIP Gateway Support of RSVP

Configuring Fair Queuing and RSVP

To configure fair queuing and RSVP, perform the following steps.

Note For details on these commands, see the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3. For an example, see the "SIP Gateway Support of RSVP and TEL URL Example".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **ip rsvp bandwidth** [*interface-kbps*[*single-flow-kbps*]]
5. **fair-queue** [*congestive-discard-threshold* [*dynamic-queues* [*reservable-queues*]]]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *number* **Example:** `Router(conf)# interface fastethernet 1`	Selects a particular Fast Ethernet interface for configuration. The argument is as follows: • *number* --Port, connector, or interface card number. On a Cisco 4500 or Cisco 4700 series, the

	Command or Action	Purpose
		network-interface module or network-processor-module number. Numbers are assigned at the factory at the time of installation or when added to a system.
Step 4	**ip rsvp bandwidth** [*interface-kbps*[*single-flow-kbps*]] **Example:** `Router(conf-if)# ip rsvp bandwidth 100 100`	Enables resource reservation protocol for IP on an interface. Arguments are as follows: • *interface-kbps* --Maximum amount of bandwidth, in kbps, that may be allotted by RSVP flows. Range: 1 to 10000000. • *single-flow-kbps* --Maximum amount of bandwidth, in kbps, that may be allocated in a single flow. Range: 1 to 10000000.
Step 5	**fair-queue** [*congestive-discard-threshold* [*dynamic-queues* [*reservable-queues*]]] **Example:** `Router(config-if)# fair-queue 32 16 100`	Enables weighted fair queuing for an interface. Arguments are as follows: • *congestive-discard-threshold* --Number of messages allowed in each queue. When a conversation reaches this threshold, new message packets are discarded. Valid values: powers of 2 in the range from 16 to 4096. Default: 64. • *dynamic-queues* --Number of dynamic queues used for best-effort conversations (that is, a normal conversation not requiring any special network services). Valid values: 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096. See tables in the fair-queue (class-default) command for the default number of dynamic queues. • *reservable-queues* --Number of reservable queues used for reserved conversations. Reservable queues are used for interfaces configured for features such as RSVP. Range: 0 to 1000. Default: 0.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits the current mode.

Configuring QoS Levels

To configure desired and acceptable QoS levels, perform the following steps.

Note For details on these commands, see the *Cisco IOS Voice Command Reference*, Release 12.3. For an example, see "SIP Gateway Support of RSVP and TEL URL Example".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **acc-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**}
5. **req-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 10 voip`	Enter VoIP dial-peer configuration modes for the specified VoIP dial peer.
Step 4	**acc-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**} **Example:** `Router(config dial-peer)# acc-qos best-effort`	Defines the acceptable QoS for any inbound and outbound call on a VoIP dial peer. Keywords are as follows: • **best-effort** --RSVP makes no bandwidth reservation. This is the default. • **controlled-load** --RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded. • **guaranteed-delay** -- RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queuing if the bandwidth reserved is not exceeded.
Step 5	**req-qos** {**best-effort** | **controlled-load** | **guaranteed-delay**} **Example:** `Router(config dial-peer)# req-qos best-effort`	Specifies the desired QoS to be used in reaching a specific dial peer. Keywords are as above.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config dial-peer)# exit`	Exits the current mode.

Configure SIP Gateway Support of TEL URL

Configuring TEL URLs for All VoIP SIP Calls

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **url** {**sip** | **tel**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Specifies the voice encapsulation type.
Step 4	**sip** **Example:** `Router(config-voi-srv)# sip`	Enters SIP configuration mode.
Step 5	**url** {**sip**	**tel**} **Example:** `Router(conf-serv-sip)# url sip`	Configures URLs to either the SIP or TEL format for your VoIP SIP calls. Keywords are as follows: • **sip** --Generate URLs in SIP format for VoIP calls. This is the default.

	Command or Action	Purpose
		• **tel** --Generate URLs in TEL format for VoIP calls.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring TEL URLs for All Dial-Peer SIP Calls

Note The **voice-class sip url command in** dial-peer configuration mode takes precedence over **the url** command **in global configuration**. However, if the **voice-class sip url** command contains the configuration of **system**, the gateway uses what was globally configured under the **url** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip url {sip | sips | system | tel}**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 4	**voice-class sip url {sip | sips | system | tel}** **Example:** `Router(config-dial-peer)# voice-class sip url sip`	Configures URLs to either the SIP or TEL format for your dial-peer SIP call. Keywords are as follows: • **sip** --Generate URLs in the SIP-format for calls on a dial-peer basis.

	Command or Action	Purpose
		• **sips** --Generate URLs in the SIPS-format for calls on a dial-peer basis. • **system** --Use the system value. This is the default. • **tel** --Generate URLs in the TEL format for calls on a dial-peer basis.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configure Reliability of SIP Provisional Responses

The following are tasks for configuring reliability of SIP provisional responses:

By default, gateways support reliable provisional responses. That is, no additional configuration tasks are necessary to enable reliable provisional responses.

However, there are instances when you may want control over the use of reliable provisional responses. For example, you may want to:

- Always require the use of reliable provisional responses (use the Required header)
- Never use reliable provisional responses

In these cases, there are two ways to configure reliable provisional responses:

- Dial-peer mode. In this mode you can configure reliable provisional responses for the specific dial peer only. Configure with the **voice-class sip rel1xx** command.
- SIP mode. In this mode you can configure reliable provisional responses globally. Configure with the **rel1xx** command.

When the **voice-class sip rel1xx** command under dial-peer configuration is configured, it takes precedence over **the global configuration of the rel1xx** command. However, if the **voice-class sip rel1xx** command contains the configuration of **system**, the gateway uses what was globally configured under the **rel1xx** command.

The table below shows the possible configurations achieved with the **voice-class sip rel1xx** and the **rel1xx** commands.It outlines the possible configurations on both the originating gateway and the terminating gateway, and the results of the various configurations.

Note When configured with the **supported** option, the SIP gateway uses the Supported header in outgoing INVITE messages. When configured with the **require** option, the SIP gateway uses the Required header in outgoing INVITE messages.

Table 58: Configuration Results Based on Originating and Terminating Gateway Configurations

Originating Gateway	Terminating Gateway	Result
supported 100rel	supported 100rel	Reliable provisional responses
supported 100rel	require 100rel	Reliable provisional responses
supported 100rel	disable	No reliable provisional responses, call proceeds
require 100rel	supported 100rel	Reliable provisional responses
require 100rel	require 100rel	Reliable provisional responses
require 100rel	disable	Call fails. TG sends 420 with “Unsupported: 100rel” header
disable	supported 100rel	No reliable provisional responses
disable	require 100rel	No reliable provisional responses
disable	disable	No reliable provisional responses

Configuring Specific Reliable Provisional Responses

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip rel1xx** {**supported** *value* | **require** *value* | **system** | **disable**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 29 voip`	Enter dial-peer configuration mode for the specified VoIP dial peer.

	Command or Action	Purpose
Step 4	**voice-class sip rel1xx** {**supported** *value* | **require** *value* | **system** | **disable**} **Example:** `Router(config-dial-peer)# voice-class sip rel1xx supported 100rel`	Enables all SIP provisional responses (other than 100 Trying) to be sent reliably to the remote SIP endpoint. Keywords and arguments are as follows: • **supported** *value* --Use provisional responses and you set the value; for instance, **100rel**. The value argument may have any value, as long as both the user-agent client (UAC) and user-agent server (UAS) configure it as the same. • **require** value--Use provisional responses and you set the value; for instance, **100rel**. The value argument may have any value, as long as both the UAC and UAS configure it the same. • **system** --Use the value configured in voice service mode. Default is the system value. • **disable** --Disable the use of rel1xx provisional responses.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Global Reliable Provisional Responses

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **rel1xx** {**supported** *value* \| **require** *value* \| **disable**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice-service configuration mode for VoIP.
Step 4	**sip** **Example:** `Router(config-voi-srv)# sip`	Enters SIP configuration mode.
Step 5	**rel1xx** {**supported** *value* | **require** *value* | **disable**} **Example:** `Router(config-srv-sip)# rel1xx supported 100rel`	Enables all SIP provisional responses (other than 100 Trying) to be sent reliably to the remote SIP endpoint. • **supported** *value* --Use provisional responses and you set the value; for instance, 100rel. The value argument may have any value, as long as both the user-agent client (UAC) and user-agent server (UAS) configure it the same. The default value is supported with the 100rel value. • **require** *value* --Use provisional responses and you set the value; for instance, 100rel. The value argument may have any value, as long as both the UAC and UAS configure it the same. • **disable** --Disables the use of reliable provisional responses.
Step 6	**exit** **Example:** `Router(config-srv-sip)# exit`	Exits the current mode.

Configuring PRACK Timers and Retries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers prack** *number*
5. **retry prack** *number*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**timers prack** *number* **Example:** `Router(config-sip-ua)# timers prack 500`	Sets the amount of time that the user agent waits before retransmitting PRACK requests. The argument is as follows: • *number* --Time (in ms) to wait before retransmitting. Range: 100 to 1000. Default: 500.
Step 5	**retry prack** *number* **Example:** `Router(config-sip-ua)# retry prack 9`	Sets the number of times that the PRACK request is retransmitted to the other user agent. The argument is as follows: • *number* --Number of retries. Range: 1 to 10. Default: 10.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring COMET Timers and Retries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers comet** *number*
5. **retry comet** *number*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**timers comet** *number* **Example:** `Router(config-sip-ua)# timers comet 100`	Sets the amount of time that the user agent waits before retransmitting COMET requests. The argument is as follows: • *number* --Time (in ms) to wait before retransmitting. Range: 100 to 1000. Default: 500.
Step 5	**retry comet** *number* **Example:** `Router(config-sip-ua)# retry comet 10`	Sets the number of times that a COMET request is retransmitted to the other user agent. The argument is as follows: • *number* --Number of retries. Range: 1 to 10. Default: 10.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Configuring Reliable-Provisional-Response Timers and Retries

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **timers rel1xx** *number*
5. **retry rel1xx** *number*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**sip-ua** **Example:** `Router(config)# sip-ua`	Enters SIP user-agent mode.
Step 4	**timers rel1xx** *number* **Example:** `Router(config-sip-ua)# timers rel1xx 500`	Sets the amount of time that the user agent waits before retransmitting the reliable 1xx responses. The argument is as follows: • *number* --Time (in ms) to wait before retransmitting. Range: 100 to 1000. Default: 500.
Step 5	**retry rel1xx** *number* **Example:** `Router(config-sip-ua)# retries rel1xx 10`	Sets the number of times the reliable 1xx response is retransmitted to the other user agent. The argument is as follows: • *number* --Number of retries. Range: 1 to 10. Default: 6.
Step 6	**exit** **Example:** `Router(config-sip-ua)# exit`	Exits the current mode.

Reenabling SIP Hold Timer Support

To configure SIP hold timer support, perform the following steps.

Note The SIP: Hold Timer Support feature is enabled by default; no configuration tasks are required to enable this feature. This task enables the feature again if it was disabled by using the **no timers hold** command.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **timers hold** *time*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**timers hold** *time* **Example:** `Router(config)# timers hold 120`	Enables the SIP hold timer and sets the timer interval. The argument is as follows: • *time* --Time, in minutes, before the gateway disconnects held calls. Range: 15 to 2880. Default: 2880.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Configuring the SIP Media Inactivity Timer

The SIP Media Inactivity Timer feature requires configuration of the **ip rtcp report interval** command and the **timer receive-rtcp** command to enable detection of RTCP packets by the gateway. When these commands are configured, the gateway uses RTCP report detection, rather than Real-Time Protocol (RTP) packet detection, to determine whether calls on the gateway are still active or should be disconnected. This method is more reliable because there are periods during voice calls when one or both parties are not sending RTP packets.

One common example of a voice session in which no RTP is sent is when a caller dials into a conference call and mutes his endpoint. If voice activity detection (VAD, also known as silence suppression) is enabled, no RTP packets are sent while the endpoint is muted. However, the muted endpoint continues to send RTCP reports at the interval specified by the **ip rtcp report interval** command.

The **timer receive-rtcp** *value* argument (or Mfactor) is multiplied with the interval that is set using the **ip rtcp report interval** command. If no RTCP packets are received in the resulting time period, the call is disconnected. The gateway signals the disconnect to the SIP network and the TDM network so that upstream and downstream devices can clear their resources. The gateway sends a SIP BYE to disconnect the call and sends a Q.931 DISCONNECT back to the TDM network to clear the call upon the expiration of the timer. The Q.931 DISCONNECT is sent with a Cause code value of 3 (no route). There is no Q.931 Progress Indicator (PI) value included in the DISCONNECT.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **gateway**
4. **timer receive-rtcp** *timer*
5. **exit**
6. **ip rtcp report interval** *value*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> Enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**gateway** **Example:** `Router(config)# gateway`	Enables the H.323 VoIP gateway.
Step 4	**timer receive-rtcp** *timer* **Example:** `Router(config-gateway)# timer receive-rtcp 100`	Enables the Real-Time Control Protocol (RTCP) timer and to configure a multiplication factor for the RTCP timer interval for the SIP. The argument is as follows: • *timer* --Multiples of the RTCP report transmission interval. Range: 2 to 1000. Default: 5.
Step 5	**exit** **Example:** `Router(config-gateway)# exit`	Exits the current mode.
Step 6	**ip rtcp report interval** *value* **Example:** `Router(config)# ip rtcp report interval 500`	Sets the average reporting interval between subsequent RTCP report transmissions. The argument is as follows: • *value* --Average interval (in ms) for RTCP report transmissions. Range: 1 to 65535. Default: 5000.

	Command or Action	Purpose
		Note RFC 1889, *RTP: A Transport Protocol for Real-Time Applications* , recommends a minimum 5-second average reporting interval between successive RTCP reports. It also recommends that this interval be varied randomly. The randomization function is performed automatically and cannot be disabled. Therefore, the reporting interval does not remain constant throughout a given voice session, but its average is the specified reporting interval.
Step 7	**exit** **Example:** `Router(config)# exit`	Exits the current mode.

Verifying SIP QoS Features

To verify configuration of SIP QoS features, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show call fallback cache**
2. **show call fallback config**
3. **show call fallback stats**
4. **show call rsvp-sync conf**
5. **show call rsvp-sync stats**
6. **show dial-peer voice**
7. **show ip rsvp reservation**
8. **show running-conf**
9. **show sip-ua retry**
10. **show sip-ua statistics**
11. **show sip-ua status**
12. **show sip-ua timers**
13. **test call fallback probe** *ip-address codec*

DETAILED STEPS

Step 1 **show call fallback cache**

Use this command to display the current ICPIF estimates for all IP addresses in the cache.

Step 2 **show call fallback config**

Use this command to display the call fallback configuration.

Step 3 **show call fallback stats**

Use this command to display call fallback statistics.

Step 4 **show call rsvp-sync conf**

Use this command to display the configuration settings for RSVP synchronization.

Step 5 **show call rsvp-sync stats**

Use this command to display statistics for calls that attempted RSVP reservation.

Example:

```
Router# show call rsvp-sync
conf   Show RSVP/Voice Synchronization Config. information
state  Show RSVP/Voice Statistics
```

The following sample output also shows configuration settings for RSVP synchronization. Of particular note in the example are the following:

- Overture Synchronization is ON--Indicates that RSVP synchronization is enabled.
- Reservation Timer is set to 10 seconds--Number of seconds for which the RSVP reservation timer is configured.

Example:

```
Router# show call rsvp-sync conf
VoIP QoS:RSVP/Voice Signaling Synchronization config:
Overture Synchronization is ON
Reservation Timer is set to 10 seconds
```

The following sample output shows configuration settings for RSVP synchronization. Of particular note in the example are the following:

- Number of calls for which QoS was initiated--Number of calls for which RSVP setup was attempted.
- Number of calls for which QoS was torn down--Number of calls for which an established RSVP reservation was released.
- Number of calls for which Reservation Success was notified--Number of calls for which an RSVP reservation was successfully established.
- Total Number of PATH Errors encountered--Number of path errors that occurred.
- Total Number of RESV Errors encountered--Number of reservation errors that occurred.
- Total Number of Reservation Timeouts encountered--Number of calls in which the reservation setup was not complete before the reservation timer expires.

Example:

```
Router# show call rsvp-sync stats
VoIP QoS:Statistics Information:
Number of calls for which QoS was initiated   : 0
Number of calls for which QoS was torn down   : 0
Number of calls for which Reservation Success was notified : 0
Total Number of PATH Errors encountered : 0
Total Number of RESV Errors encountered : 0
Total Number of Reservation Timeouts encountered : 0
```

Step 6 **show dial-peer voice**

Use this command to display detailed information for a specific voice dial peer.

Example:

```
Router# show dial-peer voice 5
VoiceOverIpPeer5
        information type = voice,
        tag = 5, destination-pattern = `5550100',
        answer-address = `', preference=0,
        numbering Type = `unknown'
        group = 5, Admin state is up, Operation state is up,
        incoming called-number = `', connections/maximum = 0/unlimited,
        DTMF Relay = disabled,
        modem passthrough = system,
        huntstop = disabled,
        in bound application associated:session
        out bound application associated
        dnis-map =
        permission :both
        incoming COR list:maximum capability
        outgoing COR list:minimum requirement
        type = voip, session-target = `ipv4:172.18.192.218',
        technology prefix:
        settle-call = disabled
        ip media DSCP = default, ip signaling DSCP = default, UDP checksum = disabled,
        session-protocol = sipv2, session-transport = system, req-qos = best-effort,
        acc-qos = best-effort,
        fax rate = voice,   payload size =  20 bytes
        fax protocol = system
        fax NSF = 0xAD0051 (default)
        codec = g711ulaw,   payload size =  160 bytes,
        Expect factor = 0, Icpif = 20,
        Playout Mode is set to default,
        Initial 60 ms, Max 300 ms
        Playout-delay Minimum mode is set to default, value 40 ms
        Expect factor = 0,
Max Redirects = 1, Icpif = 20,signaling-type = cas,
        CLID Restrict = disabled
        VAD = enabled, Poor QOV Trap = disabled,
        voice class sip url = system,
        voice class sip rel1xx = system,
        voice class perm tag = `'
        Connect Time = 0, Charged Units = 0,
        Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
        Accepted Calls = 0, Refused Calls = 0,
        Last Disconnect Cause is "",
        Last Disconnect Text is "",
        Last Setup Time = 0.
```

Step 7 **show ip rsvp reservation**

Use this command to display RSVP-related receiver information currently in the database.

The following sample output shows, in the “To” field, the IP address of the receiver.

Example:

```
Router # show ip rsvp reservation
To              From            Pro DPort Sport Next Hop      I/F   Fi Serv BPS Bytes
172.18.193.101 172.18.193.102 UDP 20532 20600                     FF LOAD 24K 120
172.18.193.102 172.18.193.101 UDP 20600 20532 172.18.193.102 Et0/0 FF LOAD 24K 120
```

Step 8 **show running-conf**

Use this command to display the contents of the currently running configuration file, the configuration for a specific interface, or map class information. Use it to display SIP user-agent statistics, including reliable provisional response information. Use it also to display configuration for the Cisco IOS VoiceXML feature.

The following sample output shows SIP user-agent statistics, including reliable provisional response information. In the following partial output, a dynamic payload value of 115 is configured, freeing up the reserved value of 101.

Example:

```
Router# show running-config
Building configuration...
Current configuration: 2024 bytes
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
hostname r4
ip subnet-zero
ip tcp synwait-time 5
no ip domain-lookup
ipx routing 0000.0000.0004
no voice hpi capture buffer
no voice hpi capture destination
fax interface-type fax-mail
mta receive maximum-recipients 0
interface Loopback0
 ip address 10.0.0.0 255.255.255.0
interface FastEthernet0/0
 ip address 10.0.0.1 255.255.255.0
 speed 100
 full-duplex
interface Serial0/0
 ip address 10.0.0.4 255.255.255.0
 encapsulation frame-relay
.
.
.
call rsvp-sync
voice-port 3/0/0
voice-port 3/0/1
mgcp ip qos dscp cs5 media
mgcp ip qos dscp cs3 signaling
no mgcp timer receive-rtcp
mgcp profile default
dial-peer cor custom
dial-peer voice 1234 voip
 rtp payload-type nte 115
alias exec co config t
alias exec br show ip int brief
alias exec i show ip route
alias exec sr show run
alias exec sri sh run interface
alias exec sio show ip ospf
alias exec sioi show ip ospf int
alias exec sion show ip ospf nei
alias exec cir clear ip route *
alias exec ix show ipx route
alias exec b show ip bgp
alias exec sis show isdn status
alias exec fm show frame map
alias exec dm show dialer map
```

```
line con 0
 exec-timeout 0 0
 privilege level 15
 password password1
 logging synchronous
line aux 0
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 password password1
 logging synchronous
 no login
end
```

The following sample output shows configuration for the Cisco IOS VoiceXML feature. If the SIP hold timer is enabled, which is the default setting, and the timer is set to the default value of 2880 minutes, command output does not display the **timers hold 2880** command. In the following partial output, the hold timer is set to a nondefault value of 18 minutes.

Example:

```
Router# show running-config
Building configuration...
Current configuration :2791 bytes
.
.
.
sip-ua
max-forwards 10
retry invite 1
retry response 4
retry bye 1
retry cancel 1
timers expires 300000
timers hold 18
.
.
.
end!
```

Step 9 **show sip-ua retry**

Use this command to display SIP retry statistics.

Example:

```
Router# show sip-ua retry
SIP UA Retry Values
invite retry count = 10  response retry count = 1
bye retry count    = 1   cancel retry count   = 8
prack retry count  = 10  comet retry count    = 10
reliable 1xx count = 6
```

Step 10 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics.

Note When 0/0 is included in a field, the first number is an inbound count and the last number is an outbound count.

Example:

```
Router# show sip-ua statistics
```

```
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 15/9, Ringing 9/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 36/9
    Success:
      OkInvite 6/4, OkBye 5/5,
      OkCancel 5/5, OkOptions 0/0,
      OkPrack 29/8, OkPreconditionMet 11/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, SeeOther 0,
      UseProxy 0, AlternateService 0
    Client Error:
      BadRequest 0/0, 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,
      RequestCancel 0/0, NotAcceptableMedia 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 9/16, Ack 4/6, Bye 5/5,
    Cancel 5/9, Options 0/0,
    Prack 8/43, Comet 0/11
Retry Statistics
    Invite 5, Bye 0, Cancel 4, Response 0,
    Prack 13, Comet 0, Reliable1xx 0
```

Step 11 show sip-ua status

Use this command to display SIP user-agent status.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 1 (rfc 2052)
```

Step 12 show sip-ua timers

Use this command to display SIP user-agent timer settings.

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs unless noted)
trying 500, expires 150000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500, hold 2880 minutes
```

Step 13 **test call fallback probe** *ip-address codec*

Use this command to test a probe to a specific IP address and display ICPIF RTR values. Keywords and arguments are as follows:

- *ip-address* --Target IP address.
- *codec* --Codec type to test. Valid values are 711 (G.711 codec) and 729 (G.729 codec).

Troubleshooting Tips

Note For general troubleshooting tips and a list of important **debug** commands, see the "General Troubleshooting Tips" section.

- Make sure that you can make a voice call.
- Make sure VoIP is working before call fallback is configured.
- Use the **debug ccsip events** command, which includes output specific to the SIP Media Inactivity Timer feature.
- Use the **debug ccsip events** command to show all SIP SPI events tracing.
- Use the debug call fallback detail command to display details of the VoIP call fallback.
- Use the debug ccsip messages command to enable CCSIP SPI messages debugging trace.
- Use the debug ccsip error command to enable SIP error debugging trace.
- Use the debug ccsip all command to enable all SIP debugging traces.
- Use the debug rtr trace command to trace the execution of an SAA operation.
- Use the debug call fallback probe command to verify that probes are being sent correctly.
- Use the **debug ccsip all** command to enable all SIP debugging capabilities or use one of the following SIP debug commands:
  - **debug ccsip calls**
  - **debug ccsip error**
  - **debug ccsip events**
  - **debug ccsip messages**
  - **debug ccsip states**

- When terminating long distance or international calls over ISDN, the terminating switch receives information from the gateway. Generally, the information received consists of the numbering plan and the ISDN number type. As a default, the gateway tags both the numbering plan and number type as *Unknown* . However, this *Unknown* tag may cause interworking issues with some switches.

You can override the default ISDN numbering plan and number type with custom values, using the **isdn map** command. This command sets values on a per-number basis or on numbers that match set patterns. The following example shows an override of any plan or type with a called or calling number that begins with the numeral 1. Thus, the ISDN setup sent to the switch is used only for long distance numbers, the numbering plan is **ISDN,** and the type of number is **National**:

```
isdn map address 1.* plan isdn type* national
```

For more details on the **isdn map** command, see the *Cisco IOS Dial Technologies Command Reference*, Release 12.3.

Following is sample output for some of these commands:

Sample Output for the debug ccsip events Command

The following example trace shows a timer being set:

```
Router# debug ccsip events
00:04:29: sipSPICreateAndStartRtpTimer: Valid RTP/RTCP session found and CLI ena
bled to create and start the inactivity timer
00:04:29: sipSPICreateAndStartRtpTimer:Media Inactivity timer created for call.
Mfactor(from CLI): 5 RTCP bandwidth: 500
RTCP Interval(in ms): 5000
Normalized RTCP interval (in ms):25000
```

The following example trace shows a timer expiring:

```
Router# debug ccsip events
02:41:03: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
*Jan 1 02:41:34.107: sipSPIRtpDiscTimerExpired:RTP/RTCP receive timer expired. Disconnect
the call.
*Jan 1 02:41:34.107: Queued event From SIP SPI to CCAPI/DNS : SIPSPI_EV_CC_CALL_DISCONNECT
*Jan 1 02:41:34.107: CCSIP-SPI-CONTROL: act_active_disconnect
```

Note The **timer receive-rtcp**command configures a media activity timer that is common to both H.323 and SIP. If set, it affects both H.323 and SIP calls.

Sample Output for the debug rtr trace Command

```
Router# debug rtr trace
Router#
*Mar  1 00:11:42.439: RTR 1: Starting An Echo Operation - IP RTR Probe 1
*Mar  1 00:11:42.439: rtt hash insert : 10.1.1.63 32117
*Mar  1 00:11:42.439:    source=10.1.1.63(32117)  dest-ip=10.1.1.67(32057) vrf tableid = 0
*Mar  1 00:11:42.439: sending control enable:
*Mar  1 00:11:42.439: cmd: command: , ip: 10.1.1.67, port: 32057, duration: 1200
*Mar  1 00:11:42.439: sending control msg:
*Mar  1 00:11:42.439:  Ver: 1 ID: 20 Len: 52
*Mar  1 00:11:42.443: receiving reply
```

```
*Mar  1 00:11:42.443:  Ver: 1 ID: 20 Len: 8
*Mar  1 00:11:42.459: sdTime: -1989906017 dsTime: 2076306018
*Mar  1 00:11:42.459:  responseTime (1): 1
*Mar  1 00:11:42.479: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.479: jitterOut: 0
*Mar  1 00:11:42.479: jitterIn: -1
*Mar  1 00:11:42.479:  responseTime (2): 1
*Mar  1 00:11:42.499: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.499: jitterOut: 0
*Mar  1 00:11:42.499: jitterIn: 0
*Mar  1 00:11:42.499:  responseTime (3): 1
*Mar  1 00:11:42.519: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.519: jitterOut: 0
*Mar  1 00:11:42.519: jitterIn: 0
*Mar  1 00:11:42.519:  responseTime (4): 1
*Mar  1 00:11:42.539: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.539: jitterOut: 0
*Mar  1 00:11:42.539: jitterIn: 0
*Mar  1 00:11:42.539:  responseTime (5): 1
*Mar  1 00:11:42.559: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.559: jitterOut: 0
*Mar  1 00:11:42.559: jitterIn: 0
*Mar  1 00:11:42.559:  responseTime (6): 1
*Mar  1 00:11:42.579: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.579: jitterOut: 0
*Mar  1 00:11:42.579: jitterIn: 0
*Mar  1 00:11:42.579:  responseTime (7): 1
*Mar  1 00:11:42.599: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.599: jitterOut: 0
*Mar  1 00:11:42.599: jitterIn: 0
*Mar  1 00:11:42.599:  responseTime (8): 1
*Mar  1 00:11:42.619: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.619: jitterOut: 0
*Mar  1 00:11:42.619: jitterIn: 0
*Mar  1 00:11:42.619:  responseTime (9): 1
*Mar  1 00:11:42.639: sdTime: -1989906017 dsTime: 2076306017
*Mar  1 00:11:42.639: jitterOut: 0
*Mar  1 00:11:42.639: jitterIn: 0
*Mar  1 00:11:42.639:  responseTime (10): 1
*Mar  1 00:11:42.639: rtt hash remove: 10.1.1.63 32117
Router# debug rtr trace
Router#
*Mar  1 00:14:12.439: RTR 1: Starting An Echo Operation - IP RTR Probe 1
*Mar  1 00:14:12.439: rtt hash insert : 10.1.1.63 32117
*Mar  1 00:14:12.439:    source=10.1.1.63(32117)  dest-ip=10.1.1.67(32057) vrf tableid = 0
*Mar  1 00:14:12.439: sending control enable:
*Mar  1 00:14:12.439: cmd: command: , ip: 10.1.1.67, port: 32057, duration: 1200
*Mar  1 00:14:12.439: sending control msg:
*Mar  1 00:14:12.439:  Ver: 1 ID: 27 Len: 52
*Mar  1 00:14:13.439: control message timeout
*Mar  1 00:14:13.439: sending control msg:
*Mar  1 00:14:13.439:  Ver: 1 ID: 28 Len: 52
*Mar  1 00:14:14.439: control message timeout
*Mar  1 00:14:14.439:    control message failure: 1
*Mar  1 00:14:14.439: rtt hash remove: 10.1.1.63 32117
*Mar  1 00:14:42.439: RTR 1: Starting An Echo Operation - IP RTR Probe 1
*Mar  1 00:14:42.439: rtt hash insert : 10.1.1.63 32117
*Mar  1 00:14:42.439:    source=10.1.1.63(32117)  dest-ip=10.1.1.67(32057) vrf tableid = 0
```

Sample Output for the debug call fallback probe Command

```
Router# debug call fallback probe
Router#
```

```
*Mar  1 00:10:12.439: fb_main: Probe timer expired, 10.1.1.67, codec:g711ulaw
*Mar  1 00:10:12.639: fb_main:NumOfRTT=10, RTTSum=10, loss=0, jitter in=0, jitter out=0->
10.1.1.67, codec:g711ulaw, delay = 28
*Mar  1 00:10:12.639:  g113_calc_icpif: loss=0, expect_factor=10, delay (w/codec delay)=28,
 Icpif=0
*Mar  1 00:10:12.639: fb_main: New smoothed values: inst_weight=100, ICPIF=0, Delay=28,
Loss=0 -> 10.1.1.67, codec:g711ulaw
3640SDP#
r  1 00:13:12.439: fb_main: Probe timer expired, 10.1.1.67, codec:g711ulaw
*Mar  1 00:13:14.439: %FALLBACK-3-PROBE_FAILURE: A probe error to 10.1.1.67 occured - control
 message failure
*Mar  1 00:13:14.439: fb_main:NumOfRTT=0, RTTSum=0, loss=100, jitter in=0, jitter out=0->
10.1.1.67, codec:g711ulaw, delay is N/A (since loss is 100 percent)
*Mar  1 00:13:14.439:  g113_calc_icpif: loss=100, expect_factor=10, delay is N/A (since
loss is 100 percent), Icpif=64
*Mar  1 00:13:14.439: fb_main: New unsmoothed values: inst_weight=100, ICPIF=64, Delay=N/A,
 Loss=100 -> 10.1.1.67, codec:g711ulaw
3/0:23(1) is in busyout state
*Mar  1 00:13:22.435: %LINK-3-UPDOWN: Interface ISDN-VOICE 3/0:23(1), changed state to
Administrative Shutdown
*Mar  1 00:13:22.439: %ISDN-6-LAYER2DOWN: Layer 2 for Interface Se3/0:23, TEI 0 changed to
 down
```

Configuration Examples for SIP QoS Features

SIP Gateway Support of RSVP and TEL URL Example

This configuration example shows RSVP for SIP calls on gateways being enabled. Gateway A is the originating gateway and Gateway B is the terminating gateway:

```
GATEWAY A
---------
Router# show running-config
.
.
.
interface Ethernet0/0
 ip address 172.18.193.101 255.255.255.0
 fair-queue 64 256 235
 ip rsvp bandwidth 7500 7500
!
voice-port 1/0/0
!
dial-peer voice 1 pots
 destination-pattern 111
 port 1/0/0
!
dial-peer voice 2 voip
 incoming called-number 111
 destination-pattern 222
 session protocol sipv2
 session target ipv4:172.18.193.102
 req-qos controlled-load
!
GATEWAY B
---------
!
interface Ethernet0/0
 ip address 172.18.193.102 255.255.255.0
```

```
 fair-queue 64 256 235
 ip rsvp bandwidth 7500 7500
!
voice-port 1/0/1
!
dial-peer voice 1 pots
 destination-pattern 222
 port 1/0/1
!
dial-peer voice 2 voip
 incoming called-number 222
 destination-pattern 111
 session protocol sipv2
 session target ipv4:172.18.193.101
 req-qos controlled-load
!
```

SIP Media Inactivity Timer Example

```
Router# show running-config
!
version 12.3
no parser cache
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
hostname madison
boot system flash
no logging buffered
aaa new-model
!
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection passsword stop-only group radius
aaa accounting connection h323 start-stop group radius
aaa session-id common
!
resource-pool disable
clock timezone EST -5
!
ip subnet-zero
ip tcp path-mtu-discovery
ip name-server 172.18.192.48
ip dhcp smart-relay
!
isdn switch-type primary-ni
!
voice service voip
h323
!
voice class codec 1
 codec preference 1 g723ar53
 codec preference 2 g723r53
 codec preference 3 g729br8
 codec preference 4 gsmfr
 codec preference 5 g726r24
 codec preference 6 g726r32
voice class codec 2
 codec preference 1 g729br8
 codec preference 2 g729r8
 codec preference 3 g723ar53
```

```
 codec preference 4 g723ar63
 codec preference 5 g723r53
 codec preference 6 g723r63
 codec preference 7 gsmfr
 codec preference 8 gsmefr
!
voice class codec 3
 codec preference 1 g726r24
 codec preference 2 gsmefr
 codec preference 3 g726r16
!
fax interface-type modem
 mta receive maximum-recipients 0
controller T1 0
 framing esf
 clock source line secondary 1
 linecode ami
 pri-group timeslots 1-24
 description summa_pbx
!
controller T1 1
 framing esf
 linecode ami
 pri-group timeslots 1-24
 description summa_pbx
!
controller T1 2
 framing sf
 linecode ami
!
controller T1 3
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb dtmf dnis
 cas-custom 0
!
gw-accounting h323 vsa
gw-accounting voip
interface Ethernet0
 ip address 172.18.193.99 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 ip rsvp bandwidth 7500 7500
!
interface Serial0:23
 no ip address
 isdn switch-type primary-ni
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn T203 10000
 isdn T306 30000
 isdn T310 4000
 isdn disconnect-cause 1
 fair-queue 64 256 0
 no cdp enable
interface Serial1:23
 no ip address
 isdn switch-type primary-ni
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn T203 10000
 isdn disconnect-cause 1
 fair-queue 64 256 0
```

```
 no cdp enable
!
interface FastEthernet0
 ip address 10.1.1.1 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
 ip rsvp bandwidth 7 7
!
ip classless
ip route 10.0.0.0 255.0.0.0 172.18.193.1
ip route 172.18.0.0 255.255.0.0 172.18.193.1
no ip http server
ip pim bidir-enable
!
ip radius source-interface Ethernet0
!
map-class dialer test
dialer voice-call
dialer-list 1 protocol ip permit
!
radius-server host 172.18.192.108 auth-port 1645 acct-port 1646
radius-server retransmit 1
radius-server key lab
radius-server vsa send accounting
radius-server vsa send authentication
call rsvp-sync
call application voice voice_billing tftp://172.18.207.16/app_passport_silent.2.0.0.0.tcl
!
voice-port 0:D
voice-port 1:D
voice-port 3:0
!
no mgcp timer receive-rtcp
!
mgcp profile default
!
dial-peer voice 10 pots
 destination-pattern 2021010119
 port 3:0
 prefix 2021010119
!
dial-peer voice 11 pots
 incoming called-number 3111100
 destination-pattern 3100802
 progress_ind progress enable 8
 port 0:D
 prefix 93100802
!
dial-peer voice 36 voip
 application session
 incoming called-number 3100802
 destination-pattern 3100801
 session protocol sipv2
 session target ipv4:172.18.193.100
 codec g726r16
!
dial-peer voice 5 voip
 destination-pattern 5550155
 session protocol sipv2
 session target ipv4:172.18.192.218
!
dial-peer voice 12 pots
```

```
 destination-pattern 3111100
 prefix 93111100
!
dial-peer voice 19 pots
 destination-pattern 2017030200
 port 1:D
 prefix 2017030200
!
dial-peer voice 30 voip
 destination-pattern 36602
 voice-class codec 2
 session protocol sipv2
 session target ipv4:172.18.193.120
 dial-peer voice 47 pots
 destination-pattern 2021030100
 port 3:0
!
dial-peer voice 3111200 pots
 destination-pattern 311200
 prefix 93100802
!
dial-peer voice 31 voip
 destination-pattern 36601
 session protocol sipv2
 session target ipv4:172.18.193.98
!
dial-peer voice 1234 voip
 incoming called-number 1234
 destination-pattern 1234
 session target loopback:rtp
!
gateway
 timer receive-rtcp 5
!
sip-ua
 aaa username proxy-auth
 retry invite 1
 retry bye 1
!
line con 0
 exec-timeout 0 0
 line aux 0
 line vty 0 4
 password password1
!
end
```

Use the **debug ccsip all** command to troubleshoot the SIP: Hold Timer Support feature. To minimize the possibility of performance impact, use this command during periods of minimal traffic. Make sure VoIP is working before hold timer support is configured.

```
Router# debug ccsip all
Feb 28 21:34:09.479:Received:
INVITE sip:36601@172.18.193.98:5060 SIP/2.0
Via:SIP/2.0/UDP
172.18.193.187:5060;branch=f104ef32-21751ddb-ce8428fe-cffdbf5-1
Record-Route:
sip:5550155.f104ef32-21751ddb-ce8428fe-cffdbf5@172.18.197.182:5060;maddr=172.18.193.187>
Via:SIP/2.0/UDP 172.18.197.182:5060;received=172.18.197.182
From:"5550155"
sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
To:<sip:36601@172.18.193.187>;tag=8CDE00-1506
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
```

```
CSeq:102 INVITE
User-Agent:CSCO/4
Contact:<sip:5550155@172.18.197.182:5060>
Content-Type:application/sdp
Content-Length:243
v=0
o=Cisco-SIPUA 2802 21073 IN IP4 172.18.197.182
s=SIP Call
c=IN IP4 0.0.0.0
t=0 0
m=audio 28478 RTP/AVP 0 8 18 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
*Feb 28 21:34:09.479:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.187:37775
*Feb 28 21:34:09.479:*****CCB found in UAS Request table. ccb=0x63C031B0
*Feb 28 21:34:09.479:CCSIP-SPI-CONTROL: act_active_new_message
*Feb 28 21:34:09.479:CCSIP-SPI-CONTROL: sact_active_new_message_request
*Feb 28 21:34:09.479:CCSIP-SPI-CONTROL: Converting TimeZone EST to SIP
default timezone = GMT
*Feb 28 21:34:09.479:sip_stats_method
*Feb 28 21:34:09.479:sact_active_new_message_request:Case of Mid-Call INVITE
*Feb 28 21:34:09.479:CCSIP-SPI-CONTROL: sipSPIHandleMidCallInvite
*Feb 28 21:34:09.479:CCSIP-SPI-CONTROL: sipSPIUASSessionTimer
*Feb 28 21:34:09.479:sipSPIDoMediaNegotiation:number of m lines is 1
*Feb 28 21:34:09.479: Codec (No Codec   ) is not in preferred list
*Feb 28 21:34:09.479:sipSPIDoAudioNegotiation:An exact codec match not
configured, using interoperable codec g729r8
*Feb 28 21:34:09.479:sipSPIDoAudioNegotiation:Codec (g729r8) Negotiation
Successful on Static Payload for m-line 1
*Feb 28 21:34:09.479:sipSPIDoPtimeNegotiation:No ptime present or
multiple ptime attributes that can't be handled
*Feb 28 21:34:09.479:sipSPIDoDTMFRelayNegotiation:m-line index 1
*Feb 28 21:34:09.479:sipSPIDoDTMFRelayNegotiation:Requested DTMF-RELAY
option(s) not found in Preferred DTMF-RELAY option list!
*Feb 28 21:34:09.479: sipSPIStreamTypeAndDtmfRelay:DTMF Relay mode :Inband Voice
*Feb 28 21:34:09.479:sip_sdp_get_modem_relay_cap_params:
*Feb 28 21:34:09.479:sip_sdp_get_modem_relay_cap_params:NSE payload from
X-cap = 0
*Feb 28 21:34:09.479:sip_select_modem_relay_params:X-tmr not present in SDP.
Disable modem relay
*Feb 28 21:34:09.479:sipSPIGetSDPDirectionAttribute:No direction attribute
present or multiple direction attributes that can't be handled
*Feb 28 21:34:09.479:sipSPIDoAudioNegotiation:Codec negotiation
successful for media line 1 payload_type=18, codec_bytes=20, codec=g729r8,
dtmf_relay=inband-voice stream_type=voice-only (0), dest_ip_address=0.0.0.0,
dest_port=28478
*Feb 28 21:34:09.479:sipSPICompareSDP
*Feb 28 21:34:09.483:sipSPICompareStreams:stream 1 dest_port:old=28478
new=28478
*Feb 28 21:34:09.483:sipSPICompareConnectionAddress
*Feb 28 21:34:09.483:sipSPICompareConnectionAddress:Call hold activated for
stream 1
*Feb 28 21:34:09.483:sipSPICompareStreams:Flags set for stream 1:
RTP_CHANGE=No
CAPS_CHANGE=No
*Feb 28 21:34:09.483:sipSPICompareSDP:Flags set for call:NEW_MEDIA=No
DSPDNLD_REQD=No
*Feb 28 21:34:09.483:sipSPIGetGtdBody:No valid GTD body found.
*Feb 28 21:34:09.483:sipSPIReplaceSDP
*Feb 28 21:34:09.483:sipSPICopySdpInfo
```

```
*Feb 28 21:34:09.483:sipSPISetHoldTimer:Starting hold timer at 15 minutes
!!Timer started
*Feb 28 21:34:09.483:sipSPIUpdCallWithSdpInfo:
Preferred Codec        :g729r8, bytes :20
Preferred  DTMF relay  :inband-voice
Preferred NTE payload  :101
Early Media            :Yes
Delayed Media          :No
Bridge Done            :Yes
New Media              :No
DSP DNLD Reqd          :No
*Feb 28 21:34:09.483:sipSPISetMediaSrcAddr: media src addr for stream 1 =
172.18.193.98
*Feb 28 21:34:09.483:sipSPIUpdCallWithSdpInfo:Stream Type:0
M-line Index             :1
State                  :STREAM_ACTIVE (5)
Callid                 :1
Negotiated Codec       :g729r8, bytes :20
Negotiated DTMF relay  :inband-voice
Negotiated NTE payload :0
Media Srce Addr/Port   :172.18.193.98:18764
Media Dest Addr/Port   :0.0.0.0:28478
*Feb 28 21:34:09.483:sipSPIProcessMediaChanges
*Feb 28 21:34:09.483:CCSIP-SPI-CONTROL: sipSPIIncomingCallSDP
*Feb 28 21:34:09.483: SDP already there use old sdp and updatemedia if needed
*Feb 28 21:34:09.483:sipSPIUpdateSrcSdpVariablePart
*Feb 28 21:34:09.483:sipSPIUpdateSrcSdpVariablePart:setting stream 1
portnum to 18764
*Feb 28 21:34:09.483:CCSIP-SPI-CONTROL: sipSPISendInviteResponse
*Feb 28 21:34:09.483:sipSPIAddLocalContact
*Feb 28 21:34:09.483:sip_generate_sdp_xcaps_list:Modem Relay and T38
disabled.
X-cap not needed
*Feb 28 21:34:09.483: Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE
*Feb 28 21:34:09.483:sip_stats_status_code
*Feb 28 21:34:09.483:Sent:
SIP/2.0 200 OK
Via:SIP/2.0/UDP
172.18.193.187:5060;branch=f104ef32-21751ddb-ce8428fe-cffdbf5-1,SIP/2.0/UDP
172.18.197.182:5060;received=172.18.197.182
From:"5550155"
sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
To:<sip:36601@172.18.193.187>;tag=8CDE00-1506
Date:Mon, 01 Mar 1993 02:34:09 GMT
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
Server:Cisco-SIPGateway/IOS-12.x
CSeq:102 INVITE
Allow:INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE,
NOTIFY,
INFO
Allow-Events:telephone-event
Contact:<sip:36601@172.18.193.98:5060>
Record-Route:
sip:5550155.f104ef32-21751ddb-ce8428fe-cffdbf5@172.18.197.182:5060;maddr=172.18.193.18
Content-Type:application/sdp
Content-Length:229
v=0
o=CiscoSystemsSIP-GW-UserAgent 6264 8268 IN IP4 172.18.193.98
s=SIP Call
c=IN IP4 172.18.193.98
t=0 0
m=audio 18764 RTP/AVP 18 19
c=IN IP4 172.18.193.98
a=rtpmap:18 G729/8000
```

```
a=rtpmap:19 CN/8000
a=fmtp:18 annexb=no
*Feb 28 21:34:09.635:Received:
ACK sip:36601@172.18.193.98:5060 SIP/2.0
Via:SIP/2.0/UDP 172.18.193.187:5060;branch=f104ef32-21751ddb-ce8428fe-cffdbf5
Record-Route:
<sip:36601.f104ef32-21751ddb-ce8428fe-cffdbf5@172.18.193.187:5060;maddr=172.18.193.187
Via:SIP/2.0/UDP 172.18.197.182:5060
From:"5550155"
sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
To:<sip:36601@172.18.193.187>;tag=8CDE00-1506
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
CSeq:102 ACK
User-Agent:CSCO/4
Content-Length:0
*Feb 28 21:34:09.635:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.187:37779
*Feb 28 21:34:09.635:*****CCB found in UAS Request table. ccb=0x63C031B0
*Feb 28 21:34:09.635:CCSIP-SPI-CONTROL: act_active_new_message
*Feb 28 21:34:09.635:CCSIP-SPI-CONTROL: sact_active_new_message_request
*Feb 28 21:34:09.635:CCSIP-SPI-CONTROL: Converting TimeZone EST to SIP
default timezone = GMT
*Feb 28 21:34:09.635:sip_stats_method
Router#
*Feb 28 21:49:09.483:act_onhold_timeout:Hold Timer Expired, tearing down
call
!!Timer expires after 15 minutes and gateway sends out BYE to the other endpoint.
*Feb 28 21:49:09.483:ccsip_set_release_source_for_peer:ownCallId[1], src[6]
*Feb 28 21:49:09.483: Queued event from SIP SPI :SIPSPI_EV_CREATE_CONNECTION
*Feb 28 21:49:09.483:0x63C031B0 :State change from (STATE_ACTIVE,
SUBSTATE_NONE)  to (STATE_ACTIVE, SUBSTATE_CONNECTING)
*Feb 28 21:49:09.483:0x63C031B0 :State change from (STATE_ACTIVE,
SUBSTATE_CONNECTING)  to (STATE_ACTIVE, SUBSTATE_CONNECTING)
*Feb 28 21:49:09.483: Queued event from SIP SPI :
SIPSPI_EV_CC_CALL_DISCONNECT
*Feb 28 21:49:09.483:CCSIP-SPI-CONTROL: sipSPICheckSocketConnection:
Connid(1)
created to 172.18.193.187:5060, local_port 51433
*Feb 28 21:49:09.483:0x63C031B0 :State change from (STATE_ACTIVE,
SUBSTATE_CONNECTING)  to (STATE_ACTIVE, SUBSTATE_NONE)
*Feb 28 21:49:09.483:sipSPIStopHoldTimer:Stopping hold timer
*Feb 28 21:49:09.483:CCSIP-SPI-CONTROL: sipSPIAddRouteHeaders status = TRUE
Route <sip:5550155@172.18.197.182:5060>
*Feb 28 21:49:09.483: Queued event from SIP SPI :SIPSPI_EV_SEND_MESSAGE
*Feb 28 21:49:09.483:sip_stats_method
*Feb 28 21:49:09.483:0x63C031B0 :State change from (STATE_ACTIVE,
SUBSTATE_NONE)  to (STATE_DISCONNECTING, SUBSTATE_NONE)
*Feb 28 21:49:09.483:CCSIP-SPI-CONTROL: act_disconnecting_disconnect
*Feb 28 21:49:09.483:Sent:
BYE
sip:5550155.d3c5ae1f-b5cf873d-17053c1f-e0126b18@172.18.197.182:5060;maddr=172.18.193.187
SIP/2.0
Via:SIP/2.0/UDP  172.18.193.98:5060
From:<sip:36601@172.18.193.187>;tag=8CDE00-1506
To:"5550155"
<sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
Date:Mon, 01 Mar 1993 02:34:09 GMT
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
User-Agent:Cisco-SIPGateway/IOS-12.x
Max-Forwards:10
Route:<sip:5550155@172.18.197.182:5060>
Timestamp:730954149
CSeq:101 BYE
Content-Length:0
```

```
*Feb 28 21:49:09.487:Received:
SIP/2.0 100 Trying
Via:SIP/2.0/UDP 172.18.193.98:5060;received=172.18.193.98
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
From:<sip:36601@172.18.193.187>;tag=8CDE00-1506
To:"5550155"
<sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
CSeq:101 BYE
Content-Length:0
*Feb 28 21:49:09.487:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.187:37781
*Feb 28 21:49:09.487:*****CCB found in UAS Response table. ccb=0x63C031B0
*Feb 28 21:49:09.487:CCSIP-SPI-CONTROL: act_disconnecting_new_message
*Feb 28 21:49:09.487:
CCSIP-SPI-CONTROL: sact_disconnecting_new_message_response
*Feb 28 21:49:09.487:CCSIP-SPI-CONTROL: sipSPICheckResponse
*Feb 28 21:49:09.487:sip_stats_status_code
*Feb 28 21:49:09.487: Roundtrip delay 4 milliseconds for method BYE
*Feb 28 21:49:09.539:Received:
SIP/2.0 200 OK
Via:SIP/2.0/UDP 172.18.193.98:5060;received=172.18.193.98
From:<sip:36601@172.18.193.187>;tag=8CDE00-1506
To:"5550155"
<sip:5550155@172.18.193.187>;tag=003094c2e56a00aa13cdcefd-61096c17
Call-ID:003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.182
CSeq:101 BYE
Server:CSCO/4
Content-Length:0
*Feb 28 21:49:09.539:HandleUdpSocketReads :Msg enqueued for SPI with IPaddr:
172.18.193.187:37784
*Feb 28 21:49:09.539:*****CCB found in UAS Response table. ccb=0x63C031B0
*Feb 28 21:49:09.539:CCSIP-SPI-CONTROL: act_disconnecting_new_message
*Feb 28 21:49:09.539:
CCSIP-SPI-CONTROL: sact_disconnecting_new_message_response
*Feb 28 21:49:09.539:CCSIP-SPI-CONTROL: sipSPICheckResponse
*Feb 28 21:49:09.539:sip_stats_status_code
*Feb 28 21:49:09.539: Roundtrip delay 56 milliseconds for method BYE
*Feb 28 21:49:09.539:CCSIP-SPI-CONTROL: sipSPICallCleanup
*Feb 28 21:49:09.539:sipSPIIcpifUpdate :CallState:3 Playout:16840
DiscTime:1014954 ConnTime 924101
*Feb 28 21:49:09.539:0x63C031B0 :State change from (STATE_DISCONNECTING,
SUBSTATE_NONE)  to (STATE_DEAD, SUBSTATE_NONE)
*Feb 28 21:49:09.539:The Call Setup Information is :
Call Control Block (CCB) :0x63C031B0
State of The Call        :STATE_DEAD
TCP Sockets Used         :NO
Calling Number           :5550155
Called Number            :36601
Number of Media Streams  :1
*Feb 28 21:49:09.539:Media Stream 1
Negotiated Codec         :g729r8
Negotiated Codec Bytes   :20
Negotiated Dtmf-relay    :0
Dtmf-relay Payload       :0
Source IP Address (Media):172.18.193.98
Source IP Port    (Media):18764
Destn  IP Address (Media):0.0.0.0
Destn  IP Port    (Media):28478
*Feb 28 21:49:09.539:Orig Destn IP Address:Port (Media):0.0.0.0:0
*Feb 28 21:49:09.539:
Source IP Address (Sig  ):172.18.193.98
Destn SIP Req Addr:Port  :172.18.193.187:5060
Destn SIP Resp Addr:Port :172.18.193.187:5060
Destination Name         :172.18.193.187
```

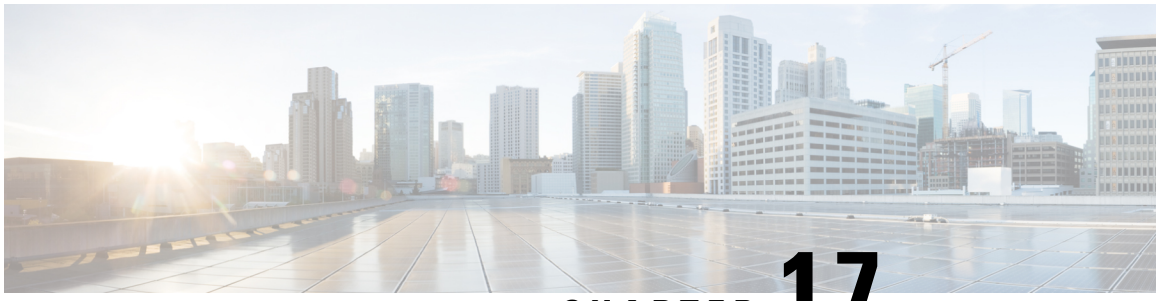
```
*Feb 28 21:49:09.539:
Disconnect Cause (CC)    :102
Disconnect Cause (SIP)   :200
*Feb 28 21:49:09.539:****Deleting from UAS Request table. ccb=0x63C031B0
key=003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.18236601
*Feb 28 21:49:09.539:****Deleting from UAS Response table. ccb=0x63C031B0
key=003094c2-e56a02b9-670be98d-1cf394e0@172.18.197.1828CDE00-1506
*Feb 28 21:49:09.539:Removing call id 1
*Feb 28 21:49:09.543:RequestCloseConnection:Closing connid 1 Local Port 51433
*Feb 28 21:49:09.543: Queued event from SIP SPI :SIPSPI_EV_CLOSE_CONNECTION
*Feb 28 21:49:09.543:sipSPIFlushEventBufferQueue:There are 0 events on the
internal queue that are going to be free'd
*Feb 28 21:49:09.543: freeing ccb 63C031B0
*Feb 28 21:49:09.543:udpsock_close_connect:Socket fd:1 closed for connid 1 with remote
port:5060
```

Additional References

General SIP References

References Mentioned in This Chapter

- *Cisco IOS Quality of Service Solutions Command Reference* , Release 12.3 at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/qos_r/index.htm



CHAPTER 17

Configuring SIP Support for SRTP

This module contains information about configuring Session Initiation Protocol (SIP) support for the Secure Real-time Transport Protocol (SRTP). SRTP is an extension of the Real-time Transport Protocol (RTP) Audio/Video Profile (AVP) and ensures the integrity of RTP and Real-Time Control Protocol (RTCP) packets that provide authentication, encryption, and the integrity of media packets between SIP endpoints.

SIP support for SRTP was introduced in Cisco IOS Release 12.4(15)T.

You can configure the handling of secure RTP calls on both a global level and on an individual dial peer basis on Cisco IOS voice gateways. You can also configure the gateway (or dial peer) either to fall back to (nonsecure) RTP or to reject (fail) the call for cases where an endpoint does not support SRTP.

The option to allow negotiation between SRTP and RTP endpoints was added for Cisco IOS Release 12.4(20)T and later releases, as was interoperability of SIP support for SRTP on Cisco IOS voice gateways with Cisco Unified Communications Manager. In Cisco IOS Release 12.4(22)T and later releases, you can configure SIP support for SRTP on Cisco Unified Border Elements (Cisco UBEs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring SIP Support for SRTP

- Establish a working IP network and configure VoIP.

Note For information about configuring VoIP, see "Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms".

- Ensure that the gateway has voice functionality configured for SIP.
- Ensure that your Cisco router has adequate memory.
- As necessary, configure the router to use Greenwich Mean Time (GMT). SIP requires that all times be sent in GMT. SIP INVITE messages are sent in GMT. However, the default for routers is to use Coordinated Universal Time (UTC). To configure the router to use GMT, issue the **clock timezone** command in global configuration mode and specify GMT.

Restrictions for Configuring SIP Support for SRTP

- The SIP gateway does not support codecs other than those listed in the table titled “SIP Codec Support by Platform and Cisco IOS Release” in the “Enhanced Codec Support for SIP Using Dynamic Payloads” section of the "Configuring SIP QoS Features" module.
- SIP requires that all times be sent in GMT.

Information About Configuring SIP Support for SRTP

The SIP Support for SRTP features use encryption to secure the media flow between two SIP endpoints. Cisco IOS voice gateways and Cisco Unified Border ElementsCisco Unified Border Elements use the Digest method for user authentication and, typically, they use Transport Layer Security (TLS) for signaling authentication and encryption.

Note To provide more flexibility, TLS signaling encryption is no longer required for SIP support of SRTP in Cisco IOS Release 12.4(22)T and later releases. Secure SIP (SIPS) is still used to establish and determine TLS but TLS is no longer a requirement for SRTP, which means calls established with SIP only (and not SIPS) can still successfully negotiate SRTP without TLS signaling encryption. This also means you could configure encryption using a different protocol, such as IPsec. However, Cisco does not recommend configuring SIP support for SRTP without TLS signaling encryption because doing so compromises the intent of forcing media encryption (SRTP).

When TLS is used, the cryptographic parameters required to successfully negotiate SRTP rely on the cryptographic attribute in the Session Description Protocol (SDP). To ensure the integrity of cryptographic parameters across a network, SRTP uses the SIPS schema (sips:*example* .com). If the Cisco IOS voice gateway or Cisco Unified Border Element is configured to use TLS encryption and sends an invite to an endpoint that

cannot provide TLS support, that endpoint rejects the INVITE message. For cases like these, you can configure the gateway or Cisco Unified Border Elements either to fall back to an RTP-only call or to reject the call.

The SIP support for SRTP features provide the following security benefits:

- Confidentiality of RTP packets--protects packet-payloads from being read by unapproved entities but does so without authorized entities having to enter a secret encryption key.
- Message authentication of RTP packets--protects the integrity of the packet against forgery, alteration, or replacement.
- Replay protection--protects the session address against denial of service attacks.

The table below describes the security level of SIP INVITE messages according to which of the four possible combinations of TLS and SRTP is configured.

Table 59: TLS-SRTP Combinations

TLS	SRTP	Description
On	On	Signaling and media are secure.
Off	On	Signaling is insecure: • If you use the **srtp fallback** command, the gateway sends an RTP-only SDP. • If you do not configure the **srtp fallback** command, the call fails and the gateway does not send an INVITE message. **Note** In Cisco IOS Release 12.4(20)T and later releases (and, for Cisco UBEs, in Cisco IOS Release 12.4(22)T and later releases), calls established with SRTP only (and not SIPS) will succeed even if the **srtp fallback** command is not configured.
On	Off	RTP-only call.
Off	Off	Signaling and media are not secure.

Cryptographic Parameters

RFC 3711 defines the SRTP cryptographic parameters, including valid syntax and values for attribute a=crypto (see the table below). Some of these parameters are declarative and apply only to the send direction of the declarer, while others are negotiable and apply to both send and receive directions.

The following shows the cryptographic attribute syntax:

a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]

The table below summarizes the syntax for the cryptographic attribute.

Table 60: Cryptographic Attribute Syntax

Attribute	Optional	Description
tag	No	The tag attribute is a unique decimal number used as an identifier for a particular cryptographic attribute to determine which of the several offered cryptographic attributes was chosen by the answerer.
crypto-suite	No	The crypto-suite attribute defines the encryption and authentication algorithm. Cisco IOS voice gateways and Cisco UBEs support default suite AES_CM_128_HMAC_SHA1_32 (AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag).
key-params	No	“inline:” <key| |salt> [“|” lifetime] [“|” MKI “:” length] key| | salt is base64 encoded contacted master key and salt.
session-params	Yes	The session-params attribute is specific to a given transport and is optional. The gateway does not generate any session-params in an outgoing INVITE message, nor will the SDP library parse them.

Call Control and Signaling

SIP uses the SRTP library to receive cryptographic keys. If you configure SRTP for the call and cryptographic context is supported, SDP offers the cryptographic parameters. If the cryptographic parameters are negotiated successfully, the parameters are downloaded to the DP, which encrypts and decrypts the packets. The sender encrypts the payload by using the AES algorithm and builds an authentication tag, which is encapsulated to the RTP packet. The receiver verifies the authentication tag and then decrypts the payload.

Default and Recommended SRTP Settings

The table below lists the default and recommended SRTP settings.

Table 61: Default and Recommended SRTP Settings

Parameter	Default	Recommended Value
Key derivation rate	0	0--Rekeying is supported
Master key length	128 bits	128 bits
Master salt key length	112 bits	112 bits
MKI indicator	0	0
MKI length	0	0
PRF	AES_CM	128
Session authentication key length	128	128
Session encryption key length	128 bits	128 bits
Session salt key length	112	112

Parameter	Default	Recommended Value
SRTP authentication	HMAC-SHA1	HMAC-SHA1
SRTCP authentication	HMAC-SHA1	HMAC-SHA1
SRTP cipher	AES_CM	AES_CM
SRTCP cipher	AES_CM	NULL
SRTP HMAC tag length	80	32 (voice)--Supported 80 (other)--Not supported
SRTCP HMAC tag length	80	80
SRTP packets maximum lifetime	2^48 packets	2^48 packets
SRTCP packets maximum lifetime	2^31 packets	2^31 packets
SRTP replay-window size	64	64--Not supported
SRTCP replay-window size	64	64--Not supported

Before an SRTP session can be established on a Cisco IOS voice gateway or Cisco UBE, the following cryptographic information must be exchanged in SDP between the two endpoints:

- Crypto suite--crypto algorithm {AES_CM_128_HMAC_SHA1_32} and the supported codec list {g711, G729, G729a}. There could be one or more crypto suites. Cisco IOS Release 12.4(15)T supports only one crypto suite.
- Crypto context--16-byte master key and a 14-byte master salt.

Generating Master Keys

The SRTP library provides an application program interface (API), srtp_generate_master_key, to generate a random master key. For encryption and authentication purposes, the key length is 128 bits (master key and session keys). Additionally, RFC 3711 introduces “salting keys”--master salts and sessions salts--and strongly recommends the use of a master salt in the key derivation of session keys. The salting keys (salts) are used to fight against pre-computation and time-memory tradeoff attacks.

The master salt (also known as the n-bit SRTP key) prevents off-line key-collision attacks on the key derivation and, when used, must be random (but can be public). The master salt is derived from the master key and is used in the key derivation of session keys. Session salts, in turn, are used in encryption to counter various attacks against additive stream ciphers. All salting keys (master salt and session salts) are 112 bits.

SRTP Offer and Answer Exchange

If you configure the gateway for SRTP (globally or on an individual dial peer) and end-to-end TLS, an outgoing INVITE message has cryptographic parameters in the SDP.

If you use the **srtp fallback** command and the called endpoint does not support SRTP (offer is rejected with a 4*xx* class error response), the gateway or Cisco Unified Border Element sends an RTP offer SDP in a new INVITE request. If you do not configure the **srtp fallback** command, the call fails.

Note In Cisco IOS Release 12.4(20)T and later releases (and, for Cisco UBEs, in Cisco IOS Release 12.4(22)T and later releases), calls established with SRTP only (and not SIPS) will succeed even if the **srtp fallback** command is not configured.

When a gateway or Cisco Unified Border Element receives an SRTP offer, negotiation is based on the inbound dial peer if specified and, if not, the global configuration. If multiple cryptographic attributes are offered, the gateway selects an SRTP offer it supports (AES_CM_128_HMAC_SHA1_32). The cryptographic attribute will include the following:

- The tag and same crypto suite from the accepted cryptographic attribute in the offer.
- A unique key the gateway generates from the SRTP library API.
- Any negotiated session parameters and its own set of declarative parameters, if any.

If this cryptographic suite is not in the list of offered attributes, or if none of the attributes are valid, the SRTP negotiation fails. If the INVITE message contains an alternative RTP offer, the gateway or Cisco Unified Border Element negotiates and the call falls back to (nonsecure) RTP mode. If there is no alternative offer and the SRTP negotiation fails, the INVITE message is rejected with a 488 error (Not Acceptable Media).

Rekeying Rules

There is no rekeying on an SRTP stream. A REINVITE/UPDATE message is used in an established SIP call to update media-related information (codec, destination address, and port number) or other features, such as call-hold. A new key need only be generated if the offer SDP has a new connection address or port. Because the source connection address and port do not change, the gateway or Cisco Unified Border Element will not generate a new master key after a key has been established for an SRTP session.

Call-Feature Interactions

This section describes call-feature interactions when SIP Support for SRTP features are configured.

Call Hold

If a gateway receives a call hold REINVITE message after an initial call setup is secured, the gateway places the existing SRTP stream on hold, and its answer in the 200 OK message depends on the offer SDP. If there is a cryptographic attribute in the offer, the gateway responds with a cryptographic attribute in its answer.

Signaling Forking

A proxy can fork an INVITE message that contains an SRTP offer, which can result in multiple SRTP streams until a 200 OK message is received. Because the gateway always honors the last answer, the gateway deletes previous SRTP streams and creates a new stream to the latest endpoint. Other endpoints might also stream to the gateway, but because the DSP knows only the last streams's cryptographic suite and key, authentication on these packets fails, and the packets are dropped.

Call Redirection

A gateway redirects a call when an INVITE message, sent to a proxy or redirect server, results in a 3xx response with a list of redirected contact addresses. The gateway handles a 3xx response based on the schema

in the contact of a 3xx message. If the message is SIP, and you configure the call for SRTP with fallback, the gateway offers an SRTP-only redirected INVITE message. If you configure for SRTP only, the offer is SRTP only.

If the schema is SIP, and you use the **srtp fallback** command to configure the call for RTP with fallback, the INVITE message has an RTP offer. If you do not configure the **srtp fallback** command, the call fails.

Note In Cisco IOS Release 12.4(20)T and later releases (and, for Cisco UBEs, in Cisco IOS Release 12.4(22)T and later releases), calls established with SRTP only (and not SIPS) will succeed even if the **srtp fallback** command is not configured.

Call Transfer

The SIP Support for SRTP feature interaction with call transfer depends on your outbound dial peer or global configuration. During a call transfer, the gateway sends an INVITE message to establish the connection to the transfer target. The gateway includes an SRTP offer in the INVITE message if the outbound dial peer or global configuration includes the SRTP offer.

T.38 Fax

The T.38 transport supported is User Datagram Protocol (UDP). A T.38 call is initiated as a voice call, which can be RTP or SRTP, and when it switches to T.38 fax mode, the fax call is not secure. When the fax is switched back to voice, the call returns to its initial voice state.

Conferencing Calls

For conferencing calls, the incoming INVITE message does not match any inbound dial peer and the message body is sent to the application in a container. The conferencing application performs the necessary negotiation and replies through PROGRESS or CONNECT events.

How to Configure SIP Support for SRTP

Before configuring SIP support for SRTP on a gateway or Cisco Unified Border Element, it is strongly recommended you first configure SIPS either globally or on an individual dial peer basis. The configuration on a dial peer overrides the global configuration.

Configuring SIPS Globally

To configure secure SIP (SIPS) globally on a Cisco IOS voice gateway or Cisco Unified Border Element, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service** {**pots** | **voatm** | **vofr** | **voip**}
4. **sip**

5. **url sips**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service** {**pots** | **voatm** | **vofr** | **voip**} **Example:** `Router(config)# voice service voip`	Enters voice service configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**url sips** **Example:** `Router(conf-serv-sip)# url sips`	Specifies generation of URLs in SIPS format for VoIP calls for all dial peers on the voice gateway or Cisco UBE.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIPS on a Dial Peer

To configure secure SIP (SIPS) on an individual dial peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **vofr** | **voip**}
4. **voice-class sip url sips**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* {**pots** | **vofr** | **voip**} **Example:** `Router(config)# dial-peer voice 111 voip`	Enters dial peer voice configuration mode.
Step 4	**voice-class sip url sips** **Example:** `Router(config-dial-peer)# voice-class sip url sips`	Specifies configuration of URLs in SIPS format for VoIP calls for a specific dial peer.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring SRTP and SRTP Fallback Globally

To configure SRTP and SRTP fallback behavior globally on a Cisco IOS voice gateway or Cisco Unified Border Element, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service** {**pots** | **voatm** | **vofr** | **voip**}
4. **srtp**
5. **srtp fallback**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service** {**pots** | **voatm** | **vofr** | **voip**} **Example:** `Router(config)# voice service voip`	Enters voice service configuration mode.
Step 4	**srtp** **Example:** `Router(conf-voi-serv)# srtp`	Configures secure RTP calls.
Step 5	**srtp fallback** **Example:** `Router(conf-voi-serv)# srtp fallback`	(Optional) Configures a fallback to RTP calls in case secure RTP calls fail due to lack of support from an endpoint.
Step 6	**exit** **Example:** `Router(conf-voi-serv)# exit`	Exits the current mode.

Configuring SRTP and SRTP Fallback on a Dial Peer

To configure SRTP and SRTP fallback behavior on an individual dial peer that overrides the global SRTP configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **vofr** | **voip** }
4. **srtp**
5. **srtp fallback**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* {**pots** | **vofr** | **voip** } **Example:** `Router(config)# dial-peer voice 111 voip`	Enters dial peer voice configuration mode.
Step 4	**srtp** **Example:** `Router(config-dial-peer)# srtp`	Configures secure RTP calls.
Step 5	**srtp fallback** **Example:** `Router(config-dial-peer)# srtp fallback`	(Optional) Configures a fallback to RTP calls in case secure RTP calls fail due to lack of support from an endpoint.
Step 6	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuration Examples for Configuring SIP Support for SRTP

The following example shows how to configure SIPS globally on a Cisco IOS voice gateway or Cisco Unified Border Element:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# voice service voip
Router(conf-voi-serv)# sip
Router(conf-serv-sip)# url sips
Router(conf-serv-sip)# exit
```

The following example shows how to configure SIPS on dial peer 111:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# dial-peer voice 111 voip
```

```
Router(config-dial-peer)# voice-class sip url sips
Router(config-dial-peer)# exit
```

The following example shows how to configure for SRTP with fallback to RTP globally on a Cisco IOS voice gateway or Cisco Unified Border Element:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# voice service voip
Router(conf-voi-serv)# srtp
Router(conf-voi-serv)# srtp fallback
Router(conf-voi-serv)# exit
```

The following example shows how to configure for SRTP with fallback to RTP on dial peer 111:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# dial-peer voice 111 voip
Router(config-dial-peer)# srtp
Router(config-dial-peer)# srtp fallback
Router(config-dial-peer)# exit
```

Additional References

The following sections provide references related to configuring the SIP Support for SRTP features.

Related Documents

Related Topic	Document Title
Cisco IOS dial peer overview	"Dial Peer Overview"
Cisco IOS dial technologies command information	*Cisco IOS Dial Technologies Command Reference*
Cisco IOS SIP overview and related documents	"Overview of SIP"
Cisco IOS software configuration guides	• *Cisco IOS Dial Technologies Configuration Guide* • *Cisco IOS SIP Configuration Guide* **Note** To locate the configuration guide specific to your Cisco IOS software release, choose the **Cisco IOS and NX-OS Software** category on the Product Support page and navigate according to your release (http://www.cisco.com/web/psa/products/index.html) .
Cisco IOS voice command information	*Cisco IOS Voice Command Reference*
Cisco IOS voice configuration information	*Cisco IOS Voice Configuration Library*

Related Topic	Document Title
Cisco Unified Border Element configuration information	*Cisco Unified Border Element Configuration Guide*
Cisco Unified CME command information	*Cisco Unified Communications Manager Express Command Reference*
Cisco Unified CME configuration information	Cisco Unified CME Support Documentation Home Page

RFCs

RFC	
draft-ietf-mmusic-sdescriptions-08.txt	Session Description Protocol Security Descriptions for Media Streams
RFC 3263	Session Initiation Protocol (SIP): Locating SIP Servers
RFC 3711	The Secure Real-time Transport Protocol (SRTP)

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring SIP Support for SRTP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 62: Feature Information for Configuring SIP Support for SRTP

Feature Name	Releases	Feature Information
SIP Support for SRTP	12.4(15)T	This feature introduces SIP support for supplementary services features, such as call hold, call transfer, call waiting, and call conference (3WC) using hook flash (HF) for FXS phones on Cisco IOS voice gateways. The following commands were introduced or modified: **srtp**, **srtp fallback**.
SIP SRTP Fallback to Nonsecure RTP	12.4(15)XY 12.4(20)T	This feature extends the existing SRTP to RTP fallback on Cisco IOS voice gateways to support a delayed offer and adds support for SRTP over SIP. The following commands were introduced or modified: **srtp negotiate**, **voice-class sip srtp negotiate**.
Interworking of Secure RTP calls for SIP and H323	12.4(20)T	This feature provides an option for a Secure RTP (SRTP) call to be connected from H323 to SIP and from SIP to SIP. Additionally, this feature extends SRTP fallback support from the Cisco IOS voice gateway to the Cisco Unified Border Element. This feature uses no new or modified commands.
SIP SRTP Fallback to Nonsecure RTP for Cisco Unified Border Elements	12.4(22)T	This feature adds support for both SRTP to RTP fallback with a delayed offer and SRTP over SIP to the Cisco Unified Border Element. This feature uses no new or modified commands.
Cisco Unified Border Element Support for SRTP-RTP Internetworking	12.4(22)YB	This feature provides the ability to support interworking between SRTP on one IP leg and RTP on another IP leg of a Cisco Unified Border Element. The following command was introduced or modified: **tls**.

Glossary

AVP --Audio/Video Profile.

CAC --Call Admission Control.

CME --Communications Manager Express.

CVP --Customer Voice Portal.

GW --gateway.

ISDN --Integrated Services Digital Network.

MIME --Multipurpose Internet Mail Extensions.

m line --The media-level section of an SDP session begins and ends with an "m" line that confines the information about the media stream.

MOH --music on hold.

OGW --originating gateway (ingress gateway).

PBX --Private Branch Exchange.

PINX --private integrated services network exchange.

PISN --private integrated services network.

QoS --quality of service.

QSIG --Q Signaling protocol.

RSVP --Resource Reservation Protocol.

RTP --Real-time Transport Protocol.

SDP --Session Description Protocol.

SIP --Session Initiation Protocol.

SRTP --Secure Real-time Transport Protocol.

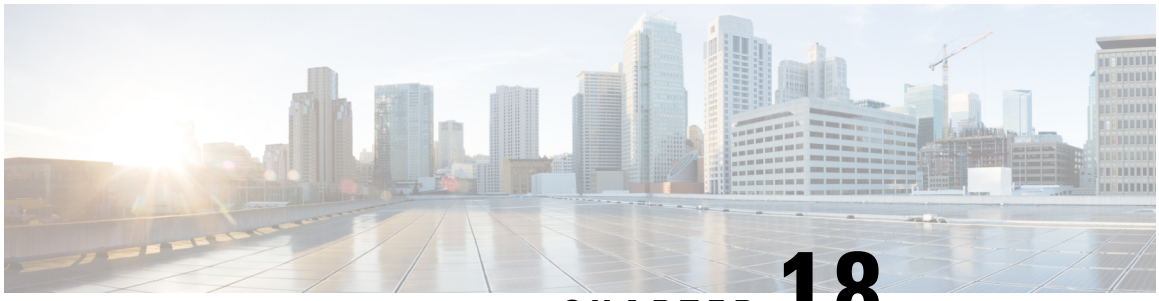
TDM --time-division multiplexing.

TGW --terminating gateway (egress gateway).

UA --user agent.

UDP --User Datagram Protocol.

URI --uniform resource identifier.



CHAPTER 18

Configuring SIP Support for Hookflash

This chapter contains information about the SIP Support for Hookflash feature that allows you to configure IP Centrex supplementary services on SIP-enabled, Foreign Exchange Station (FXS) lines. Supplementary services for the SIP Support for Hookflash feature include the following:

- Call hold
- Call waiting
- Call transfer
- 3-Way conferencing

Use the **service dsapp** command to configure supplementary Centrex-like features on FXS phones to interwork with SIP-based soft switches. The SIP Support for Hookflash feature supports the concept of a dual-line (ACTIVE and STANDBY for active and held calls) for FXS calls to support supplementary services. Hookflash triggers supplementary services based on the current state of the call.

You can configure the **service dsapp** command on individual dial peers, or configure globally for all calls entering the gateway.

Prerequisites for SIP Support for Hookflash

All Hookflash Features for FXS Ports

- Ensure that the gateway has voice functionality that is configurable for SIP.
- Establish a working IP network. For information on configuring IP, see the *Cisco IOS IP Configuration Guide*, Release 12.3.
- Configure VoIP.

Restrictions for SIP Support for Hookflash

- Release by any party other than controller causes the conference to be released when Packet Voice Digital Signal Processor (DSP) Module (PVDM2) is used.
- Release by any party on cascading 3-Way Conference, releases all the calls.
- Invocation of features such as Call Hold, Blind Transfer, Semi-Attended Transfers after establishment of 3-Way Conference, releases all the calls.
- Semi-attended transfer is not possible between users connected to gateways using G729 codec with CUCM. With G711 codec, semi-attended transfer is possible using CUCM.

Information About SIP Support for Hookflash

Use the **service dsapp**command to configure supplementary Centrex-like services on FXS phones to interwork with SIP-based softswitches. Hookflash triggers supplementary features based on the current state of the call and provides a simulation of dual-line capability for analog phones to allow one line to be active while the other line is used to control supplementary IP Centrex services. Supplementary services for the SIP Support for Hookflash feature include the following:

Call Hold

With the Call Hold feature, you can place a call on hold. When you are active with a call and you press hookflash, and there is no call that is waiting, you hear a dial tone.

If there is a call on hold, the hookflash switches between two calls; the call on hold becomes active while the active call is put on hold.

If you have a call on hold and the call hangs up, the call on hold is disconnected.

Call Holding Flows

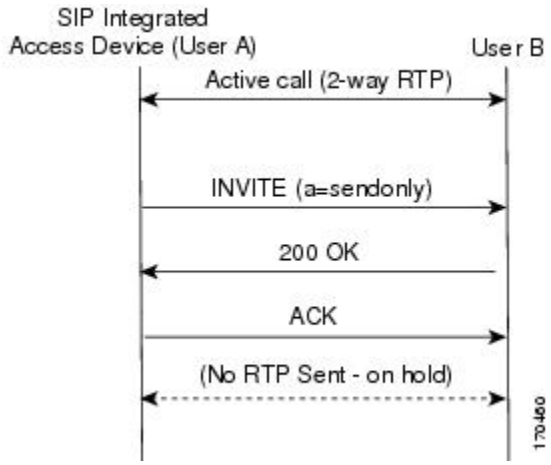
The sequence of placing a call on hold is summarized in the following steps:

1. User A and user B are active with a call.
2. By pressing hookflash, user A initiates a call hold.
3. SIP sends a call hold indication to user B.
4. User A can now initiate another active call (user C), transfer the active call (call transfer), or respond to a call-waiting indication.

Note Use the **offer call-hold** command in sip-ua configuration mode to configure the method of hold used on the gateway. For detailed information on the **offer call-hold**command, see the *Cisco IOS Voice Command Reference Guide*.

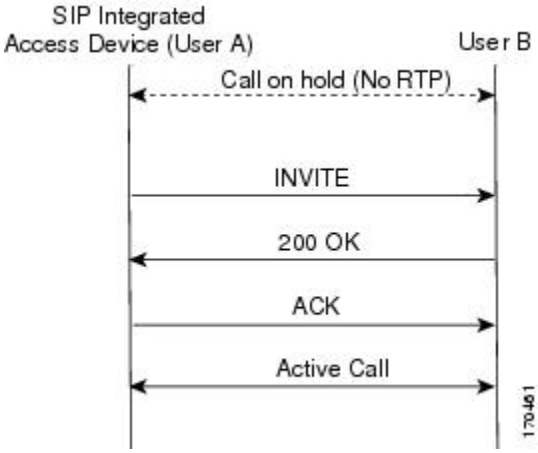
1. User A receives a second dial tone and presses hookflash.

The figure below shows the initiation of the calls hold sequence.



1. User A and User B reconnect.

The figure below shows the calls on hold resume sequence.



The table below summarizes the hookflash support for Call Hold services.

Table 63: Call Hold Hookflash Services

State	Action	Result	Response to FXS Line
Active call	Hookflash	Call placed on hold for remote party.	Second dial tone for FXS phone.
Call on hold	Hookflash	Active call.	FXS line connects to call.

State	Action	Result	Response to FXS Line
Call on hold and active call	Hookflash	Active and call on hold are swapped.	FXS line connects to previous held call.
	On hook	Active call is dropped.	Held call still active. Reminder ring on FXS line.
	Call on hold goes on hook	Call on hold is dropped.	None.
	Active call goes on hook	Active call is dropped.	Silence. Reconnects to held call after the value you specify for **disc-toggle-time** expires. See "How to Configure Disconnect Toggle Time".

Call Waiting

With the Call Waiting feature, you can receive a second call while you are on the phone with another call. When you receive a second call, you hear a call-waiting tone (a tone with a 300 ms duration). Caller ID appears on phones that support caller ID. You can use hookflash to answer a waiting call and place the previously active call on hold. By using hookflash, you can toggle between the active and a call that is on hold. If the Call Waiting feature is disabled, and you hang up the current call, the second call will hear a busy tone.

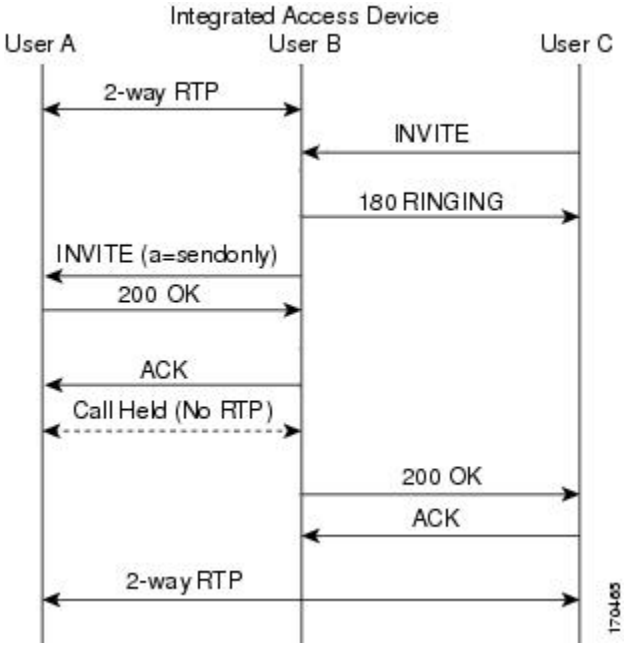
The call-waiting sequence is summarizes in the following steps:

1. User A is active with a call to user B
2. User C calls user B
3. User B presses hookflash.

The call between user A and user B is held.

1. User B connects to user C.

The figure below shows the call waiting sequence.



The table below summarizes hookflash support for Call Waiting services.

Table 64: Call Waiting Hookflash Services

State	Action	Result	Response to FXS Line
Active call and waiting call	Hookflash	Swap active call and waiting call.	FXS line connects to waiting call.
	Active call disconnects	Active call is disconnected.	Silence.
	Waiting call goes disconnects	Stay connected to active call.	None.
	Call disconnects	Active call is dropped.	Reminder ring on FXS line.

Call Transfers

Call transfers are when active calls are put on hold while a second call is established between two users. After you establish the second call and terminate the active call, the call on hold will hear a ringback. The Call Transfer feature supports all three types of call transfers--blind, semi-attended, and attended.

Blind Call Transfer

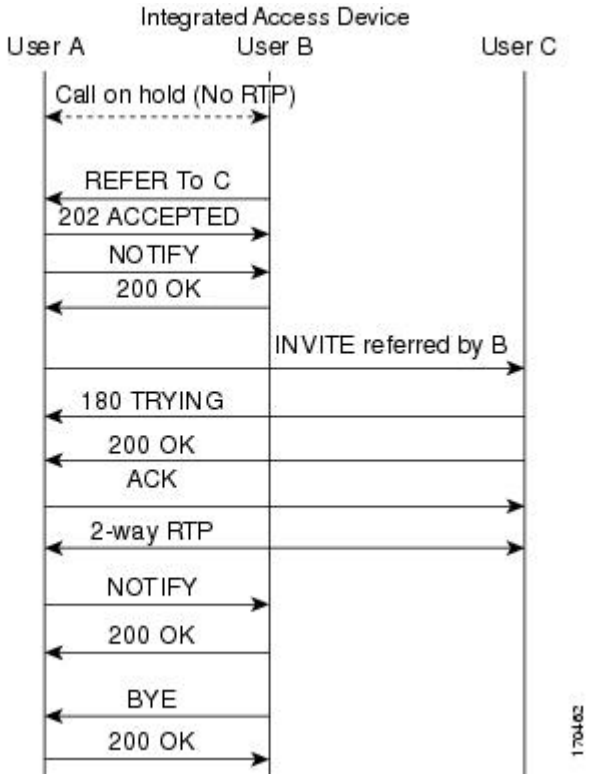
The following describes a typical Blind call-transfer scenario:

1. User A calls user B.
2. User B (transferrer) presses hookflash, places user A (transferee) on hold, and dials user C (transfer-to).

Note User B will not hear alerting for the time you configure. See "How to Configure Blind Transfer Wait Time".

1. Before the Blind call transfer trigger timer expires, user B disconnects, and the call between user A and user B is terminated.
2. User A is transferred to user C and hears a ringback if user C is available. If user C is busy, user A hears a busy tone; if user C is not busy and answers, user A and user C connect.

The figure below shows the call sequence for a Blind call transfer.

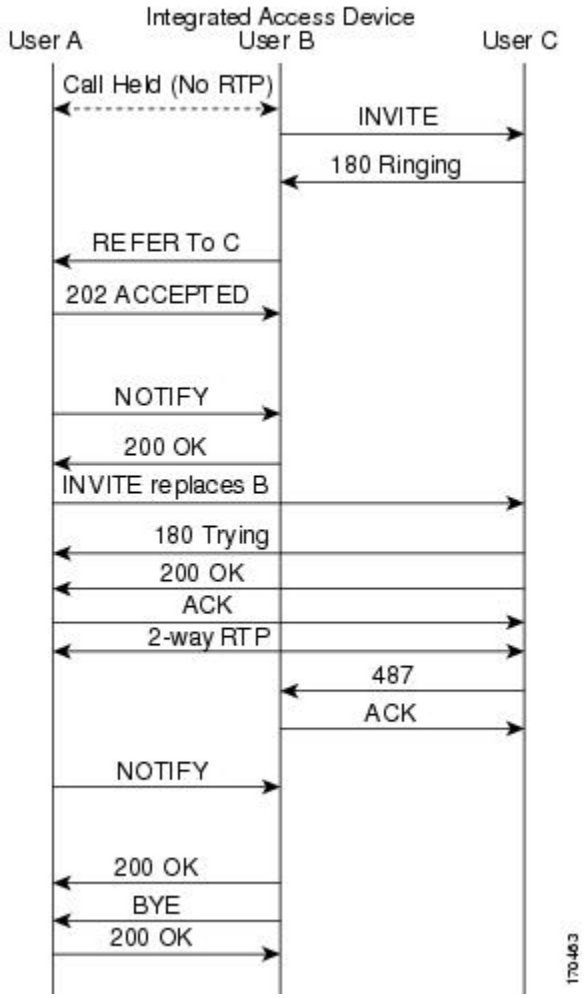


Semi-Attended Transfers

The following is a typical semi-attended transfer scenario:

1. User A calls user B.
2. User B places user A on hold and dials user C.
3. After user B hears a ringback and user C rings, user B initiates a transfer, and the call between user A and user B is terminated.
4. User A is transferred to user C and hears a ringback if user C is available. If user C is busy, user A hears a busy tone.
5. If user C is not busy and answers, user A and user C connect.

The figure below shows the call details for a semi-attended transfer.

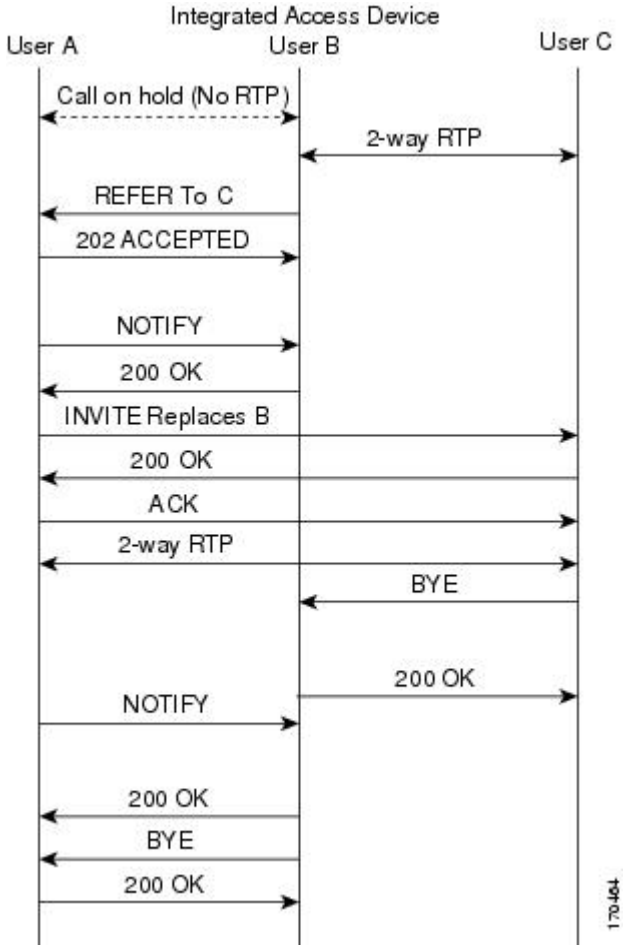


Attended Transfers

The following describes a typical attended transfer:

1. User A calls user B.
2. User B places user A on hold and dials user C.
3. After user C answers, user B goes on-hook to initiate a transfer, and the call between user A and user B is terminated.
4. User A is transferred to user C. If user C is busy when user B calls, user A hears a busy tone.
5. If user C is not busy and answers, user A and user C connect.

The figure below shows the call details for an attended transfer.



The table below summarizes the hookflash support for Call Transfer services.

Table 65: Call Transfer Hook Flash Services

State	Action	Result	Response to FXS Line
Active call	Hookflash.	Call placed on hold.	Second dial tone.
Call on hold and outgoing dialed or alerting, or active call	On hook.	Call on hold and active call transferred.	--
Call on hold and outgoing alerting call	Hookflash	Active call dropped.	FXS line connects to call o hold.

3-Way Conference

You can use the 3-Way Conference feature to establish two calls with a single connection so that all three parties can talk together. If the 3-Way Conference feature is disabled, a second hookflash will toggle between the two calls.

Note The 3-Way Conference feature supports only those SIP calls that use the g711 or g729 codecs. This feature also supports specification GR-577-CORE.

Setting Up a 3-Way Conference

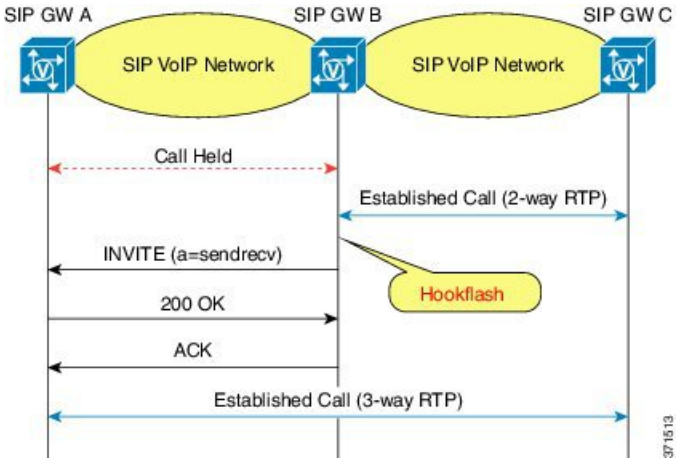
The following describes a typical 3-way conference scenario:

1. User A is talking with user B (a second-party call).
2. User A presses hookflash, receives a dial-tone, and dials user C.
3. User C answers. User A and user C are active in a second-party call.
4. User A presses hookflash to activate a 3-way conference.

In other terminology, user A is the host or controller; user B is the original call; and user C is the add-on.

Note The 3-Way Conference feature is available when the second-party call is outgoing. If the second-party call is incoming and you press hookflash, the phone toggles between the two calls.

The figure below shows the call details for 3-way conferencing.



The table below summarizes the hookflash support for 3-way conferencing services.

Table 66: 3-Way Conference Hookflash Services

State	Action	Result	Response to FXS Line
Active call	Hookflash	Call place on hold.	Second dial tone
Call on hold and active call		Join call on hold and active call.	Media mixing of both calls

Terminating a 3-Way Conference

The table below summarizes the termination of a 3-way conference:

Table 67: 3-Way Conference Termination

State	Action	Result	Response to FXS Line
Active 3-way conference	User A disconnects first	3-Way conference terminates; all users are disconnected.	Dial tone
	User B disconnects first	User A and user C establish a second-party call.	FXS line connects user A and user C.
	User C disconnects first	User A and user B establish a second-party call.	FXS line connects user A and user B.
	User A presses hookflash	User C disconnects and user A and user B establish a second-party call.	FXS line connects user A and user B.

How to Configure and Associate SIP Support for Hookflash

This section describes the procedures for configuring and associating the SIP Support for Hookflash feature. These procedures include the following:

1. Configuring supplementary service by using the **service dsapp** command.
2. Associating the supplementary services with configured dial peers.

or

Associating the supplementary services as the global default application on a gateway.

This section provides configurations for the following supplementary services and provides configuration for associating supplementary services with dial peers:

How to Configure Call Hold

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param callHold TRUE**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param callHold TRUE** **Example:** `Router(config-app-param)# param callHold TRUE`	Enables call hold.
Step 6	**exit** **Example:** `Router (config-app-param)# exit`	Exits the current mode.

How to Configure Call Waiting

To enable call waiting for a DSAPP, follow these steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param callWaiting TRUE**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param callWaiting TRUE** **Example:** `Router(config-app-param)# param callWaiting TRUE`	Enables call waiting.
Step 6	**exit** **Example:** `Router (config-app-param)# exit`	Exits the current mode.

How to Configure Call Transfer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param callTransfer TRUE**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router (config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param callTransfer TRUE** **Example:** `Router(config-app-param)# param callTransfer TRUE`	Enables call transfer.
Step 6	**exit** **Example:** `Router(config-app-param)# exit`	Exits the current mode.

How to Configure 3-Way Conferencing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param callConference TRUE**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router (config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param callConference TRUE** **Example:** `Router(config-app-param)# param callConference TRUE`	Enables 3-way conferencing.
Step 6	**exit** **Example:** `Router(config-app-param)# exit`	Exits the current mode.

How to Configure Disconnect Toggle Time

You can configure the time to wait before switching to a call on hold if an active call disconnects (commonly known as disconnect toggle time). You can configure a time-to-wait range between 10 (default) and 30 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param disc-toggle-time** *seconds*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router (config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param disc-toggle-time** *seconds* **Example:** `Router(config-app-param)# param disc-toggle-time 20`	Sets the time to wait before switching to a call on hold, if the active call disconnects (disconnect toggle time). You can specify a disconnect toggle time between 10 (default) and 30 seconds.
Step 6	**exit** **Example:** `Router(config-app-param)# exit`	Exits the current mode.

How to Configure Blind Transfer Wait Time

To configure the time the system waits before establishing a call, so that you can transfer a call by placing the phone on hook, proceed with the following steps.

Note The transferrer will not hear the alert for the time you configure because the system delays the call in case blind transfer is initiated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **application**
4. **service dsapp**
5. **param blind-xfer-wait-time** *time*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router(config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param blind-xfer-wait-time** *time* **Example:** `Router(config-app-param)# param blind-xfer-wait-time 10`	Enables call waiting.
Step 6	**exit** **Example:** `Router (config-app-param)# exit`	Exits the current mode.

How to Associate Services with a Fixed Dial Peer

After you have enabled and customized your services on a gateway by using the **service dsapp** command, you must associate these services with configured dial peers. You can associate individual dial peers, or alternately, you can configure these services globally on the gateway (see "How to Associate Services Globally on a Gateway"). If you associate these services globally, all calls entering from the FXS line side and from the SIP trunk side invoke the **service dsapp** services.

To configure a fixed dial peer used by DSAPP to set up a call to the SIP server (trunk) side, proceed with the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **service dsapp**
5. **param dialpeer** *dial-peer-tag*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**service dsapp** **Example:** `Router (config-app)# service dsapp`	Enters DSAPP parameters mode.
Step 5	**param dialpeer** *dial-peer-tag* **Example:** `Router(config-app-param)# param dialpeer 5000`	Configures a fixed dial peer used by DSAPP to set up a call to the SIP server (trunk) side, where *dial-peer-tag* is the tag of the dial peer used to place an outgoing call on the IP trunk side. The *dial-peer-tag* must be the same tag as the dial peer configured to the SIP server.
Step 6	**exit** **Example:** `Router(config-app-param)# exit`	Exits the current mode.

How to Associate Services Globally on a Gateway

After you have enabled and customized your services on a gateway by using the **service dsapp** command, you must associate these services with configured dial peers. You can associate individual dial peers ("How to Associate Services with a Fixed Dial Peer"), or alternately, you can configure these services globally on the gateway. If you associate these services globally, all calls entering from the FXS line side and from the SIP trunk side will invoke the **service dsapp** services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **application**
4. **global**
5. **service default dsapp**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**application** **Example:** `Router(config)# application`	Enters SIP gateway-application configuration mode.
Step 4	**global** **Example:** `Router (config-app)# global`	Enters SIP gateway-application-global configuration mode.
Step 5	**service default dsapp** **Example:** `Router (config-app-global)# service default dsapp`	Globally sets dsapp as the default application. All calls entering the gateway (from the FXS line side and the SIP trunk side) invoke the dsapp application.
Step 6	**exit** **Example:**	Exits the current mode.

	Command or Action	Purpose
	`Router(config-app-global)# exit`	

Verifying SIP Support for Hookflash

After the 3-way conference is established, perform this task to verify the codec used for the conference.

SUMMARY STEPS

1. **enable**
2. **show call active voice compact**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Step 2 **show call active voice compact**

Example:

```
Device# show call active voice compact

<callID>  A/O FAX T<sec> Codec       type        Peer Address       IP R<ip>:<udp>
Total call-legs: 3
   6358  ANS   T209     g729br8      VOIP        P1006            9.40.3.244:16442
   6359  ORG   T210     g729br8      TELE        P1995
   6363  ORG   T175     g729br8      VOIP        P1111008       9.40.3.245:16386
```

Troubleshooting SIP Support for Hookflash

You can use the following commands to troubleshoot the SIP Support for Hookflash feature:

- **debug voice application session**
- **debug ccsip all**(SIP message level debug)
- **debug voice ccapi inout**

Configuration Examples for SIP Support for Hookflash

Configuring Call Hold Example

```
Gateway#
 configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)#
 application
Gateway(config-app)# service dsapp
Gateway
(config-app-param)# param callHold TRUE
```

Configuring Call Waiting Example

```
Gateway#
 configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)#
 application
Gateway(config-app)# service dsapp
Gateway
(config-app-param)# param callWaiting TRUE
```

Configuring Call Transfer Example

```
Gateway#
 configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)#
 application
Gateway(config-app)# service dsapp
Gateway
(config-app-param)# param callTransfer TRUE
```

Configuring 3-Way Conferencing Example

```
Gateway#
 configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)#
 application
Gateway(config-app)# service dsapp
Gateway
(config-app-param)# param callConference TRUE
```

Configuring Disconnect Toggle Time Example

In this example, a disconnect toggle time is configured; the toggle time specifies the amount of time in seconds the system waits before committing the call transfer, after the originating call is placed on hook.

```
Gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)# application
Gateway(config-app)# service dsapp
Gateway(config-app-param)# param disc-toggle-time 10
```

Configuring Blind Transfer Wait Time Example

In this example, a blind transfer wait time is configured that specifies the amount of time in seconds the system waits before committing the call transfer after the originating call is placed on hook.

```
Gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)# application
Gateway(config-app)# service dsapp
Gateway(config-app-param)# param blind-xfer-wait-time 10
```

Configuring a Fixed Dial Peer Used for Outgoing Calls on SIP Trunk Side Example

In this example, a fixed dial peer is configured to set up the call to the SIP server (trunk) side.

```
Gateway# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Gateway(conf)# application
Gateway(config-app)# service dsapp
Gateway(config-app-param)# param dialpeer 5000
```

Associating Services with a Fixed Dial Peer Example

In this example, a fixed dial peer is configured to set up the call to the SIP server (trunk) side. The line in bold shows the dial peer statement.

```
Gateway# show running log
.
!
application
 service dsapp
 param dialpeer 1234
 param disc-toggle-time 15
 param callWaiting TRUE
 param callConference TRUE
 param blind-xfer-wait-time 10
 param callTransfer TRUE
!
voice-port 1/0/0
 station-id-name Example1
 station-id number 1234567890
```

```
!
voice-port 1/0/1
 station-id-name Example2
 station-id number 1234567891
!
voice-port 1/0/2
 station-id-name Example31
 station-id number 1234567892
!
dial-peer voice 1234 voip
 service dsapp
 destination-pattern.T
 session protocol sipv2
 session target ipv4:10.1.1.1
 dtmf-relay rtp-nte
 codec g711ulaw
!
dial-peer voice 9753 voip
 service dsapp
 destination-pattern.T
 session protocol sipv2
 session target ipv4:15.0.0.15
 dtmf-relay rtp-nte
 codec g729r8
!
dial-peer voice 100 pots
 service dsapp
 destination-pattern.1234567890
 port 1/0/0
 prefix 1234567890
!
dial-peer voice 101 pots
 service dsapp
 destination-pattern.1234567891
 port 1/0/1
 prefix 1234567891
!
dial-peer voice 102 pots
 service dsapp
 destination-pattern.1234567892
 port 1/0/2
 prefix 1234567892
!
!
sip-ua
 registar ipv4:10.1.1.1 expires 3600
!
```

Associating Services Globally on a Gateway Example

In this example, the gateway is associated globally with supplementary services. The lines in bold show the dial peer statement.

```
Gateway# show running log
.
!
application
 service dsapp
 param disc-toggle-time 15
 param callWaiting TRUE
 param callConference TRUE
 param blind-xfer-wait-time 10
```

```
 param callTransfer TRUE
!
voice-port 1/0/0
 station-id-name Example1
 station-id number 1234567890
!
voice-port 1/0/1
 station-id-name Example2
 station-id number 1234567891
!
voice-port 1/0/2
 station-id-name Example31
 station-id number 1234567892
!
dial-peer voice 1234 voip
 service dsapp
 destination-pattern 1800T
 session protocol sipv2
 session target ipv4:10.1.1.1
 dtmf-relay rtp-nte
 codec g729r8
!
dial-peer voice 9753 voip
 service dsapp
 destination-pattern.T
 session protocol sipv2
 session target ipv4:10.1.1.1
 dtmf-relay rtp-nte
 codec g711ulaw
!
dial-peer voice 100 pots
 preference 8
 service dsapp
 destination-pattern.6234567890
 port 1/0/0
 prefix 6234567890
!
dial-peer voice 101 pots
 preference 8
 service dsapp
 destination-pattern.6234567892
 port 1/0/1
 prefix 6234567892
!
dial-peer voice 102 pots
 preference 8
 service dsapp
 destination-pattern.6234567893
 port 1/0/2
 prefix 6234567893
!
!
dial-peer hunt 2
!
sip-ua
 registar ipv4:10.1.1.1 expires 3600
!
```

Additional References

The following sections provide references related to the SIP Support for Hookflash feature.

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

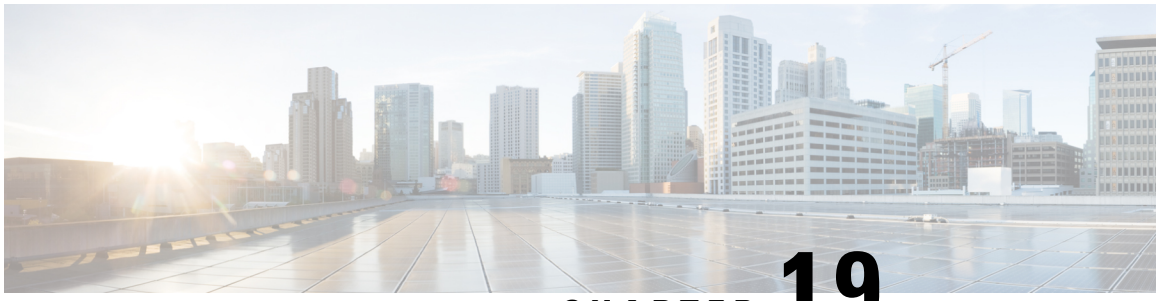
Feature Information for SIP Support for Hookflash

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 68: Feature Information for SIP Support for Hookflash

Feature Name	Releases	Feature Information
SIP Support for Hookflash	12.4(11)T	This feature was introduced. SIP Support for Hookflash feature allows you to configure IP Centrex supplementary services on SIP-enabled, Foreign Exchange Station (FXS) lines.
SIP Support for Hookflash	15.4(2)T	The feature was enhanced to support hookflash using G729 codec for 3-way conference.



CHAPTER 19

SIP Warning Header

The Warning Header text and Warning Code in a Session Initiation Protocol (SIP) response are used to point to the exact cause of failure. All system failures are, by default, reported in the warning header of a SIP error response 3xx/4xx/5xx. Reporting of certain failures (categorized as threshold failures), which disclose system capacity can be controlled.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About SIP Warning Header Debugging

The system level failures, which would be enhanced by returning a Warning Header with specific text in the appropriate SIP response to point to the exact cause of failure are as follows:

Failure Scenarios	SIP Warning Header Phrase
Dial-peer mismatch	“No matching outgoing dial-peer”
Spike count exceeds call spike threshold (Threshold Failure)	“Call Spiked. Concurrent calls exceed Spike Count Threshold”
Maximum number of connections set per peer exceeded (Threshold Failure)	“Maximum Number of Connections reached”
QoS Negotiation	“QoS Negotiation Failure”

Failure Scenarios	SIP Warning Header Phrase
SRTP fallback failure	“Cannot fallback to RTP. SRTP configured on dial peer”
Transcoder not configured	“Transcoder Not Configured”
Transcoder reservation failure during SRTP-RTP Interworking	“Transcoder reservation failure during SRTP-RTP I/W”
Transcoder reservation for transrating scenario	“Transcoder reservation failure for transrating scenario”
SIP Service Shutdown	“SIP Service is Shutdown”
Error binding the local IP Address due invalid GW mode	“Local IP Bind Failure. Invalid GW mode of operation”
Inconsistent URI Scheme in a SIP Message	“Inconsistent URI scheme in Req-URI/To Header/Contact Header”
Join header processing error	“Error parsing Join Header”
Proxy-Authorization Header	“Error parsing Proxy-Authorization Header”
SIP Registrar disabled	“Registrar is not enabled”
SIP Registrar process not up	“Registrar process is not up”

The **warn-header ext-text all** command under voice service SIP configuration mode is used to enable or disable SIP warning header debugging. The default behavior is to enable sending notification of all system failures in the SIP warning header.

How to Configure SIP Warning Header

Configuring SIP Warning Header Debugging

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **warn-header ext-text all**
6. **warn-header ext-text suppress threshold-failures**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters global VoIP configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters voice service SIP configuration mode.
Step 5	**warn-header ext-text all** **Example:** `Device(conf-serv-sip)# warn-header ext-text all`	Enables sending notification of all system failures in SIP Warning Header. **Note** Use the **no warn-header ext-text all** command to disable sending notification of all system failures in SIP Warning Header.
Step 6	**warn-header ext-text suppress threshold-failures** **Example:** `Device(conf-serv-sip)# warn-header ext-text suppress threshold-failures`	Disables sending notification of threshold failures in SIP Warning Header. **Note** Use the **no warn-header ext-text suppress threshold-failures** command to enable sending notification of threshold failures in SIP Warning Header.
Step 7	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Troubleshooting and Debugging SIP Warning Header

SUMMARY STEPS

1. **enable**
2. **debug ccsip messages**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Step 2 **debug ccsip messages**

Shows all Session Initiation Protocol (SIP) Service Provider Interface (SPI) message tracing. This is a sample output showing the SIP response message with SIP Warning Header.

Example:

```
Device# debug ccsip messages

SIP/2.0 500 Internal Server Error Via: SIP/2.0/UDP 9.45.33.11:5081;branch=z9hG4bK-10854-1-0
  From: sipp <sip:111@9.45.33.11:5081>;tag=10854SIPpTag001
  To: sut <sip:222@9.43.29.50:5060>;tag=38404-23F2
  Date: Tue, 05 Feb 2013 05:49:37 GMT
  Call-ID: 1-10854@9.45.33.11
  CSeq: 1 INVITE
  Allow-Events: telephone-event
  Warning: 399 9.43.29.50 "Transcoder Not Configured"  //SIP response message with SIP Warning Header//

  Server: Cisco-SIPGateway/IOS-15.3.20130202.123643.
  Reason: Q.850;cause=16
  Content-Length: 0
```

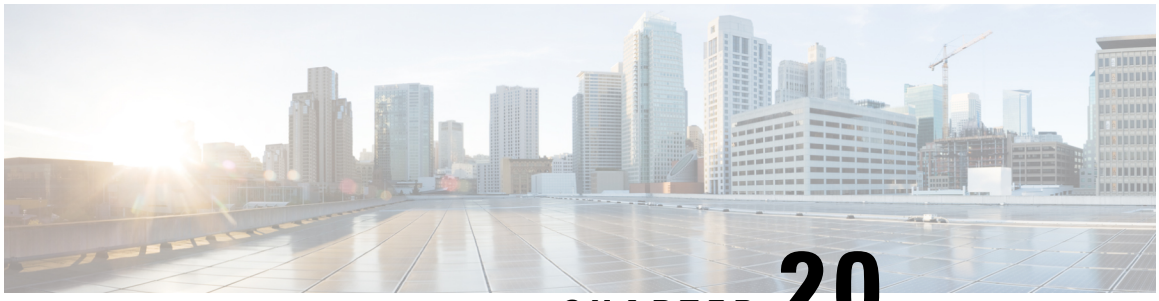
Feature Information for SIP Warning Header Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 69: Feature Information for SIP Warning Header Enhancements

Feature Name	Releases	Feature Information
SIP Warning Header Enhancements	15.3(2)T	The Warning Header text and Warning Code in a Session Initiation Protocol (SIP) response are used to point to the exact cause of failure. All system failures are, by default, reported in the warning header of a SIP error response 3xx/4xx/5xx. Reporting of certain failures (categorized as threshold failures), which disclose system capacity can be controlled.



CHAPTER 20

Verifying and Troubleshooting SIP Features

Basic Troubleshooting Procedures

Cisco routers provide numerous integrated commands to assist you in monitoring and troubleshooting your internetwork:

- **show** commands help you monitor installation behavior and normal network behavior, and isolate problem areas.
- **debug** commands help you isolate protocol and configuration problems.
- **ping** commands help you determine connectivity between devices on your network.
- **trace** commands provide a method of determining the route by which packets reach their destination.

This chapter discusses use of **show** and **debug** commands.

Note Under moderate traffic loads, **debug** commands produce a high volume of output. We therefore recommend that, as a general rule, you use **show** commands first and use **debug** commands with caution.

Generally, you should proceed as follows:

1. Determine whether or not VoIP is working.
2. Determine whether or not you can make a voice call.
3. Verify that SIP-supported codecs are used. Support for codecs varies on different platforms; use the **codec ?** command to determine the codecs available on a specific platform.
4. Isolate and reproduce the failure.
5. Collect relevant information from **show** and **debug** commands, configuration files, and protocol analyzers.
6. Identify the first indication of failure in protocol traces or internal **debug** commandoutput.

7. Look for the cause in configuration files.

Note General troubleshooting of problems affecting basic functionality such as dial peers, digit translation, and IP connectivity is beyond the scope of this chapter. For links to additional troubleshooting help, see "Additional References".

Using show Commands

To verify SIP gateway status and configuration, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. **show sip service**
2. **show sip-ua register status**
3. **show sip-ua statistics**
4. **show sip-ua status**
5. **show sip-ua timers**

DETAILED STEPS

Step 1 show sip service

Use this command to display the status of SIP call service on a SIP gateway.

The following sample output shows that SIP call service is enabled:

Example:

```
Router# show sip service
SIP Service is up
```

The following sample output shows that SIP call service was shut down with the **shutdown** command:

Example:

```
Router# show sip service
SIP service is shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop** command:

Example:

```
Router# show sip service
SIP service is shut
under 'voice service voip', 'sip' submode
```

The following sample output shows that SIP call service was shut down with the **shutdown forced** command:

Example:

```
Router# show sip service
SIP service is forced shut globally
under 'voice service voip'
```

The following sample output shows that SIP call service was shut down with the **call service stop forced** command:

Example:

```
Router# show sip service
SIP service is forced shut
under 'voice service voip', 'sip' submode
```

Step 2 **show sip-ua register status**

Use this command to display the status of E.164 numbers that a SIP gateway has registered with an external primary SIP registrar.

Example:

```
Router# show sip-ua register status
Line peer expires(sec) registered
4001 20001 596          no
4002 20002 596          no
5100 1     596          no
9998 2     596          no
```

Step 3 **show sip-ua statistics**

Use this command to display response, traffic, and retry SIP statistics, including whether call redirection is disabled.

The following sample shows that four registers were sent:

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
     Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OkSubscribe 0/0, OkNOTIFY 0/0,
      OkInfo 0/0, 202Accepted 0/0
      OkRegister 12/49
     Redirection (Inbound only except for MovedTemp(Inbound/Outbound)) :
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0/0, UseProxy 0,
      AlternateService 0
      Client Error:
      BadRequest 0/0, 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,
      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, RequestCancel 0/0,
      NotAcceptableMedia 0/0, BadEvent 0/0,
      SETooSmall 0/0
     Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
     Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
      Miscellaneous counters:
      RedirectRspMappedToClientErr 0
SIP Total Traffic Statistics (Inbound/Outbound)
      Invite 0/0, Ack 0/0, Bye 0/0,
      Cancel 0/0, Options 0/0,
      Prack 0/0, Comet 0/0,
      Subscribe 0/0, NOTIFY 0/0,
      Refer 0/0, Info 0/0
      Register 49/16
Retry Statistics
      Invite 0, Bye 0, Cancel 0, Response 0,
      Prack 0, Comet 0, Reliable1xx 0, NOTIFY 0
      Register 4
SDP application statistics:
Parses: 0, Builds 0
Invalid token order: 0, Invalid param: 0
Not SDP desc: 0, No resource: 0
Last time SIP Statistics were cleared: <never>
```

The following sample output shows the RedirectResponseMappedToClientError status message. An incremented number indicates that 3*xx* responses are to be treated as 4*xx* responses. When call redirection is enabled (default), the RedirectResponseMappedToClientError status message is not incremented.

Example:

```
Router# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
      OkInvite 0/0, OkBye 0/0,
      OkCancel 0/0, OkOptions 0/0,
      OkPrack 0/0, OkPreconditionMet 0/0,
      OKSubscribe 0/0, OkNotify 0/0,
      202Accepted 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, UseProxy 0,
      AlternateService 0
    Client Error:
      BadRequest 0/0, 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,
      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, RequestCancel 0/0
```

```
      NotAcceptableMedia 0/0, BadEvent 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
    Miscellaneous counters:
      RedirectResponseMappedToClientError 1,
SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0,
    Prack 0/0, Comet 0/0,
    Subscribe 0/0, Notify 0/0,
    Refer 0/0
Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0,
    Prack 0, Comet 0, Reliable1xx 0, Notify 0
SDP application statistics:
 Parses: 0,  Builds 0
 Invalid token order: 0,  Invalid param: 0
 Not SDP desc: 0,  No resource: 0
```

Step 4 show sip-ua status

Use this command to display status for the SIP user agent (UA), including whether call redirection is enabled or disabled.

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 1 (rfc 2052)
Redirection (3xx) message handling: ENABLED
```

Step 5 show sip-ua timers

Use this command to display the current settings for the SIP user-agent (UA) timers.

The following sample output shows the waiting time before a register request is sent--that is, the value that is set with the **timers register** command:

Example:

```
Router# show sip-ua timers
SIP UA Timer Values (millisecs)
trying 500, expires 180000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500
refer 500, register 500
```

Using debug Commands

Note Commands are listed in alphabetical order.

- Use the **debug aaa authentication** command to display high-level diagnostics related to AAA logins.
- Use the **debug asnl events**command to verify that the SIP subscription server is up. The output displays a pending message if, for example, the client is unsuccessful in communicating with the server.
- Use the debug call fallback family of commands to display details of VoIP call fallback.
- Use the **debug cch323**family of commands to provide debugging output for various components within an H.323 subsystem.
- Use the **debug ccsip**family of commands for general SIP debugging, including viewing direction-attribute settings and port and network address-translation traces. Use any of the following related commands:
  - **debug ccsip all--**Enables all SIP-related debugging
  - **debug ccsip calls**--Enables tracing of all SIP service-provider interface (SPI) calls
  - **debug ccsip error**--Enables tracing of SIP SPI errors
  - **debug ccsip events**--Enables tracing of all SIP SPI events
  - **debug ccsip info**--Enables tracing of general SIP SPI information, including verification that call redirection is disabled
  - **debug ccsip media**--Enables tracing of SIP media streams
  - **debug ccsip messages**--Enables all SIP SPI message tracing, such as those that are exchanged between the SIP user-agent client (UAC) and the access server
  - **debug ccsip preauth**--Enables diagnostic reporting of authentication, authorization, and accounting (AAA) preauthentication for SIP calls
  - **debug ccsip states**--Enables tracing of all SIP SPI state tracing
  - **debug ccsip transport**--Enables tracing of the SIP transport handler and the TCP or User Datagram Protocol (UDP) process
- Use the **debug isdn q931**command to display information about call setup and teardown of ISDN network connections (layer 3) between the local router (user side) and the network.
- Use the **debug kpm**l command to enable debug tracing of KPML parser and builder errors.
- Use the **debug radius** command to enable debug tracing of RADIUS attributes.
- Use the **debug rpms-proc preauth** command to enable debug tracing on the RPMS process for H.323 calls, SIP calls, or both H.323 and SIP calls.
- Use the debug rtr trace command to trace the execution of an SAA operation.
- Use the **debug voip** family of commands, including the following:
  - **debug voip ccapi protoheaders --**Displays messages sent between the originating and terminating gateways. If no headers are being received by the terminating gateway, verify that the **header-passing** command is enabled on the originating gateway.
  - **debug voip ivr script--**Displays any errors that might occur when the Tcl script is run

- **debug voip rtp session named-event 101 --**Displays information important to DTMF-relay debugging, if you are using codec types g726r16 or g726r24. Be sure to append the argument *101* to thecommand to prevent the console screen from flooding with messages and all calls from failing.

Sample output for some of these commands follows:

Sample Output for the debug ccsip events Command

- The example shows how the Proxy-Authorization header is broken down into a decoded username and password.

```
Router# debug ccsip events
CCSIP SPI: SIP Call Events tracing is enabled
21:03:21: sippmh_parse_proxy_auth: Challenge is 'Basic'.
21:03:21: sippmh_parse_proxy_auth: Base64 user-pass string is 'MTIzNDU2Nzg5MDEyMzQ1Njou'.
21:03:21: sip_process_proxy_auth: Decoded user-pass string is '1234567890123456:.'.
21:03:21: sip_process_proxy_auth: Username is '1234567890123456'.
21:03:21: sip_process_proxy_auth: Pass is '.'.
21:03:21: sipSPIAddBillingInfoToCcb: sipCallId for billing records =
10872472-173611CC-81E9C73D-F836C2B6@172.18.192.19421:03:21: ****Adding to UAS Request table
```

Sample Output for the debug ccsip info Command

This example shows only the portion of the debug output that shows that call redirection is disabled. When call redirection is enabled (default), there are no debug line changes.

```
Router# debug ccsip info
00:20:32: HandleUdpSocketReads :Msg enqueued for SPI with IPaddr: 172.18.207.10
:5060
00:20:32: CCSIP-SPI-CONTROL:  act_sentinvite_new_message
00:20:32: CCSIP-SPI-CONTROL:  sipSPICheckResponse
00:20:32: sip_stats_status_code
00:20:32: ccsip_get_code_class: !!Call Redirection feature is disabled on the GW
00:20:32: ccsip_map_call_redirect_responses: !!Mapping 302 response to 480
00:20:32:  Roundtrip delay 4 milliseconds for method INVITE
```

Additional References

- *Cisco IOS Debug Command Reference,* Release 12.3T
- *Cisco IOS Voice Troubleshooting and Monitoring Guide* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/voipt_c/
- *Cisco IOS Voice, Video, and Fax Configuration Guide* , Release 12.2 at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvvfax_c/index.htm
- Cisco Technical Support at http://www.cisco.com/en/US/support/index.html
- *Troubleshooting and Debugging VoIP Call Basics* at http://www.cisco.com/warp/public/788/voip/voip_debugcalls.html
- *VoIP Debug Commands* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/1700/1750/1750voip/debug.htm

INDEX

A

C

D

R

S

T

V