



Achieving SIP RFC Compliance

This chapter describes how to use or configure your Cisco SIP gateway so as to comply with published SIP standards. It discusses the following features:

- 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” on page 163.](#)

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.



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2007 Cisco Systems, Inc. All rights reserved.

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 Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note**

See “RFCs” section on page 35 for more detailed information about obsoleted, updated, and new RFCs.

Contents

- [Prerequisites for SIP RFC Compliance, page 68](#)
- [Restrictions for SIP RFC Compliance, page 68](#)
- [Information About SIP RFC Compliance, page 69](#)
- [How to Configure SIP RFC Compliance, page 101](#)
- [Configuration Examples for SIP RFC Compliance, page 113](#)
- [Additional References, page 115](#)

Prerequisites for SIP RFC Compliance

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

**Note**

For information on reliable provisional responses, see *SIP Gateway Support of RSVP and TEL URL*.

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, support for bandwidth (b=) SDP attribute equal to 0 is not supported.

Information About SIP RFC Compliance

This section contains the following information:

- [SIP RFC 2543 Compliance, page 69](#)
- [SIP RFC 2782 Compliance, page 69](#)
- [SIP RFC 3261 Compliance, page 69](#)
- [SIP RFC 3261, RFC 3262, and RFC 3264 Compliance, page 98](#)

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” section on page 68](#) and [“SIP RFC 3261 Compliance” section on page 69](#) 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

This section contains the following information about basic SIP functionality:

- [SIP Header Fields, Network Components, and Methods, page 70](#)
- [SIP Responses, page 73](#)
- [SIP SDP Usage, Transport Layer Protocols, and DNS Records, page 77](#)
- [SIP Extensions, page 78](#)
- [SIP Security, page 79](#)
- [SIP DTMF Relay, page 79](#)
- [SIP Fax Relay and T.38, page 80](#)
- [SIP URL Comparison, page 82](#)
- [487 Sent for BYE Requests, page 82](#)
- [3xx Redirection Responses, page 83](#)
- [DNS SRV Query Procedure, page 83](#)
- [CANCEL Request Route Header, page 83](#)
- [Interpret User Parameters, page 83](#)
- [user=phone Parameter, page 83](#)
- [303 and 411 SIP Cause Codes, page 83](#)
- [Flexibility of Content-Type Header, page 84](#)
- [Optional SDP “s=” Line, page 84](#)
- [Allow Header Addition to INVITEs and 2xx Responses, page 84](#)
- [Simultaneous Cancel and 2xx Class Response, page 84](#)
- [UPDATE-Request Processing, page 84](#)

SIP Header Fields, Network Components, and Methods

Table 2 through Table 4 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

Table 2 **SIP Header Fields (continued)**

Header Field	Supported by Cisco Gateways?	
CC-Diversion / Diversion	Yes	
Contact		
Content-Disposition		
Content-Encoding	No	
Content-Encoding	Yes	
Content-Language	No	
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		
Response-Key		No
Retry-After		

Table 2 *SIP Header Fields (continued)*

Header Field	Supported by Cisco Gateways?
Retry-After	Yes
Route	
RSeq	
Server	
Session-Expires	
Subject	
Supported	Yes
Timestamp	
To	
Unsupported	
User-Agent	
Via	
Warning	
WWW-Authenticate	
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.

Table 4 **SIP Methods (continued)**

Method	Supported by Cisco Gateways?
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

Table 5 through Table 10 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.

Table 5 **1xx Responses (continued)**

1xx Responses	Comments
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.
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.
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.

Table 8 **4xx Responses (continued)**

4xx Responses	Comments
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.
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.
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

Table 11 through Table 13 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

Table 14 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.

Table 14 *SIP Extensions (continued)*

SIP Extension	Comments
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.
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

Table 15 and Table 16 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.

Table 17 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

Table 18 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. Table 19 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, <i>Procedures for Real-Time Group 3 Facsimile Communication over IP Networks</i> , 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

Table 19 *T.38 Fax Requirements (continued)*

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: <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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
SIPt38-14	Configuration of SIP enabled gateways include management of SIP T.38 specific configurable choices.	Mandatory	Yes. The following are configurable: <ul style="list-style-type: none"> bitrate TCP/UDP (UDP only) hs and ls redundancy ECM

Table 19 T.38 Fax Requirements (continued)

Requirement	Description	Mandatory or Optional	Supported?
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.

With Cisco IOS Release 12.3, the *maddr* and *transport* parameters are no longer allowed in Cisco SIP gateway implementations. The *user-param* parameter is now an acceptable parameter for comparison.

If a compared parameter is omitted or not present, it is matched on the basis of its default value. [Table 20](#) 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 on page 46.

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.

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 2xx 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 3261 specification:

- [SIP UPDATE Requests, page 84](#)
- [Via Header Parameters and Merged Request Detection, page 89](#)
- [Loose-Routing and the Record-Route Header, page 89](#)
- [Multiple INVITE Requests Before a Final Response, page 89](#)
- [Mid-call Re-INVITE Request Failure, page 90](#)
- [PRACK Request with a New Offer, page 91](#)
- [Reliable Provisional Response Failure, page 91](#)

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

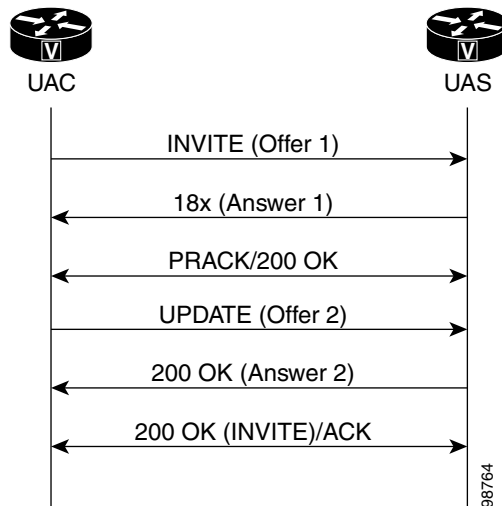
The following sections discuss how UPDATE requests are received and processed in various scenarios and call flows.

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.

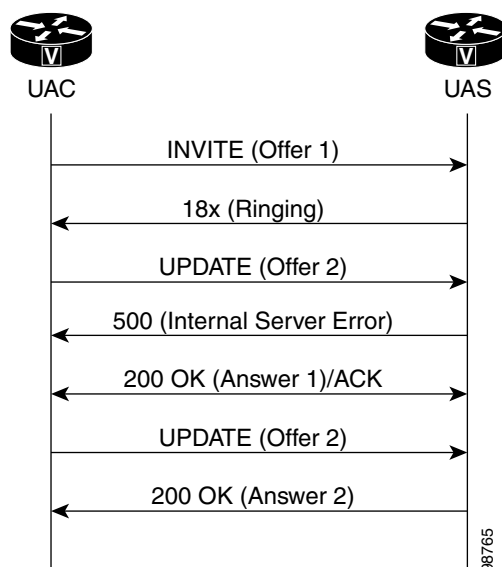
Figure 13 shows a call where the UAS sent a reliable provisional response (ANSWER 1) to an INVITE request (Offer 1). The 18x early media response indicated the gateway capability to support UPDATES. The UAC sent a provisional acknowledgement (PRACK) and received a 200 OK response to the PRACK request. 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**



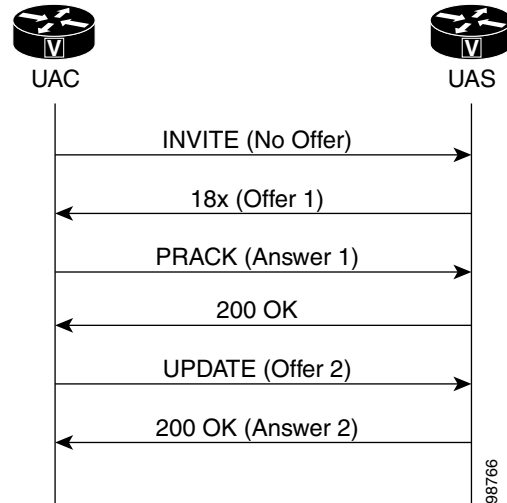
In Figure 14 the gateway received an UPDATE (Offer 2) before responding to the INVITE request (Offer 1), causing the gateway to reject the request by sending a 500 Internal Server Error with a Retry-After header field set to a randomly chosen value between zero and ten seconds.

Figure 14 **Initial UPDATE Rejected**



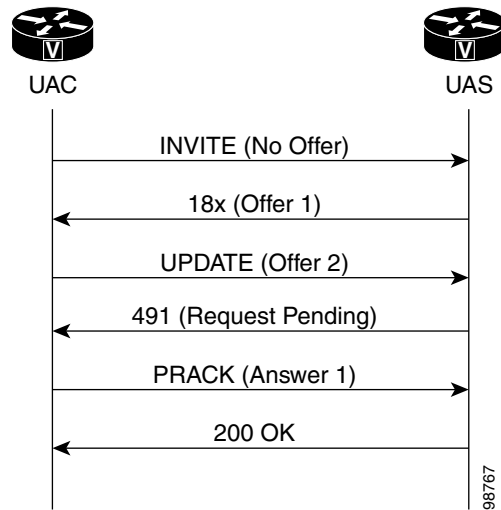
In [Figure 15](#) the initial INVITE request did not contain an offer, and the UAS gateway sent SDP with reliable provisional response (Offer 1) which was treated by the UAC as an offer.

Figure 15 UPDATE Request for Delayed Media



In [Figure 16](#) 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

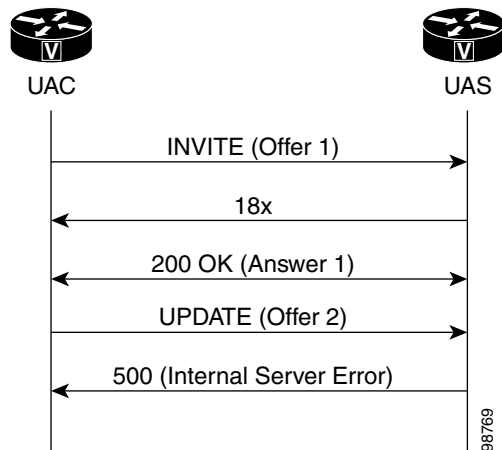


Error Responses to UPDATE Request Processing Before the Call Is Active

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 [Figure 17](#):

- 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 [Figure 18](#)).

Figure 18 UPDATE Request in the Active State

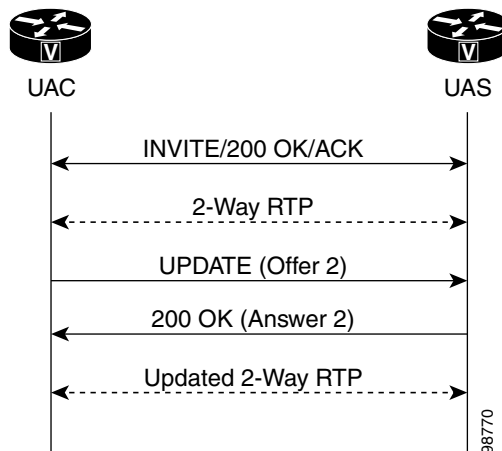
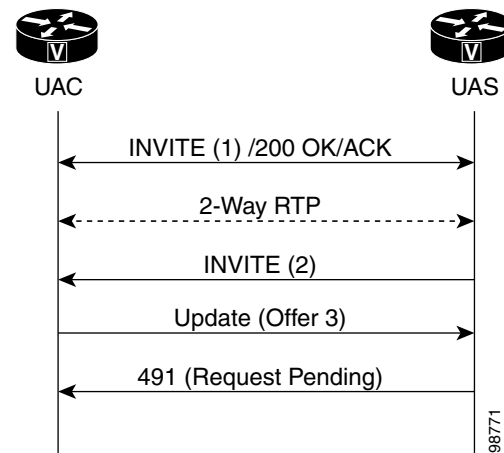


Figure 19 shows a UAC that sent a mid-call INVITE request which has not yet been answered. In this state, when the gateway receives an UPDATE request with a new offer, it sends a 491 Request Pending error.

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

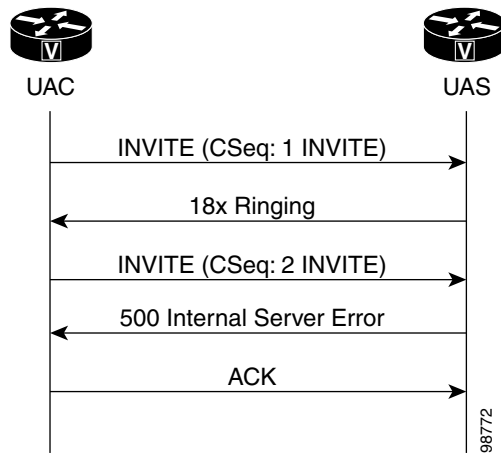
The SIP Extensions for Caller Identity and Privacy feature supports loose-routing, a mechanism that helps keep the request target and next route destination separate. The lr parameter, used in the uniform resource indicator (URI) that a proxy places in the Record-Route header, indicates proxy compatibility 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 Figure 20). 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

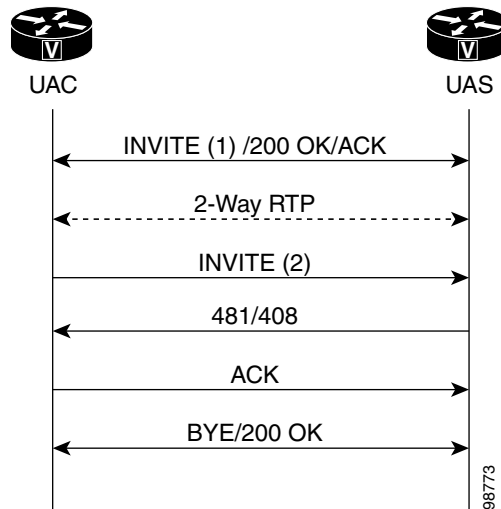


Mid-call Re-INVITE Request Failure

The SIP Extensions for Caller Identity and Privacy feature implements the mid-call re-INVITE request failure treatment shown in Figure 21. 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

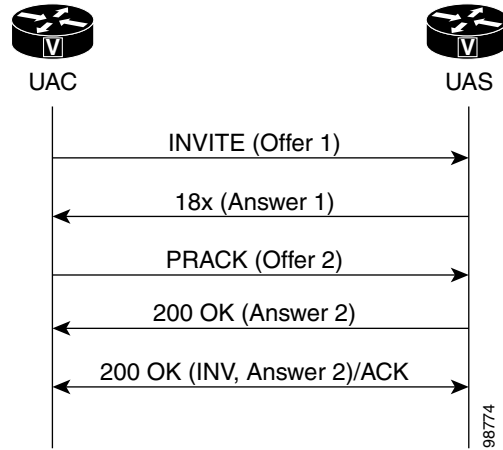
Figure 21 Dialog Termination After a 481 or 408 Response to Re-INVITE Request



PRACK Request with a New Offer

The SIP Extensions for Caller Identity and Privacy feature supports a PRACK request with a new offer (see Figure 22). 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.

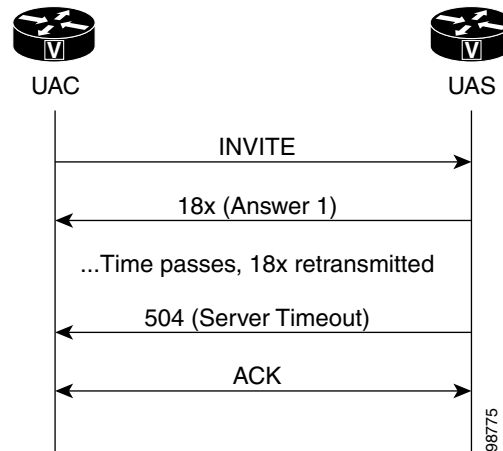
Figure 22 Offer in PRACK Accepted



Reliable Provisional Response Failure

The SIP Extensions for Caller Identity and Privacy feature provides the treatment shown in Figure 23 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.

This section contains the following information:

- [SIP Messaging Enhancements, page 98](#)
- [SIP TCP and UDP Connection Enhancements, page 99](#)
- [Dynamic Transport Switching \(UDP to TCP\) for Large SIP Requests, page 100](#)
- [Call-Hold Enhancement, page 100](#)
- [Expanded Range of the max-forwards Command, page 101](#)

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, page 99](#)
- [Reuse of TCP and UDP Connections, page 99](#)
- [Transaction-Based Transport Switching and Usage, page 99](#)
- [Detection of Remote End Connection Closures, page 100](#)
- [Creation of New Connections for Sending Responses in Case the Original Connection Dropped, page 100](#)

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 SIP gateways follow the new guideline, and SIP 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 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

This section contains the following procedures:

- [Configuring Compliance to RFC 2543, page 101](#)
- [Configuring Compliance to RFC 2782, page 101](#)
- [Configuring Compliance to RFC 3261, page 102](#)
- [Configuring Compliance to RFC 3261, RFC 3262, and RFC 3264, page 102](#)
- [Verifying SIP RFC Compliance, page 109](#)
- [Troubleshooting Tips, page 111](#)



Note

- Before you perform a procedure, familiarize yourself with the following information:
 - [“Prerequisites for SIP RFC Compliance” section on page 68](#)
 - [“Restrictions for SIP RFC Compliance” section on page 68](#)
- For help with a procedure, see the verification and troubleshooting sections listed above.

Configuring Compliance to RFC 2543

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

Configuring Compliance to RFC 2782

To configure compliance with RFC 2782, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **srv version**
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: <ul style="list-style-type: none"> • 1—Domain-name prefix of format protocol.transport. (RFC 2052 style) • 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

This section contains the following procedures:

- [Configure SIP Messaging, page 103](#)
- [Configure TCP and UDP Connection Enhancements, page 103](#)
- [Configure Dynamic Transport Switching \(UDP to TCP\) for Large SIP Requests, page 104](#)
- [Configure Call-Hold, page 107](#)
- [Configure Max Forwards, page 108](#)

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**
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	timers connection aging <i>timer-value</i> 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: <ul style="list-style-type: none"> • <i>timer-value</i>—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, page 104](#)
- [Configuring Dynamic Transport Switching for Large SIP Requests on a Global Basis, page 105](#)

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 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.

	Command or Action	Purpose
Step 3	<code>dial-peer voice tag voip</code> Example: Router(config)# dial-peer voice 25 voip	Enters dial-peer configuration mode for the specified VoIP dial peer.
Step 4	<code>voice-class sip transport switch udp tcp</code> 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: <ul style="list-style-type: none"> • 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	<code>exit</code> 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: <ul style="list-style-type: none"> • 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.

**Note**

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 the [“Verifying SIP RFC Compliance”](#) section on page 109 and [“Troubleshooting Tips”](#) section on page 111.

Configure Call-Hold

To specify how the 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**
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 SIP gateway should initiate call-hold requests. Keywords are as follows: <ul style="list-style-type: none"> • 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**
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 <i>number</i> 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: <ul style="list-style-type: none"> • <i>number</i>—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**
2. **show sip-ua statistics**

DETAILED STEPS

Step 1 **show sip-ua connections**

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.

```
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:

```
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

```

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

```

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

```

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

```

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.

```

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 on page 56.

- 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

This section provides the following configuration example:

- [SIP Gateway Compliance to RFC 3261, RFC 3262, and RFC 3264: Example, page 113](#)



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
    rellxx 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

General SIP References

- “[SIP Features Roadmap](#)” on page 1—Describes how to access Cisco Feature Navigator; also lists and describes, by Cisco IOS release, SIP features for that release.
- “[Overview of SIP](#)” on page 9—Describes underlying SIP technology; also lists related documents, standards, MIBs, RFCs, and how to obtain technical assistance.

References Mentioned in This Chapter

- *SIP Gateway Support of RSVP and TEL URL* at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122x/122xb/122xb_2/vvfresrv.htm#1027153

