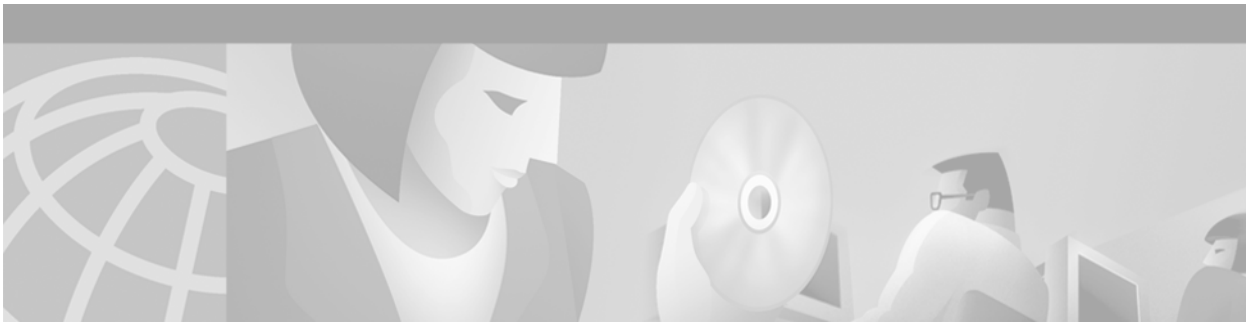


Configuring Session Initiation Protocol for Voice over IP

This chapter introduces the Session Initiation Protocol (SIP). 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 functionality enables Cisco access platforms to signal the setup of voice and multimedia calls over IP networks. The SIP feature also provides nonproprietary advantages in the following areas:

- Protocol extensibility
- System scalability
- Personal mobility services
- Interoperability with different vendors

This chapter contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “Identifying Supported Platforms” in the “Using Cisco IOS Software” chapter.

SIP Overview

SIP is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints.

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

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—Through Session Description Protocol (SDP), SIP determines the lowest level of common services between the endpoints. 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 means 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.

Components of SIP

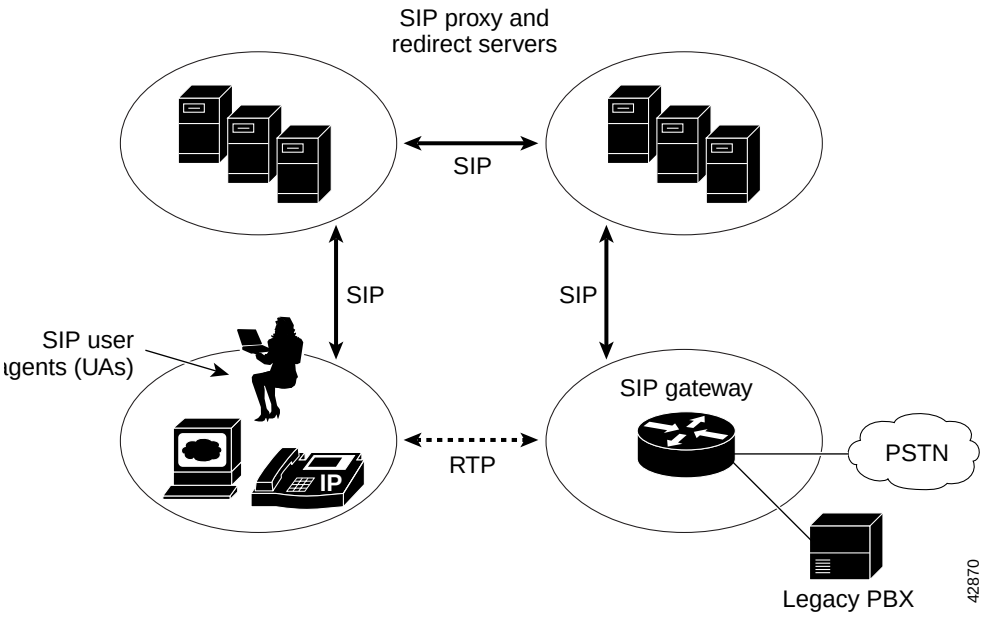
SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs). A user agent can function in one of the following roles:

- 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 UA that initiated the request.

From an architectural standpoint, the physical components of a SIP network can be grouped into two categories: clients and servers. Figure 68 illustrates the architecture of a SIP network.

Figure 68 SIP Architecture



Note 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.

SIP Clients

SIP clients include the following:

- Phones—Can act as either a UAS or UAC. SoftPhones (PCs that have phone capabilities installed) and Cisco SIP IP phones can initiate SIP requests and respond to requests.
- 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

SIP servers include the following:

- Proxy server—Receives SIP messages and forwards them to the next SIP server in the network. The proxy server is an intermediate device that receives SIP requests from a client and then forwards the requests on behalf of the client. 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. The client then contacts the next hop server or UAS directly.
- Registrar server—Processes requests from UACs for registration of their current location. Registrar servers are often located near 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 a network and ultimately to 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 username or an E.164 address.

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

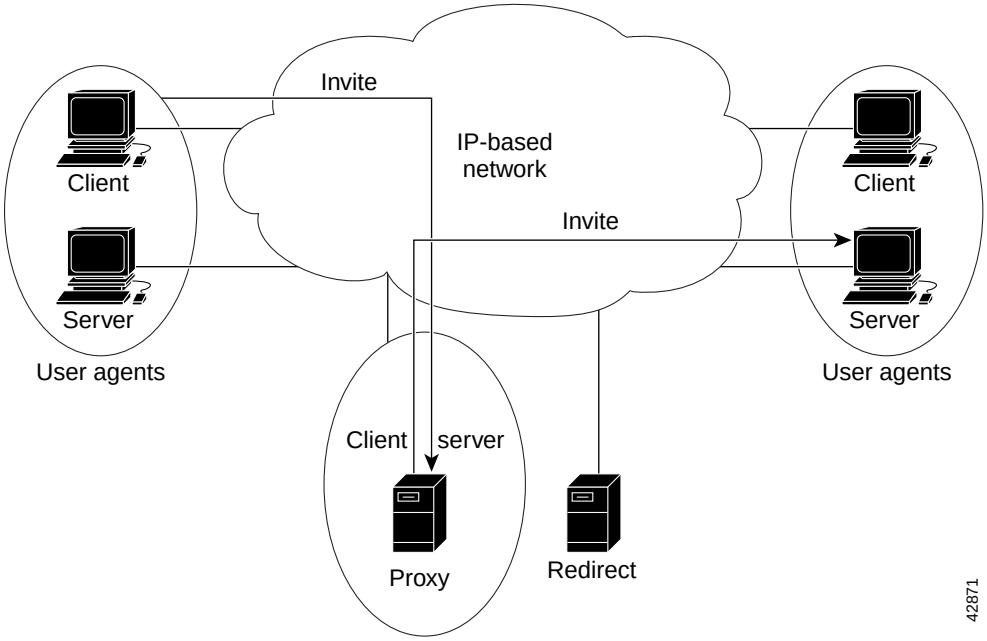
When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller (in the “from” header field) and the address of the intended callee (in the “to” header field). The following sections provide simple examples of successful point-to-point calls established using a proxy and a redirect server.

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, the SIP server might return more than one address for the end user. If the request is coming through a SIP proxy server, the proxy server will try 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.

Using a Proxy Server

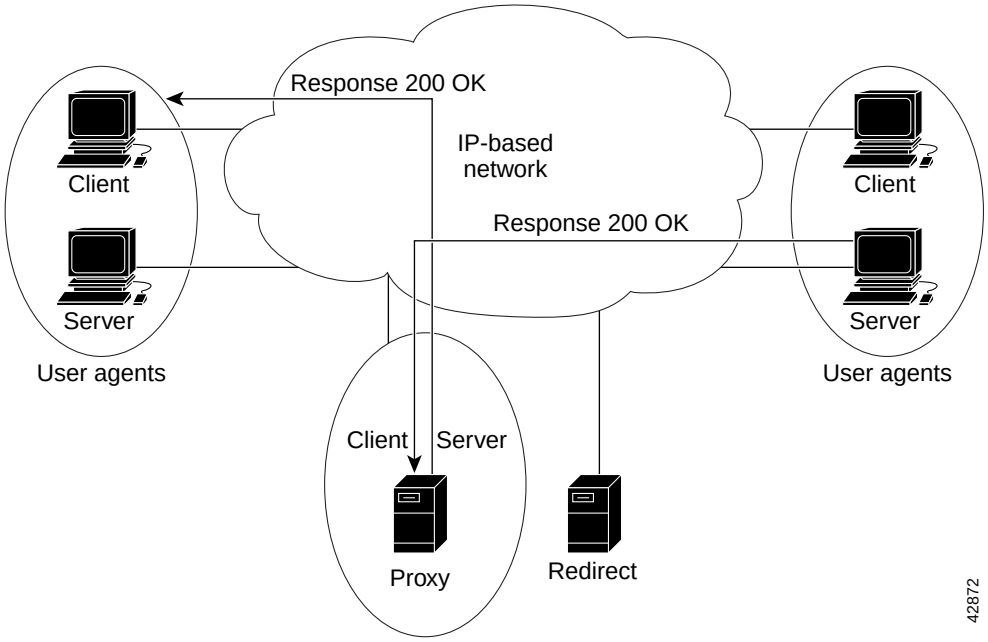
If a proxy server is used, the caller UA sends an INVITE request to the proxy server. The proxy server determines the path and then forwards the request to the callee, as shown in Figure 69.

Figure 69 SIP Request Through a Proxy Server



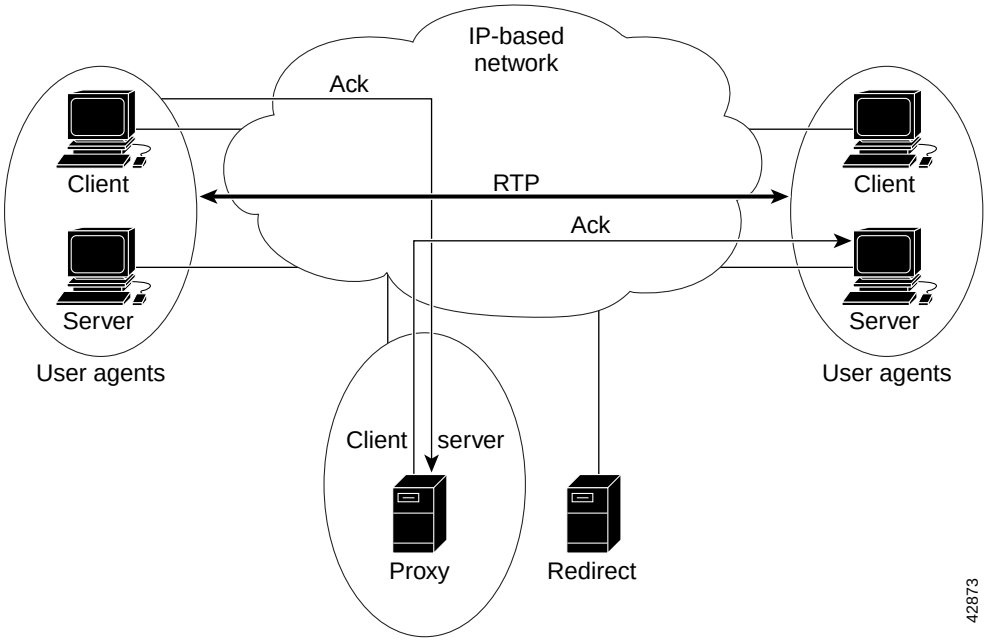
The callee responds to the proxy server, which in turn forwards the response to the caller, as shown in Figure 70.

Figure 70 SIP Response Through a Proxy Server



The proxy server forwards the acknowledgments of both parties. A session is then established between the caller and callee. Real-Time Transfer Protocol (RTP) is used for the communication between the caller and the callee, as shown in Figure 71.

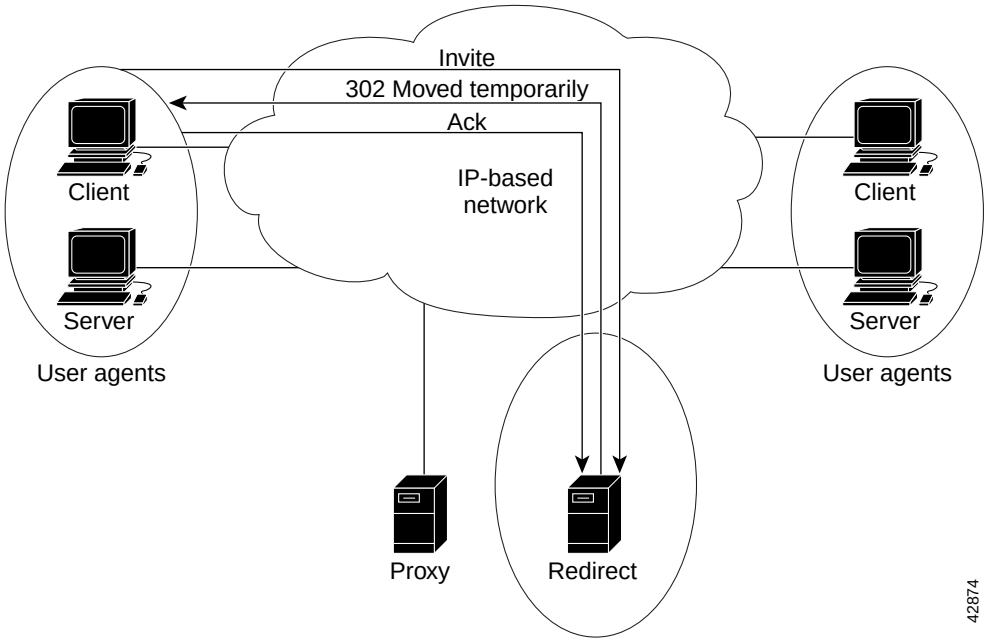
Figure 71 SIP Session Through a Proxy Server



Using a Redirect Server

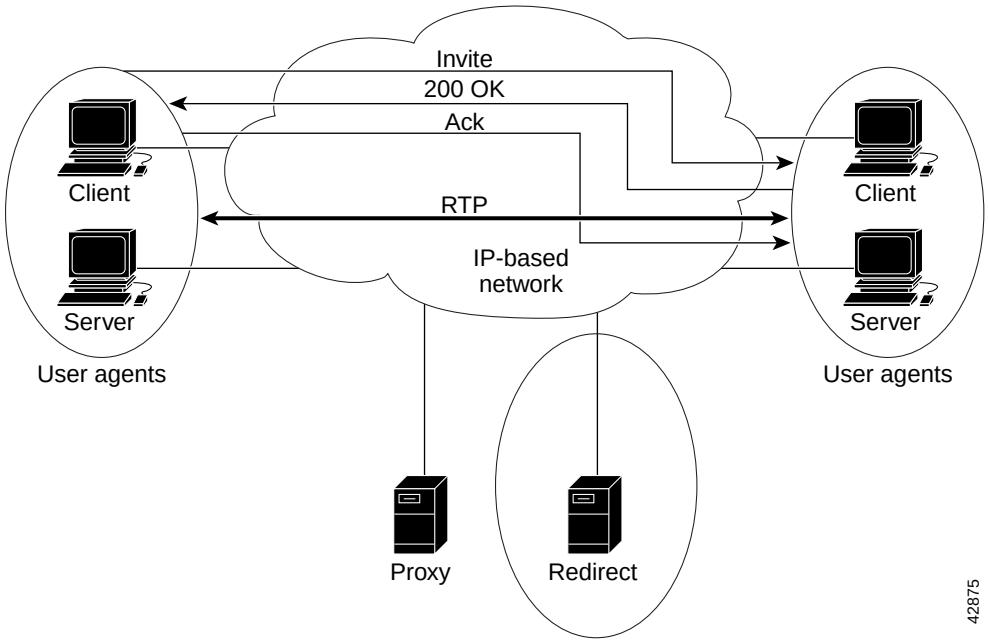
If a redirect server is used, the caller UA sends an INVITE request to the redirect server. The redirect server contacts the location server to determine the path to the callee, and the redirect server sends that information back to the caller. The caller then acknowledges receipt of the information, as shown in Figure 72.

Figure 72 SIP Request Through a Redirect Server



The caller then sends a request to the device indicated in the redirection information (which could be the callee or another server that will forward the request). Once the request reaches the callee, it sends back a response, and the caller acknowledges the response. RTP is used for the communication between the caller and the callee, as shown in Figure 73.

Figure 73 SIP Session Through a Redirect Server



SIP Enhancements

SIP provides the following feature enhancements:

- Ability to specify the maximum number of SIP redirects.
- Ability to specify SIP or H.323 on a dial-peer basis.
- Configurable SIP message timers and retries.
- Interoperability with unified call services (UCS).
- Support for a variety of signaling protocols, including ISDN, PRI, and channel associated signaling (CAS).
- Support for a variety of interfaces, including
  - Analog interfaces: Foreign Exchange Station (FXS)/Foreign Exchange Office (FXO)/recEive and transMit (E&M) analog interfaces.
  - Digital interfaces: T1 CAS, T1 PRI, E1 CAS, E1 PRI, and E1 R2
- Support for SIP redirection messages and interaction with SIP proxies. The gateway can redirect an unanswered call to another SIP gateway or SIP-enabled IP phone. In addition, the gateway supports proxy-routed calls.
- Interoperability with DNS servers, including support for DNS SRV and “A” records to look up SIP URLs according to RFC2052 formatting.
- Support for SIP over TCP and User Datagram Protocol (UDP).
- Support RTP/RTCP for media transport in VoIP networks.
- Support for the following codecs:
  - G711ulaw—0
  - G711alaw—8
  - G723r63—4
  - G726r32—2
  - G728—15
  - G729r8—18
- Support for record-route headers.
- Support for IP quality of service (QoS) and IP precedence.
- Support for IP Security (IPSec) for SIP signaling messages.
- Authentication, authorization, and accounting (AAA) support. For accounting, the gateway device generates call data record (CDR) accounting records for export. For authentication, the SIP gateway sends validation requests to the AAA server. For authorization, the existing access lists are used.
- Support for call hold and call transfer features. The call hold sends a midcall INVITE message, which requests that the remote endpoint stop sending media streams. The call transfer is done without consultation (blind transfer). The transfer can be initiated by a remote SIP endpoint.
- Support for configurable expiration time for SIP INVITEs and maximum number of proxies or redirect servers that can forward a SIP request.
- Ability to hide the identity of the calling party by setting the ISDN presentation indicator.

SIP Restrictions and Considerations

Before configuring your router (Cisco 2600, Cisco 3600, or Cisco AS5300) with the SIP feature, you should note the following restrictions and considerations:

- The SIP gateway does not support codecs other than those listed in the section, "SIP Enhancements."
- SIP requires that all times be sent in Greenwich Mean Time (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.
- With call transfer, 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, 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, will not be 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.
- 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.
- 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.

SIP Prerequisite Tasks

Before you configure your router with the SIP feature, you must perform the following tasks:

- Configure your gateway to support voice functionality for SIP or H.323.
- Establish a working IP network.

  For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*.
- Configure VoIP.
- Ensure that your Cisco 2600 or Cisco 3600 series router has16 MB Flash and 64 MB DRAM memory, minimum. A Cisco AS5300 must have 16 MB Flash and 64 MB DRAM memory, minimum.

SIP Configuration Tasks List

To configure SIP functions on the Cisco AS5300, Cisco 2600, or the Cisco 3600 series router, perform the following tasks:

- Configuring SIP Support for VoIP Dial Peers, page 376
- Changing the Configuration of the SIP User Agent, page 377 (Optional)
- Configuring SIP Call Transfer, page 378 (Optional)
- Configuring Gateway Accounting, page 379 (Optional)

For more information on SIP configuration, including call flows, refer to the document *Session Initiation Protocol Gateway Call Flows, Version 2* in Cisco IOS Release 12.1(3)T found on Cisco.com.

Configuring SIP Support for VoIP Dial Peers

To configure SIP support for a VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters dial-peer configuration mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# session transport {udp | tcp}`	Enters the session transport type for the SIP user agent. The default is **udp**. The transport protocol (**udp** or **tcp**) specified with the **session transport** command must be identical to the protocol specified with the **transport** command.
Step 3	`Router(config-dial-peer)# session protocol {cisco | sipv2}`	Enters the session protocol type. The keywords are as follows: • **cisco**—Configures the dial peer to use proprietary CiscoVoIP session protocol. • **sipv2**—Configures the dial peer to use IETF SIP. SIP users should use this option.

	Command	Purpose
Step 4	`Router(config-sip-ua)# sip-server {dns:[hostname] | ipv4:ip_addr:[port-num]}`	Enters the host name or IP address of the SIP server interface. If you use this command, you can then specify **session target sip-server** for each dial peer instead of repeatedly entering the SIP server interface address for each dial peer. The keywords and arguments are as follows: • **dns:***hostname*—Sets the global SIP server interface to a domain name server (DNS) host name. A valid DNS host name takes the following format: *name.gateway . xyz.* • **ipv4:***ip_addr:*—Sets the IP address. • *portnum* (Optional)—Sets the UDP port number for the SIP server.
Step 5	`Router(config-dial-peer)# session target {sip-server | dns:[$s$. | $d$. | $e$. | $u$. [hostname]| ipv4:ip_addr:[port-num]}`	Specifies a network-specific address for a dial peer. The keywords and arguments are as follows: • **sip-server**— Sets the session target to the global SIP server. Used when the sip-server command has already specified the host name or IP address of the SIP server interface. • **dns:***hostname*—Sets the global SIP server interface to a domain name server (DNS) host name. A valid DNS host name takes the following format: *name.gateway . xyz.* • **ipv4:***ip_addr:*—Sets the IP address. • *portnum*—(Optional) Sets the UDP port number for the SIP server. **Note** Wildcards can be used when defining the session target for VoIP peers.

Changing the Configuration of the SIP User Agent

It is not necessary to configure a SIP user agent (UA) in order to place a call. A SIP UA is configured to listen by default. However, if you want to adjust any of the settings, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# sip-ua`	Enters the SIP user agent (sip-ua) configuration mode to configure SIP-UA related commands.
Step 2	`Router(config-sip-ua)# transport {udp | tcp}`	Configures the SIP user agent (sip-ua) for SIP signaling messages. The default is **udp**. The transport protocol (**udp** or **tcp**) specified with the **session transport** command must be identical to the protocol specified with the **transport** command.

	Command	Purpose
Step 3	`Router(config-sip-ua)# timers {trying number | connect number | disconnect number | expires number}`	(Optional) Configures the SIP signaling timers. The keywords are as follows: • **trying**—Sets the time to wait for a 100 response to an INVITE request. The default is 500. • **connect**—Sets the time to wait for a 200 response to an ACK request. The default is 500. • **disconnect**—Sets the time to wait for a 200 response to a BYE request. The default is 500. • **expires**—Limits the time duration (in milliseconds) for which an INVITE is valid. The default is 180000.
Step 4	`Router(config-sip-ua)# retry {invite number | response number | bye number | cancel number}`	(Optional) Configures the SIP signaling timers for retry attempts. The keywords are as follows: • **invite**—Number of INVITE retries. The default is 6. • **response**—Number of RESPONSE retries. The default is 6. • **bye**—Number of BYE retries. The default is 10. • **cancel**—Number of Cancel retries. The default is 10.
Step 5	`Router(config-sip-ua)# max-forwards number`	(Optional) Limits the number of proxy or redirect servers that can forward a request. The default is 6.
Step 6	`Router(config-sip-ua)# max-redirects number`	(Optional) Sets the maximum number of redirect servers. The default is 1.
Step 7	`Router(config-sip-ua)# default {max-forwards | retry {invite | response | bye | cancel} | sip-server | timers {trying | connect | disconnect | expires} | transport}`	(Optional) Resets the value of a SIP user agent command to its default.

Configuring SIP Call Transfer

To configure SIP call transfer for a POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number pots`	Enters dial-peer configuration mode to configure a POTS dial peer.
Step 2	`Router(config-dial-peer)# application session`	Specifies that the standard session application will be invoked for this dial peer.
Step 3	`Router(config-dial-peer)# destination-pattern pattern`	Specifies the telephone number associated with the dial peer.
Step 4	`Router(config-dial-peer)# port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	(Cisco 2600 and Cisco 3600 series routers) Specifies the local voice port through which incoming VoIP calls will be received.
Step 5	`Router(config-dial-peer)# port {controller number:D}`	(Cisco AS5300 universal access server) Specifies the local voice port through which incoming VoIP calls will be received.

To configure SIP call transfer for a VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* **voip**	Enters the dial-peer mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# application session`	Specifies that the standard session application will be invoked for this dial peer.
Step 3	`Router(config-dial-peer)# destination-pattern` *pattern*	Specifies the telephone number associated with the dial peer.
Step 4	`Router(config-dial-peer)# session target ipv4:`*x.x.x.x*	Specifies the IP address of the destination gateway for outbound dial peers.

Note For information about the commands used to configure translation rules, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

Configuring Gateway Accounting

There are three keywords that configure gateway accounting for SIP:

- The **voip** keyword sends the call data record (CDR) to the RADIUS server. Use this keyword with the SIP feature.
- The **H323** keyword sends the call data record (CDR) to the RADIUS server.
- The **syslog** keyword uses the system logging facility to record the CDRs.

To enable gateway-specific accounting for SIP, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gw-accounting {voip | syslog | h323 [syslog]}`	(Optional) Enables gateway-specific accounting in global configuration mode.

For general accounting information, refer to the *Cisco IOS Security Configuration Guide*.

Verifying SIP Configuration

Enter the **show running-config** command to verify your configuration, or use the **show sip-ua** command to verify the SIP configurations.

The following example shows sample output for the **show sip-ua statistics** command:

```
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
    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
    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,
      NoExistAnywhere 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

Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0
```

The following example shows sample output for the **show sip-ua status** command:

```
Router# show sip-ua status

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

The following example shows sample output for the **show sip-ua timers** command:

```
Router# show sip-ua timers

SIP UA Timer Values (millisecs)
trying 500, expires 180000, connect 500, disconnect 500
```

SIP Configuration Examples

The following shows a basic SIP configuration. This output was created by using the **show running-config** command.

```
Router1# show running-config

Building configuration...

Current configuration:
!
version 12.2
service timestamps debug datetime
service timestamps log uptime
no service password-encryption
!
hostname router1
!
!
!
clock timezone GMT 5
voice-card 1
!
ip subnet-zero
ip tcp path-mtu-discovery
ip name-server 172.18.192.48
!
isdn voice-call-failure 0
!
!
controller T1 1/0
 framing esf
 clock source line primary
 linecode b8zs
!
controller T1 1/1
!
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice class codec 1
 codec preference 1 g711alaw
 codec preference 2 g723r63
 codec preference 3 g723r53
!
!
dial-peer voice 100 pots
 destination-pattern 3660110
 port 2/0/0
!
dial-peer voice 200 pots
 application session
 destination-pattern 3660120
 port 2/0/1
!
dial-peer voice 101 voip
 destination-pattern 3660210
 session protocol sipv2
 session target ipv4:172.16.244.73
 codec g711ulaw
!
```

```
dial-peer voice 201 voip
 application session
 destination-pattern 3660220
 session protocol sipv2
 session target dns:3660-2.sip.com
 codec g711ulaw
!
dial-peer voice 999 voip
 destination-pattern 5551111
 session protocol sipv2
 session target ipv4:172.20.53.89
 session transport tcp
!
dial-peer voice 300 pots
 destination-pattern 2101100
!
dial-peer voice 350 voip
 destination-pattern 3100607
 session protocol sipv2
 session target ipv4:172.18.192.197
 codec g711ulaw
!
dial-peer voice 301 voip
 application session
 destination-pattern 1234
 session protocol sipv2
 session target ipv4:172.18.192.193
 codec g711ulaw
!
dial-peer voice 333 voip
 application session
 destination-pattern 1235
 session protocol sipv2
 session target ipv4:172.18.192.199
 codec g711ulaw
!
dial-peer voice 888 voip
 destination-pattern 888
 session protocol sipv2
 session target ipv4:172.20.53.89
 session transport tcp
 codec g711ulaw
!
dial-peer voice 260011 voip
 destination-pattern 260011
 session protocol sipv2
 session target ipv4:172.18.192.164
 codec g711ulaw
!
dial-peer voice 444 voip
 destination-pattern 2339000
 session protocol sipv2
 session target ipv4:172.18.192.205
 codec g711ulaw
!
dial-peer voice 111 voip
 destination-pattern 111
 session protocol sipv2
 session target sip-server
 codec g711ulaw
!
dial-peer voice 7777777 voip
 destination-pattern 19197777777
 session protocol sipv2
```

```
 session target ipv4:172.18.192.38
 codec g711ulaw
!
!
sip-ua
retry invite 2
retry response 2
retry bye 2
retry cancel 2
no inband-alerting
sip-server dns:server
!
!
interface FastEthernet0/0
 ip address 172.18.192.194 255.255.255.0
 load-interval 30
 speed auto
 half-duplex
!
interface FastEthernet0/1
 ip address 172.16.245.230 255.255.255.224
 load-interval 30
 speed auto
 half-duplex
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.192.1
ip route 172.16.0.0 255.255.0.0 172.16.245.225
no ip http server
!
access-list 101 permit ip host 10.0.2.30 host 10.0.2.31
access-list 101 deny   udp any eq rip any
access-list 101 deny   udp any any eq rip
access-list 101 deny   udp any eq isakmp any
access-list 101 deny   udp any any eq isakmp
access-list 101 permit ip any any
snmp-server engineID local 000000090200003094202740
snmp-server community public RW
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password xxx
 login
!
end
```