Cisco SIP Proxy Server Administrator Guide

Version 2.2

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

Customer Order Number:
Text Part Number:

Cisco SIP Proxy Server Administrator Guide

CONTENTS

INDEX

Preface

This guide describes how to configure a Cisco SIP proxy server (Cisco SPS) version 2.2 to be operational in a VoIP network. It also includes reference information related to compliance and call flows.

Note Use this document in conjunction with the following:

- *Cisco SIP Proxy Server Version 2.2 Installation Guide*, which describes how to install Cisco SPS
- *Release Notes for the Cisco SIP Proxy Server (SPS) Version 2.2*, which presents release information, open and resolved caveats, service and support information, and more

Preface Content

This preface contains the following:

CD Content

The Cisco SPS CD contains the following:

- Linux RPM Package Manager (rpm) software
- Solaris pkg software
- Conversion script—To upgrade from Cisco SPS version 2.x to version 2.2.
- GUI-installer executables (Linux, Solaris, and Windows versions)
- *Cisco SIP Proxy Server Version 2.2 Administrator Guide*—Describes how to configure, monitor, maintain, and troubleshoot Cisco SPS.
- *Cisco SIP Proxy Server Version 2.2 Installation Guide*—Describes how to install Cisco SPS.

- *Release Notes for the Cisco SIP Proxy Server (SPS) Version 2.2*—Presents system and memory requirements; hardware, software, and firmware release information; new and changed system information; installation notes; limitations and restrictions; open and resolved caveats; troubleshooting information; and service and support information. Offers configuration tips.
- *Cisco SPS XML Interface Specification*—Describes the HTTP server/client interface that carries and parses XML data.
- *Cisco SPS RADIUS Interface Specification*—Describes the RADIUS client interface that may be used for accounting and authentication.

Note You can also obtain the guides and release notes, including any updates, at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/index.htm or http://www.cisco.com/en/US/products/sw/voicesw/ps2157/index.html.

Document Content

The guide contains the following (Table 1).

Table 1 Document Organization

Chapter or Appendix	Content
Chapter 1, "Cisco SPS 2.2 Overview"	Overview of SIP; features of and prerequisites for using Cisco SPS
Chapter 2, "Configuring Cisco SPS"	How to use the provisioning GUI to configure the Cisco SPS
Chapter 3, "Operating and Maintaining Cisco SPS"	How to start and stop the Cisco SPS, work with log files, and back up and restore configuration data
Chapter 4, "Monitoring System Status"	How to use CIAgent
Chapter 5, "Troubleshooting"	Symptoms and error messages, along with possible causes and recommended actions
Appendix A, "SIP Compliance"	How the Cisco SPS complies with the IETF definition of SIP as described in RFC 2543; an overview of SIP concepts and services
Appendix B, "Manual Configuration"	How to manually edit text-based configuration files if the provisioning-system GUI is not used
Appendix C, "Manual Operation and Maintenance"	How to manually start and stop the system and manage the registry, routing, and MySQL databases if the provisioning-system GUI is not used
Appendix D, "DNS Setup"	How to set up DNS processes
Appendix E, "SIP Call-Flow Scenarios"	How SIP messages are exchanged during various call scenarios
Glossary	
Index	

Audience and Objectives

Network engineers, system administrators, and telecommunication engineers should use this guide to learn how to configure a Cisco SPS on the network.

Cisco SPS configuration tasks are considered to be administration-level tasks and require a working knowledge of UNIX, including configuration of user shells. They also require an understanding of IP networking and telephony concepts.

Note This guide does not provide sufficient information for you to fully implement a SIP VoIP network.

Typographic Conventions

Table 2 describes conventions that are used in this document.

Table 2 Document Conventions

Convention	Description
boldface	Commands and keywords.
italic	Command input that you supply.
[ ]	Keywords or arguments that appear within square brackets are optional.
{ x | x | x }	Alternative, mutually exclusive keywords are grouped in braces and separated by vertical bars.
^ or Ctrl	The key labeled *Control*. For example, when you read ^*D* or *Ctrl-D*, you should hold down the Control key while you press the D key.
`screen font`	Examples of information displayed on the screen.
boldface screen font	Examples of information that you must enter.
< >	Nonprinting characters, such as passwords, appear in angle brackets.
[ ]	Default responses to system prompts appear in square brackets.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/techsupport

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

Cisco documentation and additional literature are available in the Product Documentation DVD package, which may have shipped with your product. The Product Documentation DVD is updated regularly and may be more current than printed documentation.

The Product Documentation DVD is a comprehensive library of technical product documentation on portable media. The DVD enables you to access multiple versions of hardware and software installation, configuration, and command guides for Cisco products and to view technical documentation in HTML. With the DVD, you have access to the same documentation that is found on the Cisco website without being connected to the Internet. Certain products also have .pdf versions of the documentation available.

The Product Documentation DVD is available as a single unit or as a subscription. Registered Cisco.com users (Cisco direct customers) can order a Product Documentation DVD (product number DOC-DOCDVD=) from Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Ordering Documentation

Beginning June 30, 2005, registered Cisco.com users may order Cisco documentation at the Product Documentation Store in the Cisco Marketplace at this URL:

http://www.cisco.com/go/marketplace/

Nonregistered Cisco.com users can order technical documentation from 8:00 a.m. to 5:00 p.m. (0800 to 1700) PDT by calling 1 866 463-3487 in the United States and Canada, or elsewhere by calling 011 408 519-5055. You can also order documentation by e-mail at tech-doc-store-mkpl@external.cisco.com or by fax at 1 408 519-5001 in the United States and Canada, or elsewhere at 011 408 519-5001.

Documentation Feedback

You can rate and provide feedback about Cisco technical documents by completing the online feedback form that appears with the technical documents on Cisco.com.

You can send comments about Cisco documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com

  An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.
- Nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.*x* through 8.*x*.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Technical Support & Documentation website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Technical Support & Documentation Website

The Cisco Technical Support & Documentation website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support & Documentation website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

- Severity 1 (S1)—Your network is "down," or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.
- Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.
- Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.
- Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

  or view the digital edition at this URL:

  http://ciscoiq.texterity.com/ciscoiq/sample/

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- Networking products offered by Cisco Systems, as well as customer support services, can be obtained at this URL:

  http://www.cisco.com/en/US/products/index.html

- Networking Professionals Connection is an interactive website for networking professionals to share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:

  http://www.cisco.com/discuss/networking

- World-class networking training is available from Cisco. You can view current offerings at this URL:

  http://www.cisco.com/en/US/learning/index.html

CHAPTER 1

Cisco SPS 2.2 Overview

The typical Cisco SPS user is a system administrator who uses local interfaces to configure, provision, monitor, and control SIP proxy-server operation, preferably through the server GUI interface or, alternatively, through static configuration files. This chapter provides the background information that such an administrator requires before performing such tasks.

The Cisco SIP proxy server (Cisco SPS) is a call-control software package that enables service providers and others to build scalable, reliable packet voice networks and to provide call-session management in a VoIP network. It has can also serve as a registrar or redirect server. It provides a full array of call-routing capabilities for maximizing network performance in both small and large packet voice networks.

Cisco SPS has the capabilities of an edge proxy server, performing such functions as authentication, accounting, registration, network-access control, and security. It can also has the capabilities of an infrastructure proxy server, performing such functions as next-hop routing based on received or translated destination URLs.

New with Cisco SPS 2.2 are the list of supported platforms and operating systems (described in the installation guide) and the addition of caller-privacy enhancements.

Cisco SPS is based on Session Initiation Protocol (SIP), a text-based protocol for setting up, modifying, and tearing down both unicast and multicast multimedia conferences.

This chapter provides an overview of Cisco SPS.

Contents

Prerequisites

Hardware and Software

Hardware and software requirements are described in the *Cisco SIP Proxy Server Version 2.2 Installation Guide*, available on your Cisco SPS CD or at http://www.cisco.com/en/US/products/sw/voicesw/ps2157/prod_technical_documentation.html.

Note As described in the reference above, Cisco SPS runs only on the following operating systems:

- Red Hat Enterprise Linux AS, ES, or WS 3.0
- Sun Solaris 8 Operating Environment

It is critical that you use only a supported operating system. Cisco SPS does not run on other than these.

System Permissions, Strategy, and Functionality

Permissions and Knowledge

You must have the following:

- Administrator privileges
- Familiarity with UNIX commands and shells

Architecture Strategy

You must have determined which one of the following proxy-server architectures to implement:

- One standalone server
- A farm of two servers

Configuration Strategy

You must have determined which one of the following strategies to use to configure SIP directives:

- GUI-based provisioning system, also called the provisioning GUI
- Manual configuration of the SIP directives (sipd.conf) file

Note As a general rule, use GUI rather than manual methods to configure and operate your system.

- GUI methods are described in Chapter 2, "Configuring Cisco SPS" and Chapter 3, "Operating and Maintaining Cisco SPS."
- Manual methods are described in Appendix B, "Manual Configuration" and Appendix C, "Manual Operation and Maintenance."

Restrictions

Hardware and Software

Members of a proxy-server farm may run on different hardware. However, they must run the same operating system (mixed Linux/Solaris farms are not supported) and the same version of the Cisco SPS software.

Time Synchronization

Members of a proxy-server farm must be time-synchronized to a common clock.

Network Management

Do not run Network Information System (NIS)—a network-lookup service for managing a network of computers—on Cisco SPS systems. Doing so can cause long delays when you add a new user or user group. If you do run NIS, be sure to note any instructions for stopping it before doing configuration tasks.

Permissions

You must run the proxy servers with either csps (default account ID) or root permission.

Log-File Size

Log-file size is limited to 2 Gb. You must therefore ensure that neither the access_log file nor the error_log file exceeds this size.

SIP Basics

SIP is the Internet Engineering Task Force (IETF) standard for multimedia conferencing over IP. A lightweight, generic, ASCII-based signaling protocol that provides session control, it is complementary to Simple Gateway Control Protocol (SGCP) and Media Gateway Control Protocol (MGCP), both of which provide device control. It offers significant performance advantages over H.323; it also more easily enables applications, provides a simple method to map Signaling System 7 (SS7) to Internet networks, provides loop detection, and supports conferencing.

Like other VoIP protocols, SIP provides signaling and session management within a packet-telephony network. Signaling allows call information to be carried across network boundaries. Session management controls the attributes of an end-to-end call.

SIP does the following, in the stated order:

1. Determines the location of the target endpoint.

   SIP supports address resolution, name mapping, and call redirection.

2. Determines the media capabilities of the target endpoint.

   Via Session Description Protocol (SDP), SIP determines the highest level of common services between the endpoints. Conferences are established through use of only those media capabilities that can be supported by all endpoints.

Note A conference is an established session (or call) between two or more endpoints. Conferences consist of two or more users and can be established through multicast or multiple unicast sessions. In this document, a conference and a call are synonymous.

3. 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 already on the phone or did not answer in the allotted number of rings. It then returns a message indicating why the target endpoint is unavailable.

4. Establishes a session between the originating and target endpoint.

   If a call can be completed, SIP establishes a session between the endpoints. SIP also supports midcall changes such as addition of another endpoint to the conference or changing of a media characteristic or codec.

5. Transfers calls from one endpoint to another and terminates calls.

   During 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 sessions between all parties.

SIP Components

SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs). A user agent can function in either of two roles: user-agent client or user-agent server. Definitions in the following sections are adapted from RFC 3261.

User-Agent Client

A user-agent client (UAC) is a peer that initiates a SIP call request. More formally, it is a logical entity that creates a new request and then uses the client-transaction-state machinery to send it. A UAC typically initiates a call through a proxy server such as Cisco SPS, and relies on the server to locate the desired recipient and obtain requested or required network services.

The role of UAC lasts only for the duration of that transaction. In other words, if a piece of software initiates a request, it acts as a UAC for the duration of that transaction. If it receives a request later, it assumes the role of a user-agent server for the processing of that transaction.

Clients are of the following types:

- Phones that can act as either UAS or UAC. Softphones (PCs with installed phone capabilities) and Cisco SIP IP phones can initiate and respond to requests.
- Gateways that translate transmission format, communications procedures, and codecs between SIP conferencing endpoints and other terminal types. Other call-control functions include call setup and clearing on both the LAN and the switched-circuit network side.

The UAC core is the set of processing functions that are required of a UAC and that reside above the transaction and transport layers.

User-Agent Server

A user-agent server (UAS) is a peer that receives and responds to call requests. More formally, it is a logical entity that generates a response to a SIP request; the response is to accept, reject, or redirect the request. This role lasts only for the duration of that transaction. In other words, if a piece of software responds to a request, it acts as a UAS for the duration of that transaction. If it generates a request later, it assumes the role of a UAC for the processing of that transaction.

A UAS can interact with other applications such as Lightweight Directory Access Protocol (LDAP) servers, databases, and Extensible Markup Language (XML) applications that provide back-end services such a directory, authentication, and billing.

Servers are of the following types:

- Proxy servers
- Redirect servers
- Registrar servers
- Location servers

Proxy Servers

A proxy server initiates requests on behalf of and receives requests from a client. More formally, a proxy server is an intermediate entity that acts as both a server and a client for the purpose of making requests on behalf of other clients. A proxy server primarily handles routing, ensuring that a request is sent to another entity closer to the targeted user. Proxy servers are also useful for enforcing policy (for example, ensuring that a user is allowed to make a call). A proxy server interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.

Proxy servers provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security. They are often colocated with redirect or registrar servers.

Redirect Servers

A redirect server is a UAS that generates 3xx responses to requests that it receives, directing the client to contact an alternate set of URIs. It receives requests, strips out the address in the request, checks its address tables for any other addresses that might be mapped to the one in the request, and returns the results of the address mapping to the client for the next one or more hops that a message should take. The client then contacts the next-hop server or UAS directly. Redirect servers are often colocated with proxy or registrar servers.

Registrar Servers

A registrar server is a server that accepts REGISTER requests from UACs for registration of their current location. It places the information received in those requests into the location service for the domain that it handles. Registrar servers are often colocated with proxy or redirect servers.

Location Services

A location service is used by a SIP redirect or proxy server to obtain information about a called party's possible locations. It contains a list of bindings of address-of-record keys to zero or more contact addresses. Bindings can be created and removed in many ways; the SIP specification defines a REGISTER method for updating bindings. Location services are often colocated with redirect servers.

SIP Network Architecture

A typical SIP endpoint can function as either a UAC or a UAS during a transaction, its role depending on whether it initiates or receives and responds to a request.

A UAC can directly contact a UAS if it knows the location of the UAS and does not want any special services from the network. However, a UAC typically initiates a call through a proxy server and relies on the proxy server to locate the desired UAS and obtain any special services from the network. The SIP messaging path from UAC to UAS can involve multiple proxy servers, and in such scenarios Cisco SPS interfaces at a peer level with other proxy servers.

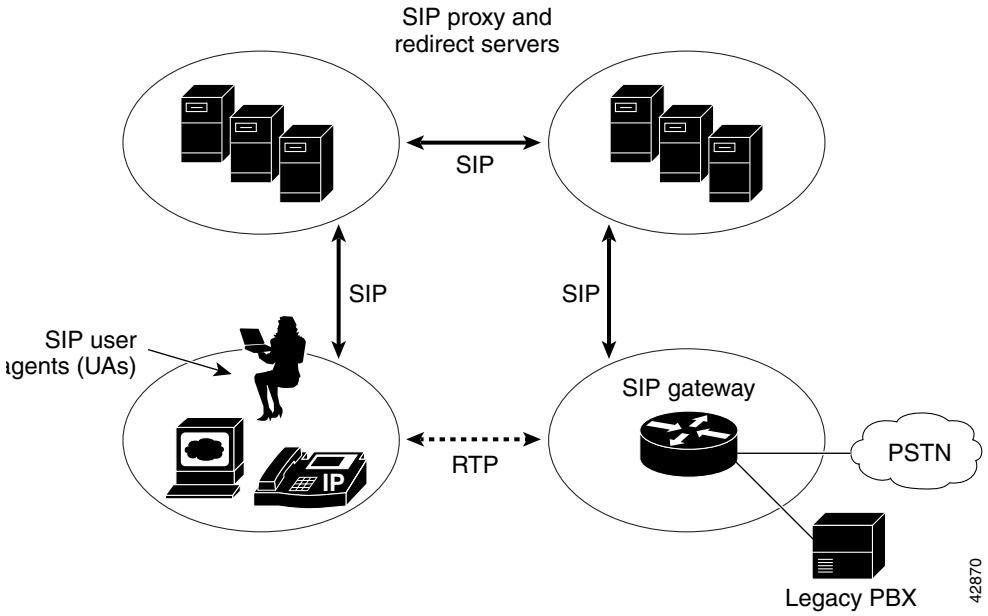
SIP requests can be sent with any reliable or unreliable protocol. Cisco SPS supports the use of the following for sending and receiving SIP requests and responses:

- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- Transport Layer Security (TLS)

Cisco SPS can use any physical-layer interface in the server that supports Internet Protocol (IP).

Figure 1-1 shows the architecture of a SIP network.

Figure 1-1 SIP Network Architecture



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 form 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 provides this information to the location server upon request.

A user initiates a call by sending a SIP request to a SIP server (either a proxy server or a redirect server). The request includes the address or record (AOR) of the called party in the Request URI.

Over time, a SIP end user—that is, a subscriber—might move between end systems. The location of the subscriber 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 subscriber. Because the subscriber can be logged in at more than one station, the server might return more than one address. If the request comes through a SIP proxy server, the proxy server tries each of the returned addresses until it locates the subscriber. If the request comes through a SIP redirect server, the redirect server forwards all of the addresses to the caller listed in the Contact header field of the invitation response.

Cisco SPS Basics

Cisco SPS sits in the core of a SIP network and routes calls among other proxy servers, voice gateways, IP endpoints (such as IP phones), and application servers. It enables user registration, authentication, and call-routing decisions to be made within the network, and identifies the next hop in the path to the called party. During call setup and teardown, Cisco SPS can generate accounting messages and pass them to a RADIUS server to form call-detail records (CDRs).

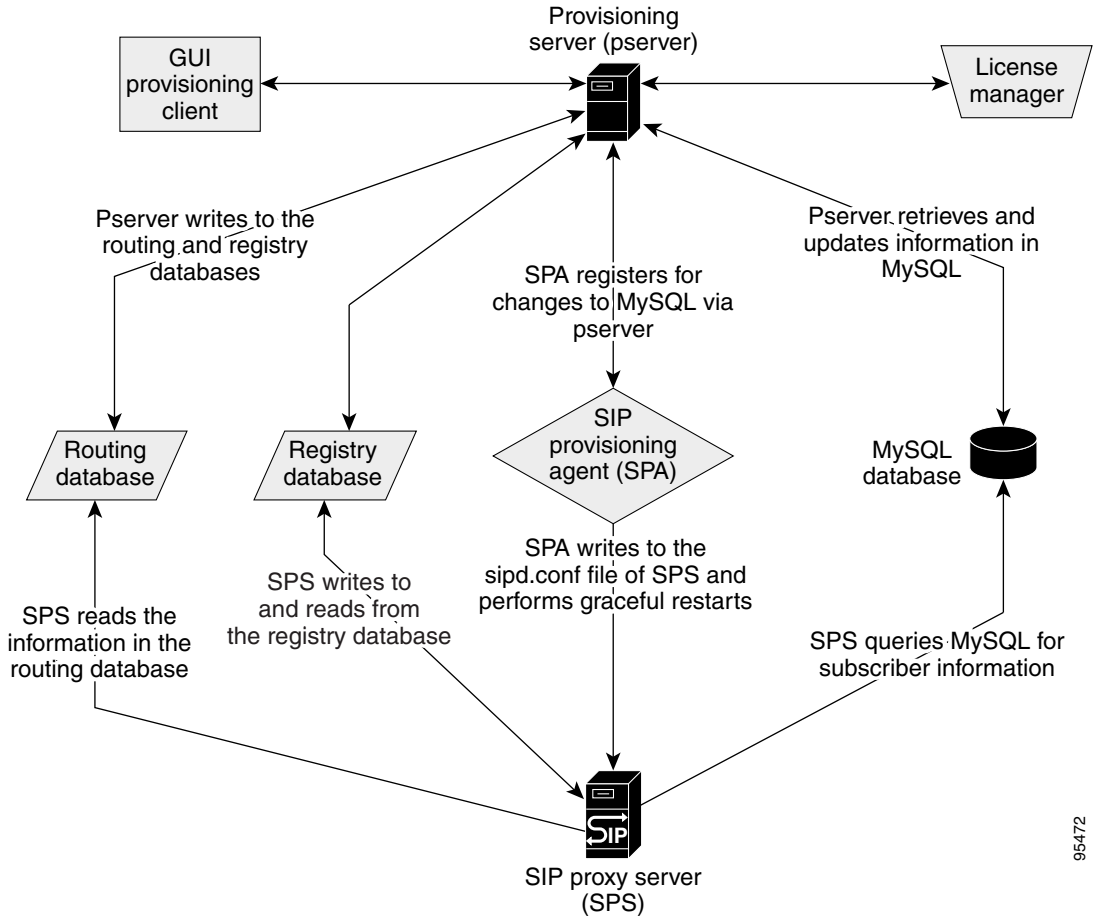
Components

Cisco SPS components (see Figure 1-2) include the following:

- Proxy server
- Provisioning server
- MySQL database
- Provisioning GUI client of the Cisco SPS GUI-based provisioning system
- SIP provisioning agent
- Registry database
- Routing database
- License manager

Each of these elements is described after Figure 1-2.

Figure 1-2 Cisco SPS Components



Proxy Server (sipd)

The proxy server provides the primary capabilities required for call-session management in a VoIP network. It processes SIP requests and responses. It can be configured to function as a transaction stateful or stateless server and to provide additional server modes and features.

Cisco SPS supports the use of a proxy-server farm containing up to two proxy servers.

Provisioning Server (pserver)

The provisioning server is used by the Cisco SPS GUI-based provisioning system. A license manager is automatically installed when the provisioning server is installed.

MySQL Database

MySQL is a popular open-source database whose architecture makes it extremely fast and easy to customize. This database stores and accesses provisioning-system and subscriber-feature data. Subscriber features (call forwarding and local authentication, but not RADIUS) are automatically included.

Cisco SPS supports up to two replicated and synchronized MySQL databases.

Provisioning GUI Client of the Cisco SPS GUI-Based Provisioning System

The provisioning GUI client—also called the GUI-based provisioning system—retrieves and displays current information in the MySQL database by means of the provisioning server. It also retrieves and displays information outside of the MySQL database, such as registrations and routes.

If you modify this information, the GUI sends the updates to the provisioning server. If you modify information that is related to the license, the license manager does additional processing on this information, after which the provisioning server updates the database with the information. Meanwhile, each proxy-server provisioning client (there can be more than one on a farm) has registered to receive specific changes to the database. If a client is notified of a change, it requests that the provisioning server send the changed information and then updates its configuration file and its routing and registry data. The proxy server can use the new registry and routing information immediately, and uses the new configuration file if it is gracefully restarted.

The GUI is currently designed to change only certain things, such as sipd.conf, subscriber data, routes, and registries. It does not change other things such as spa.conf and ps.conf.

You can install the client independently of the provisioning server.

SIP Provisioning Agent (spa)

The SIP provisioning agent resides on a farm member and handles requests that the provisioning server gets from the GUI. It receives requests from the provisioning server, accesses and updates (as needed) the SIP directives (sipd.conf) file, and provides feedback, by way of the provisioning server, to the GUI.

The IP address in the spa (spa.conf) file is needed so that the spa process can determine whether the data is intended for this particular process. The spa process makes the determination by comparing the data with the IP address in spa.conf; if they do not match, it decides that the data is not intended for this process.

Changes to the MySQL database that the SIP provisioning agent does not register for include subscriber information such as call-forwarding destinations and authentication passwords. These changes pass from the database directly to sipd.

Registry Database

The registry database contains location information for registered endpoints. The database is stored in memory-mapped files within shared memory so that information persists between restarts. Registry information is exchanged with all members of a proxy-server farm. The database contains two types of information: dynamic information that is received from endpoints when they register their contact information and static information that is configured by means of the provisioning GUI client.

Information pertaining to a single registered user (SIP endpoint) is called a registration. The registry database is thus a collection of registrations.

Registration is made or renewed either dynamically or statically:

- Dynamic registration—An endpoint registers itself by sending a REGISTER request to a Cisco SPS, which then adds a registration to the registry. By default, SIP endpoints register once every hour, and each previous registration expires after an hour. Dynamic registration is the norm.
- Static registration—You, as system administrator, explicitly create a registration for a SIP endpoint that is incapable of registering for itself (such as a SIP gateway). Static registrations are generally permanent. They are not the norm, because such endpoints are generally represented by a route instead of a registration.

Routing Database

The routing database contains static route information that the proxy server uses to forward requests toward endpoints that are either not registered with the local registrar server, reside within a different domain, or exist in the PSTN. Static routes are configured based on next-hop IP addresses or next-hop domains. Routing information is configured by means of the provisioning GUI client. As with the registry database, the routing database is stored in memory-mapped files within shared memory so that information persists between restarts.

License Manager

The license manager maintains the license key required for activation of the proxy server. You need a valid license to install, start, and restart SPS. Two types of licenses are available: evaluation and permanent. An evaluation license expires after a set period of time; after the expiration date, the proxy server can only be gracefully restarted.

Server Modes

Cisco SPS can function as a proxy server, a redirect server, or a registrar server. The first two can be transaction-stateful or transaction-stateless.

- A transaction-stateful server remembers incoming and outgoing requests, provides reliable retransmission of proxied requests, and returns the best final responses.
- A transaction-stateless server forgets all information once a request or response has been processed. It merely forwards requests and responses.

A transaction includes the following:

- Received request
- Request or requests (if forked) forwarded downstream
- Responses received from downstream hosts
- Best response returned upstream

Proxy Server

A proxy server is an intermediate device that receives SIP requests from a client and then initiates requests on the client's behalf. It can be transaction-stateful or transaction-stateless. If transaction-stateful, it also does the following:

- Creates a transaction control block (TCB)
- Remembers incoming and outgoing requests
- Provides reliable retransmissions for unreliable transports
- Returns the best final response or responses upstream

Cisco SPS functions by default as both a proxy server and a registrar server for all calls, although you can change this default functionality.

Redirect Server

A redirect server does the following:

- Accepts SIP requests
- Maps the address in the request-URI to zero or more new addresses
- Returns these addresses as contacts in a SIP 3xx response to the UAC

A redirect server can be transaction-stateful or transaction-stateless. If transaction-stateless, it does not create a TCB on receiving an INVITE request.

Registrar Server

A Cisco SPS functioning in the role of registrar server does the following:

- Processes requests from UACs for registration of their current location
- Maintains registration information, which it can share with other registrar servers in its server farm
- Provides location services to the proxy server

Cisco SPS functions by default as both a proxy server and a registrar server for all calls, although this default functionality can be changed.

You must configure each registrar server to function also as either a proxy server or a redirect server. Standalone registrar servers are not supported.

User IDs and Passwords

As a general rule, use the default user IDs and passwords shown in Table 1-1.

Table 1-1 System User IDs and Passwords

System	Default Account ID	Default Password
Cisco SPS	cspsuser	cspsuser
MySQL	guest	nobody

Features

Cisco SPS supports the following major features and capabilities, listed alphabetically:

- Access and Error Logging, page 1-12
- Accounting, page 1-12
- Address Translation, Next-Hop Routing, and IP Resolution, page 1-13
- Authentication. Authorization, and Access Control Lists, page 1-16
- DNS Support, page 1-17
- IP Security, page 1-18
- Proxy-Server Farms, page 1-18
- Registrar Server for Multiple Domains, page 1-18
- Registry and Route Configurations, page 1-20
- Spiralled and Looped Request Detection, page 1-21
- Subscribers, page 1-21
- TLS Support, page 1-22

Note For information on configuring these features and capabilities, see Chapter 2, "Configuring Cisco SPS."

In addition, Cisco SPS provides the following additional capabilities, some of which were described earlier:

- Functions as a transaction-stateful or transaction-stateless proxy server, transaction-stateful or transaction-stateless redirect server, and registrar server
- Handles call forwarding
- Has a GUI-based provisioning system for configuring the server and accessing the embedded routing and registry databases and the embedded MySQL subscriber database
- Forks requests and distinguishes spiralled requests from looped requests
- Supports SIP over UDP or TCP
- Interoperates with Cisco SIP gateways, SIP IP phones, and unified messaging
- Supports parameters relevant to Network Address Translation (NAT) traversal
- Has a domain-specific registration, authentication, accounting, and subscriber database
- Handles preauthorization queries to a resource-policy-management system
- Offers an SNMP interface by means of CIAgent with basic platform MIBs for server (including MySQL) status and start/stop/restart
- Provides database robustness:
  - Registration and route databases are memory-mapped files that facilitate both quick access and persistence should servers be stopped and restarted.
  - The subscriber database is stored in MySQL. If a proxy-server farm contains multiple MySQL databases, subscriber data is synchronized between them and persists between server stops and restarts.

Access and Error Logging

Cisco SPS uses both standard Apache and SIP-specific logging functionality. Standard Apache logging functionality is configured on Cisco SPS as a whole. SIP-specific logging functionality is configured on a per-module basis.

For information on log locations, interpreting log files, and customizing the type and amount of information that they include, see Chapter 3, “Operating and Maintaining Cisco SPS.”

Accounting

If you enable accounting services and configure the interface to a RADIUS server, Cisco SPS sends accounting records to the RADIUS server.

Cisco SPS uses basic start-stop records with a combination of standard RADIUS attributes and Cisco vendor-specific attributes (VSAs). Additionally, you can configure Cisco SPS to add any desired SIP headers as VSAs in accounting requests.

Cisco SPS receives the BYE message only if record-route is enabled. In an unsuccessful call, it writes a STOP message when the best non-200 (4/5/6XX) response is received for an INVITE message.

Tip For information about Cisco SPS and the RADIUS server, see the following:

- *Cisco SPS RADIUS Interface Specification* at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/index.htm or http://www.cisco.com/en/US/products/sw/voicesw/ps2157/index.html
- *RADIUS VSA Voice Implementation Guide* at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm

Address Translation, Next-Hop Routing, and IP Resolution

Cisco SPS performs the following steps to deliver messages from endpoint to endpoint:

1. Address translation—Translates an incoming request-URI into an outgoing request-URI.
2. Next-hop routing—Obtains a set of fully qualified domain names (FQDNs) or IP addresses with transport type and port numbers for each of the SIP entities found in the translation step. This step involves the following features and more:
   - SRV lookup for static route—SRV lookup on the Static_Route_NextHop field if the Static_Route_NextHopPort field in a static route is not specified or is zero.
   - Registration, Admission, and Status Protocol (RAS) LRQ message transmission to gatekeeper—Sending the LRQ message to the H.323 gatekeeper to obtain the next-hop gateway transport address.
3. IP resolution—Converts each next hop found in the next-hop route lookup step into an IP address.

Note If a route header is present in the SIP request message, Cisco SPS bypasses translation and next-hop routing steps. You must enable record-route on the server for subsequent requests to contain a route header.

Address Translation

During address translation, Cisco SPS processes the request-URI of an incoming request and returns a list of contacts, each providing a URL for use in the outgoing request.

If you enable number expansion, Cisco SPS applies the global set of expansion rules to the user portion of the relevant URLs for which the host portion is the Cisco SPS. For REGISTER messages, this applies to the To, From, Contact, and optionally the Authorization headers. For INVITE messages, this applies to the Request URI, From, and optionally the Proxy-Authorization headers.

Note Headers are not rewritten. The expanded versions are used internally for authentication, accounting, translation, and routing purposes.

Cisco SPS translation modules, in the order in which Cisco SPS calls them, are as follows:

- Call Forward Unconditional (mod_sip_call_forward)
- Registry (mod_sip_registry)
- ENUM (mod_sip_enum)
- GKTMP (mod_sip_gktmp)

The first module to return one or more contacts completes the translation step, and the remaining modules are not called. For example, if the Registry module returns a contact, then neither the ENUM nor the GKTMP module is called. If none of the translation modules returns a contact, the core proxy module (mod_sip) returns a contact based on the incoming Request-URI and that Request-URI is used in the next-hop routing step.

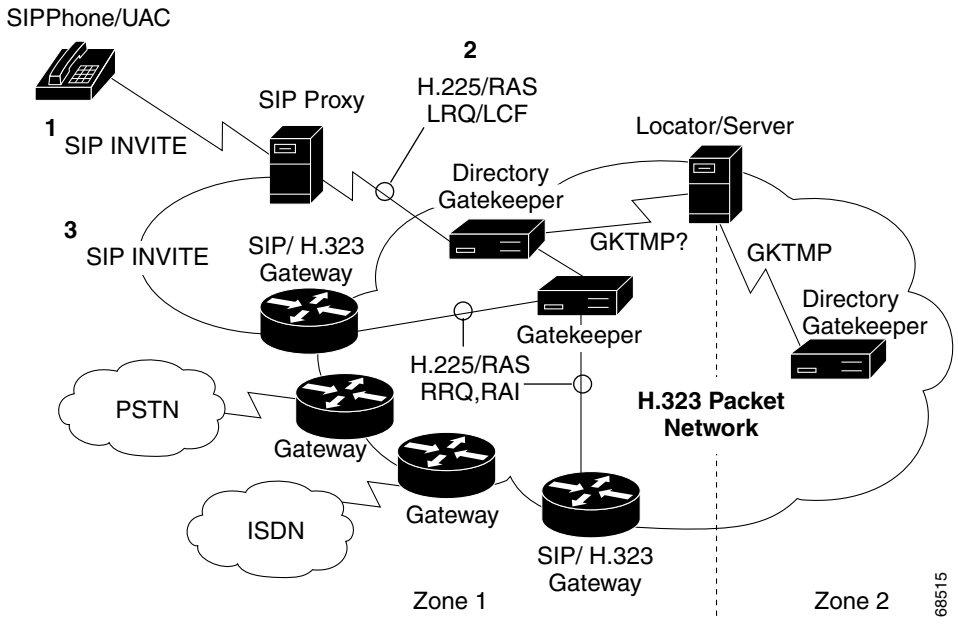
Next-Hop Routing

The next step in SIP request processing is to determine the next-hop route for each contact. Next-hop routing takes each translated request-URI (contact) and locates a set of next-hop SIP entities capable of routing a message to the new request-URI. This step involves two advanced features:

- SRV Lookup for Static Route—If the Static_Route_NextHopPort field of a static route is not specified or is zero, Cisco SPS performs an SRV lookup on the Static_Route_NextHop field.
- H.323 RAS module—The RAS module allows communications between a SIP proxy server and an H.323 gatekeeper. Cisco SPS can send an ASN.1-encoded RAS LRQ message to an H.323 gatekeeper that responds with a RAS LCF message.

Figure 1-3 illustrates how the proxy server sends an LRQ H.225 RAS message to a gatekeeper that responds with an LCF message that can contain a gateway transport address.

Figure 1-3 Cisco SPS and H.323 Gatekeeper Message Processing



The RAS module supports the following:

- H.323 request in progress (RIP) message—This message contains a time-delay value. Upon receipt of this message, Cisco SPS resets its timer and waits for the final response from the gatekeeper.
- Directory gatekeeper—This gatekeeper forwards requests to the other gatekeepers and returns the highest-priority and lowest-cost gateway information to the requesting endpoint. The value of the time-to-live (TTL) field in the LRQ message must be greater than zero. The default is 6 hops.

- Sequential LRQ—Cisco SPS can send LRQ requests sequentially based on the priority specified for the gatekeeper clusters in the sipd.conf file. Configurable parameters include a timeout value for each individual request, a total LRQ request window, and an indication as to whether the module should wait for the best LCF within the LRQ time window or return the first valid LCF response to the proxy server.
- Blast LRQ—Cisco SPS can send LRQ requests to all gatekeeper clusters before listening to the responses from the gatekeepers. It can also be configured to either take the first valid LCF response or wait for the best response within the LRQ time window.
- Multi-alternate endpoint—If the selected LCF from a gatekeeper contains alternate endpoints, Cisco SPS can parse these endpoints and store them in the route table. The routes in the alternate endpoints have lower priority than the route for the primary endpoint.
- Redundant gatekeeper—To establish redundancy. you must maintain a list of prioritized gatekeeper clusters in the Cisco SPS configuration. The following trial process occurs:
  1. The proxy server tries one of the gatekeepers randomly in the cluster that has the highest priority.
  2. If the trial goes beyond the timeout value specified in the RASTimeoutInterval directive, the proxy server tries the next gatekeeper in the same cluster in a round-robin manner.
  3. If the proxy server receives an LRJ message, it tries a gatekeeper in the next-highest-priority cluster and so forth.
- Tech prefix—Cisco SPS prepends the technology prefix (one, two, or three digits followed by #) to the expanded dialed number in the LRQ request. The prefix can also be included in the Request-URI of the INVITE message forwarded to the SIP/H.323 gateway. Multiple technology prefixes can be configured in the sipd.conf file based on the expanded dialed number.
- Pavo extension—Cisco SPS can include Pavo extensions (CallIdentifier, RedirectIEInfo, CallingOctet3a) in the LRQ from the CC-Diversion header of the incoming INVITE message.

During location of the next-hop SIP entities, the following occurs:

1. If the host portion of the new request-URI is the address (FQDN or IP address) of the server itself, next-hop routing is performed by means of the user portion of the request-URI. E.164 routing makes use of this method.
2. If the Static_Route_NextHopPort field of a static route is not specified or is 0, Cisco SPS tries to do an SRV lookup on the Static_Route_NextHop field.
   - If the lookup succeeds, it uses the algorithm outlined in RFC 2782 to select one destination.
   - If the lookup fails, it tries alternate destinations and the proxy server does a simple DNS A lookup on the Static_Route_NextHop field of the static route. This field should contain a name for the A lookup.
3. If no route is found in the E.164 routing, Cisco SPS sends an RAS LRQ message to the H.323 gatekeeper cluster that has the highest-priority gateway. According to configurations and availability of the gatekeepers being contacted, one of the following messages is returned before the time limit in the configured timeout window expires:
   - LCF (location confirm)
   - LRJ (location reject)
   - RIP (response in progress)

   The RIP message is activated when the gatekeeper cluster connects to a remote gatekeeper (by means of UDP).

4. If the host portion of the new request-URI is not the address (FQDN or IP address) of the server itself, Cisco SPS performs domain routing using the host portion of the Request-URI.

IP Resolution

IP resolution is the conversion of each hop found by means of next-hop routing into an IP address. Standard IP resolution (through gethostbyname) is performed by either DNS, NIS, NIS+, or host file, depending on the IP resolution subsystem configured on the system where Cisco SPS is located.

Authentication. Authorization, and Access Control Lists

The Cisco SIP proxy server can provide authentication and authorization.

A major tool in providing authentication are access control lists (ACLs). ACLs can be upstream or downstream (previously they were upstream only). Upstream ACL continues to be used for authentication, and both upstream and downstream ACLs are used for determining trust relationships as they pertain to privacy functionality.

Two locations are supported: Authentication can occur at a RADIUS server or at the proxy server.

Two types of authentication are supported: HTTP digest authentication and HTTP basic authentication, both as described in RFC 2617. Either type can occur at either location.

Note Due to its weak security, basic authentication has been deprecated. This is a change from RFC 2543. It is not disabled or removed from Cisco SPS, but will no longer be supported or extended to interwork with new or modified functionality. We strongly discourage the use of basic authentication.

A RADIUS server operates in accordance with the RADIUS protocol—an IETF protocol based on UDP. RADIUS supports the exchange of a set of attribute/value pairs between client and server. For example, a Cisco SPS acting as a RADIUS client exchanges attribute/value pairs with a RADIUS server to provide authentication.

During authentication, the UAC password is stored as follows:

- For RADIUS-supported authentication, it is stored at the RADIUS server.
- For proxy-supported authentication, it is stored in a subscriber table in a MySQL database.

Tip For information about Cisco SPS and the RADIUS server, see the RADIUS interface specifications at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/index.htm or http://www.cisco.com/en/US/products/sw/voicesw/ps2157/index.html.

Authentication

The default authentication scheme is HTTP digest authentication performed at the Cisco SPS.

When digest authentication and basic authentication are performed at the proxy server, the username, as found in the authorization header or the proxy-authorization header, is the key to query the MySQL database.

If authentication takes place at the RADIUS server, Cisco SPS passes the username as one of the attribute/value pairs to the RADIUS server, where it can be used to key the user search before authentication. Additionally, you can configure Cisco SPS to add any desired SIP headers as VSAs in the authentication request to the RADIUS server.

Cisco SPS can expand the UserName before MySQL lookup or before passing it to the RADIUS server. This enables phone numbers to be expanded to full E.164 numbers before being processed.

If the virtual proxy host feature is enabled, the Username@domain (username found in the Authorization/proxy-Authorization header; domain name found in the From header) is the key used to query the MySQL database, just as when authentication is performed on a RADIUS server and the RadiusUserNameAttrAddDomain directive is enabled. Cisco SPS passes the Username@domain directive as one of the attribute/value pairs to the RADIUS server, where it can be used as the key for user searches.

Note Cisco SPS does not support native Apache-based virtual hosts. Native Apache-based virtual host refers to providing the illusion of more than one server on one system. For example, companies sharing a web server can have their own domains (www.company1.com and www.company2.com) and access to the web server.

Cisco SPS provides access-control lists and user authentication that you can combine to provide more complex access control. Some devices such as SIP gateways do not authenticate themselves. You can specify them in the access-control list, which instructs SPS to accept REGISTER and INVITE requests from them without the need for additional authentication.

You can configure Cisco SPS to challenge REGISTER and INVITE requests from other devices for user authentication.

Authorization

An authenticated user is authorized. Currently, you cannot authorize authenticated users according to specific user capabilities for service-provider voice applications.

DNS Support

Cisco SPS implements (in accordance with RFC 3263) DNS procedures for SIP clients and servers, including the use of naming-authority pointer (NAPTR), server (SRV), and ready-to-receive (RR) lookups by SIP entities. It also uses SIP-specific extensions for handling failures during lookups. RFC 3263 obsoletes RFC 2543 guidelines for DNS procedures and provides DNS configuration guidelines to SIP server administrators to help create secure and correct SIP services.

Cisco SPS uses DNS procedures for the following reasons:

- To support scalability and high availability. Typically, customers deploy SIP services in a farm of homogeneously configured proxy servers. DNS enables you to configure these farm members with prioritization and weights (thus supplying a crude level of capacity-based load balancing). DNS also provides failover capability in both upstream and downstream directions.
- To discover SIP servers in external domains. Specifically, SPS needs to determine the IP address, port, and transport protocol for the server. SPS can support TCP, UDP, and TLS for SIP signaling. SPS needs to be able to automatically determine which transport protocols are available with next-hop servers and in what order of preference.

Cisco SPS configuration for advanced DNS support is tightly coupled with the type of DNS support and configuration available. It is highly recommended that you configure the domain for which you are installing the proxy server using the guidelines mentioned in this document (see Appendix D, “DNS Setup”).

IP Security

IP security (IPSec) provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets moving between participating IPSec devices (peers) such as Cisco SPS and a UAC, SIP gateway, or another Cisco SPS. With IPSec, data can move across a public network without fear of observation, modification, or spoofing.

All IPSec combinations can successfully secure traffic to and from Cisco SPS on the specified Solaris and Red Hat Linux platforms.

Proxy-Server Farms

Cisco SPS is designed for deployment in proxy-server farms of up to two servers that act as one virtual server. Multiple proxy servers share database information and provide both redundancy and high availability.

A farm is represented by a domain or shared server name, where DNS determines the appropriate IP address. For example, a common ServerName configured for the farm may resolve in DNS to two hostnames and/or IP addresses, one for each server.

Registry and routing information is synchronized across farm members dynamically (that is, as they are committed; when you enter a registration or route on one farm member, the new information is immediately made available to both farm members). Configuration and subscriber information is synchronized dynamically as well. Configuration changes require a graceful restart, or in some cases a full restart.

A farm supports up to 20,000 registrations, 20,000 routes, and 20,000 subscribers.

Registrar Server for Multiple Domains

Cisco SPS can act as a registrar server for multiple domains: a single domain (ProxyDomain) for fully expanded E.164 numbers plus and zero or more additional domains within which private name spaces can exist.

For example, the single E.164 domain can be cisco.com; additional private domains can be a.com and b.com. In this situation, Cisco SPS accepts REGISTER messages for the following domains:

- <*>@a.com
- <*>@b.com
- <*>@cisco.com
- <e.164-number>@cisco.com
- <e.164-number>@a.com
- <e.164-number>@b.com

Cisco SPS enters registrations for these domains in the registry database as follows:

- It treats an INVITE message that is received for <e.164-number>@*.com as an INVITE message for <e.164-number>@cisco.com. All registrations for <e.164-number>@*.com are represented by a single entry in the registry database.
- It treats an INVITE message that is received for user@a.com, user@b.com, user@cisco.com as an INVITE message for user@a.com, user@b.com, or user@cisco.com, respectively. Registrations for user@a.com, user@b.com, and user@cisco.com result in three separate entries in the registry database.

Note that creation and administration of multiple domains of overlapping name spaces requires coordinated effort on the part of the Cisco SPS administrator and the administrators of the supported domains.

As an example of multiple domains, consider CompanyA and CompanyB in area code 408. CompanyA uses four-digit extensions 2000 to 7999 and CompanyB uses four-digit extensions 2000 to 7999. No DIDs are associated with these extensions. For users at CompanyA and CompanyB who need DIDs, CompanyA gets 1-408-555-[8000-8999] and CompanyB gets 1-408-666-[8000-8999]. Cisco SPS enters registrations in this example as follows:

- 2xxx at CompanyA registers as 2xxx@companyA.com
- 2xxx at CompanyB registers as 2xxx@companyB.com
- 8xxx at CompanyA registers as +14085558xxx@companyA.com
- 8xxx at CompanyB registers as +14086668xxx@companyB.com

Note x is a wildcard and can be replaced by any digit. For example, 2xxx can be 2000, 2001, 2999, and so on.

Call processing occurs as follows:

1. 2001 calls 2000
   - 2001 at CompanyA calls 2000 at CompanyA as 2000@companyA.com [1]
   - 2001 at CompanyA calls 2000 at CompanyB as 2000@companyB.com [2]
   - 2001 at CompanyB calls 2000 at CompanyB as 2000@companyB.com [1]
   - 2001 at CompanyB calls 2000 at CompanyA as 2000@companyA.com [2]
2. 8001 calls 2000
   - 8001 at CompanyA calls 2000 at CompanyA as 2000@companyA.com [1]
   - 8001 at CompanyA calls 2000 at CompanyB as 2000@companyB.com [2]
   - 8001 at CompanyB calls 2000 at CompanyB as 2000@companyB.com [1]
   - 8001 at CompanyB calls 2000 at CompanyA as 2000@companyA.com [2]
3. 2001 calls 8000
   - 2001 at CompanyA calls 8000 at CompanyA as +14085558000@companyA.com[3]
   - 2001 at CompanyA calls 8000 at CompanyB as +14086668000@companyA.com[4]
   - 2001 at CompanyB calls 8000 at CompanyB as +14086668000@companyB.com[3]
   - 2001 at CompanyB calls 8000 at CompanyA as +14085558000@companyB.com[4]
4. 8001 calls 8000
   - 8001 at CompanyA calls 8000 at CompanyA as +14085558000@companyA.com[3]
   - 8001 at CompanyA calls 8000 at CompanyB as +14086668000@companyA.com[4]
   - 8001 at CompanyB calls 8000 at CompanyB as +14086668000@companyB.com[3]
   - 8001 at CompanyB calls 8000 at CompanyA as +14085558000@companyB.com 4]
5. 2000 and 8000 call 18183635839
   - 2000 at CompanyA calls 18183635839 as +18183635839@companyA.com [5]
   - 8000 at CompanyA calls 18183635839 as +18183635839@companyA.com [5]

- 2000 at CompanyB calls 18183635839 as +18183635839@companyB.com [5]
- 8000 at CompanyB calls 18183635839 as +18183635839@companyB.com [5]

6. Authorized user John Doe (with neither CompanyA nor CompanyB) places calls as follows:
   - Calls 2000 at CompanyA as 2000@companyA.com [2]
   - Calls 2000 at CompanyB as 2000@companyB.com [2]
   - Calls 8000 at CompanyA as +14085558000@<proxy>.com [6]
   - Calls 8000 at CompanyB as +14086668000@<proxy>.com [6]

Note

- [1] The phones at both companies expand 2xxx to 2xxx@<proxy>.
- [2] This is done through full URL dialing.
- [3] The phones at companyA expand 8xxx to +14085558xxx@<proxy> and the phones at companyB expand 8xxx to +14086668xxx@<proxy>.
- [4] The phones at both companies expand [2–9]xxxxxx to +1408xxxxxxx@<proxy>. All authenticated users, regardless of domain, have access to the same routes for forwarding calls for +1408xxxxxxx.
- [5] The phones at both companies expand 1xxxxxxxxxx to +1xxxxxxxxxx@<proxy>. All authenticated users, regardless of domain, have access to the same routes for forwarding calls for +1xxxxxxxxxx.
- [6] Cisco SPS supports a default domain for E.164 numbers; such numbers are designated by the plus (+) sign.

Registry and Route Configurations

Basic Principles

A registry is a memory-mapped file maintained on each Cisco SPS server. It contains information for up to 20,000 registered users. (A user is a SIP endpoint that receives calls.) The file can be accessed quickly and persists across restarts. Registries are synchronized among farm members.

The route database is a memory-mapped file maintained on each Cisco SPS server. A route is a destination pattern plus an associated next-hop address and other attributes. It contains information for up to 20,000 routes. For example, a gateway serving the 408 area code in North America can be represented by a route as follows:

```
destination pattern    +1408.......
next hop               <address of gateway>
etc.
```

A route may similarly be used to specify the traffic that is handled by another proxy. The file can be accessed quickly and persists across restarts. Routes are synchronized between farm members.

Information pertaining to a single registered user (SIP endpoint) is called a registration. The registry is thus a collection of registrations.

SPS does not determine capabilities based on the registration; it determines only location (that is, transport, IP address, and port to be used to reach the endpoint). Static registration information exists simply to interwork with SIP endpoints that are not capable of registering for themselves.

Registry and Route Configuration Strategies

Whether you create registrations or routes is mostly a matter of preference. Most administrators tend to use static registration if the number of FXS ports is small and a route if the number is large. They also tend to use static registration if an FXS port is associated with a subscriber (for example, for call forwarding).

An example of a nonsubscriber registry is a Cisco IOS gateway connected to a PBX or a gateway that has analog phones connected via foreign exchange station (FXS) ports. If the gateway cannot dynamically register its endpoints with Cisco SPS, you must statically register them so that they can receive sessions from other endpoints. The FXS ports on the gateways are therefore subscribers with static registrations.

If a gateway has FXS ports for just a small number of extensions—say, for example, 5000 and 5001—you can create subscribers for 5000 and 5001, each with call forwarding to voice mail (5000@voicemail-server, 5001@voicemail-server), and then enter static registrations for 5000 and 5001 with the following contacts:

```
5000@<gateway-ip-address>:<gateway-port>;user=phone
5001@<gateway-ip-address>:<gateway-port>;user=phone
```

If a gateway has many (say, 10 or more) FXS ports, you might prefer to create a route rather than many static registrations. For example, if 100 FXS ports are mapped to users 5000–5099, you could create 100 subscribers with no static registrations, and then create a route such as the following:

```
destination-pattern: 50..
next-hop: <gateway-ip-address>
etc.
```

You could even create a route for a single endpoint such as the following:

```
destination-pattern: 5043
next-hop: <gateway-ip-address>
etc.
```

Spiralled and Looped Request Detection

A spiralled request is a request that revisits Cisco SPS with a new request-URI and that SPS considers to be a new transaction. A looped request is a request that contains the proxy's Via header; the request-URI is the same as it was when it first visited the proxy server.

At the proxy server, this has the following effects:

- The same call can be logged more than once.
- Different features can be invoked.
- If the Record-Route field in the INVITE message is enabled in the proxy server, the record-route procedure is performed more than once.

Note The Call-ID, From, To, and CSeq.seqnum fields of a spiralled request are the same as those of the request that visited the proxy server the first time.

Subscribers

A subscriber is a user (SIP endpoint) that has static subscriber information such as user ID, password, and call-forwarding settings. Subscribers generally register dynamically by means of a periodic REGISTER request from the subscriber's SIP endpoint (called a SIP user agent).

Subscriber records reside in the MySQL database subscriber table and are used for authorization and per-user call forwarding. Subscriber records are synchronized between replicated MySQL databases.

A subscriber can have multiple SIP agents and therefore multiple registrations. For example, a subscriber with a given user ID, password, call forwarding number, and so on might have an office phone, a cell phone, and a home phone, all of which register. When the subscriber receives a call, all three phones ring.

In general, you should not need to register a subscriber statically. In any case, you should never register the same contact for a subscriber both statically and dynamically. A mismatch could result between what the provisioning system considers to be registered and the registry information in shared memory that call processing uses.

TLS Support

Transport Layer Security (TLS) is an IETF protocol that replaces Secure Socket Layer (SSL) encryption technology. It provides secure transactions such as transmission of credit-card numbers for e-commerce.

Cisco SPS provides TLS for SIP signaling according to RFC 3261 recommendations. It supports TLSv1 and the following two cipher suits:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

The proxy server handles TLS using RSA key exchange. (RSA stands for Rivest, Shamir, and Adelman, inventors of the technique.) RSA is a public-key cryptographic system for encryption and authentication with Advanced Encryption Standard (AES)-128 or 3 Data Encryption Standard (DES) crypto methods in cipher-block-chaining (CBC) mode. It uses Hash-Based Message Authentication Code—Secure Hash Algorithm (HMAC-SHA) for message-authentication check. Cisco SPS does not explicitly change ciphers (a cipher is a cryptographic algorithm for encryption and decryption) during a session, but it allows session renegotiation (rehandshake) by clients. Cisco SPS supports session resumption; currently only 256 sessions can be cached.

Cisco SPS does not act as a certification authority, but requires that proper X.509 certificates and keys be installed. (For information on how to install TLS certificates and keys, refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.) From the Cisco SPS point of view, it does not matter if the certificate is from a well-known authority or is self-signed by the user. The proxy server needs only to have its certificate, key, and the root certificates installed as part of configuration.

Cisco SPS does not distribute certificates during registration. Rather, you must manage and distribute certificates. You are free to act as certification authority and sign or distribute certificates to your clients, in which case you need to install your self-signed root certificate in the proxy-server configuration.

TLS Client Behavior

The proxy server (as an SSL client) connects with a TLSv1 handshake. If there is a handshake failure, it falls back (for backward compatibility) to an SSLv23 handshake. (An SSL handshake differs from an SSL connection type, which is always TLSv1.) Once a handshake is successful, the proxy server verifies the authenticity of the downstream entity and tries to match the credentials presented in the certificate with the target of the URL for outgoing messages. In accordance with the SIP standard, the certificate should prove the authenticity of the entity by matching certificate credentials—IP address, host name, FQDN, or DNS. The proxy server looks at the SubjAltName extension for these values and then, for backward compatibility, it looks at the CN (common name) field.

TLS Server Behavior

On the server side, Cisco SPS accepts TLSv1 connections and can handshake in TLSv1 as well as SSLv23 mode. You can configure the proxy server to perform mutual TLS authentication, and hence challenge the client to prove its authenticity. It then extracts certificate credentials (IP, host name, FQDN, or domain name) and matches them with the Via header in the SIP message.

Additional References

For additional information related to Cisco SPS, look at the material listed in the following sections.

Related Documents

Related Topic	Document Title
Cisco SIP proxy server installation and configuration	• *Cisco SIP Proxy Server Version 2.2 Administrator Guide,* at http://www.cisco.com/en/US/products/sw/voicesw/ps2157/prod_technical_documentation.html • *Cisco SIP Proxy Server Version 2.2 Installation Guide*, at http://www.cisco.com/en/US/products/sw/voicesw/ps2157/prod_technical_documentation.html • *Release Notes for the Cisco SIP Proxy Server (SPS) Version 2.2* at http://www.cisco.com/en/US/products/sw/voicesw/ps2157/prod_technical_documentation.html
Technologies referenced within Cisco SPS software and documentation	• ADAPTIVE Communication Environment at http://www.cs.wustl.edu/~schmidt/ACE.html • Apache Software Foundation at http://www.apache.org • Linux FreeS/WAN at http://www.freeswan.org/ • Linux Online at http://www.linux.org • MySQL at http://www.mysql.com • Red Hat at http://www.redhat.com • Solaris Data Encryption website: http://wwws.sun.com/software/solaris/encryption/ • Sun Microsystems at http://www.sun.com • Sun ONE Software at http://wwws.sun.com

Standards

Cisco SPS supports the following standards and protocols.

	Function

MIBs

MIBs[1]	MIBs Link
• CRITAPP-MIB (critagt)—Start and stop CSPS • LOG-MIB (logagt)—Monitor CSPS error_log and access_log sizes • DISMAN-SCRIPT-MIB (smagt)—Gracefully restart CSPS • DISMAN-EVENT-MIB (eventagt)—Monitor CPU load • HOST-RESOURCES-MIB (hostagt)—Check memory size, disk space • RFC1213-MIB (mib2agt)—Check link up/down status • SYSAPPL-MIB (sappagt)—Check what applications are installed and running on the system	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB locator at the following URL: http://www.cisco.com/go/mibs

1. Not all supported MIBs are listed.

RFCs

RFCs[1]	Title
768	User Datagram Protocol (UDP)
791	Internet Protocol (IP)
793	Transport Control Protocol (TCP)
1035	Domain Name System (DNS)
2246	Transport Layer Security (TLS)
2327	Session Description Protocol (SDP)
2543	*Session Initiation Protocol (SIP)*
2617	HTTP Authentication: Basic and Digest Access Authentication
2865	Remote Authentication Dial In User Service (RADIUS)
2866	RADIUS Accounting

RFCs[1]	Title
2916	E.164 number and DNS
3261	*SIP: Session Initiation Protocol*
3263	*SIP: Locating SIP Servers*
3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks

1. Not all supported RFCs are listed.

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

CHAPTER 2

Configuring Cisco SPS

Configuring Cisco SIP proxy server (Cisco SPS) involves tasks such as establishing a server farm, setting up individual proxy servers to handle tasks, and adding subscribers, registries, and routes.

You can configure Cisco SPS in either of two ways:

- Using the GUI-based provisioning system (recommended)
- Manually editing text-based files

This chapter describes use of the GUI-based provisioning system. We strongly urge you to use the GUI-based provisioning system to configure the Cisco SPS. (For information on manual configuration, see Appendix B, "Manual Configuration.")

Caution All Cisco SPS 1.*x* versions require manual configuration. Therefore, for backward compatibility, Cisco SPS supports manual editing of all configuration files. However, if you use the GUI-based provisioning system, do not attempt manual editing. Manual changes to any configuration file written by the GUI are overwritten when the GUI is used again.

Contents

Note
- For troubleshooting information, see Chapter 5, "Troubleshooting" and Appendix E, "SIP Call-Flow Scenarios."

- In configuration windows, an asterisk, *, indicates a required entry.
- The GUI-based provisioning system contains menus for activities described both in this chapter and in Chapter 3, "Operating and Maintaining Cisco SPS."

Prerequisites

- Install Cisco SPS and activate the license (see the *Cisco SIP Proxy Server Version 2.2 Installation Guide*).

Note As described in the reference above, Cisco SPS runs only on the following operating systems: Red Hat Enterprise Linux AS, ES, or WS 3.0; or Sun Solaris 8 Operating Environment. It is critical that you use only a supported operating system. Cisco SPS does not run on other than these.

- Familiarize yourself with the concept of graceful restart. Graceful restart causes the system to use a new SIP directives file. The proxy-server daemon (parent process) remains alive, rereads the directives file, tears down child processes as they become idle, and spawns new child processes with the new configuration. Because call processing is not interrupted, graceful restart is a useful way to activate a new configuration without dropping calls.
- Familiarize yourself with procedures for accessing the Cisco SPS GUI and for sorting and searching data, as described in the tip below.

Tip **To access the Cisco SPS GUI, do the following:**

1. Go to the following (default) directory or your Windows desktop:

   Linux: `/usr/local/sip/gui/`

   Solaris: `/opt/sip/gui/`

2. Open the GUI by entering the **CiscoSPS** command or double-clicking the CiscoSPS desktop icon.
3. Enter your password (default is cspsuser).
4. During installation or upon previous use, did you enter the correct pserver hostname?
   - If yes, click **OK**.
   - If no, click **more>>**, enter the hostname and port number (the default port is 26005), and click **OK**.

   The Cisco SPS main menu displays. The pserver hostname and port number automatically reappear at the next login.

To sort and search for entries, do any of the following:

- To resize a column, place the cursor on the vertical line dividing column headers and drag it to a desired position. To rearrange column order, place the cursor on a header and drag it to a desired position.
- To display only specific entries, use the search tool (field, operator, search string) at page top.
- To display all entries, use the search tool with the search string set to *.

Note If your list of entries is extremely large (the system limit is 20,000 records), do not display all entries. Apply a filter to display only specific entries.

- To display entries in a particular order, use the column-heading sort arrows.
- To repeat the last search that you performed, click **Refresh**.

How to Configure Farms and Proxy Servers

You can view and change farm and proxy-server settings.

Note To add or delete a proxy server, do not use this procedure. Rather, use the sps_setup script, as described in the *Cisco SPS Proxy Server Version 2.2 Installation Guide*.

Configurable farm and proxy-server settings are listed alphabetically in Table 2-1. The table describes the purpose of each setting and what directives are configurable for that setting. To see how the settings appear in the GUI interface Farm/Proxies window, see Figure 2-1.

Note The Farm/Proxies window contains substantial online help that explains configurable directives. Information on configurable directives is presented in more detail below for those settings that are new or most greatly enhanced with the current Cisco SPS release: access control and privacy.

Table 2-1 Configurable Farm/Proxies Settings

Setting	Purpose	Configurable Directives
Access Control	Controls whether specified upstream and downstream network devices can send requests through the proxy server.	• Access order—Default access state and the order in which allow and deny directives are evaluated. Choices are the following: – Deny,Allow—Evaluate deny directives before allow directives. Access is allowed by default. – Allow,Deny—Evaluate allow directives before deny directives. Access is denied by default. • Satisfy conditions—Access policy if both deny/allow and authentication are used and the authentication module is turned on (if the module is turned off, the authentication check is considered successful). Valid values are as follows: – all—Sending host must be allowed and must pass authentication. – any—Sending host is allowed or denied but passed authentication. • Deny/allow rules—You can set up deny and allow rules with the following components: – Address—Hostname, IP address, or other characteristic – Upstream (from)—Whether SPS can receive messages with privacy-related headers from that address – Downstream (to) —Whether SPS can send messages with privacy-related headers to that address
Accounting	Controls the sending of accounting records by the proxy server to a pair of RADIUS servers.	• Whether to send the following: server-side records for successful calls, client-side records for successful calls, records for unsuccessful calls • Record and time formats • Primary and secondary RADIUS server information • SIP headers

Table 2-1 Configurable Farm/Proxies Settings (continued)

Setting	Purpose	Configurable Directives
Authentication	Verifies the identity of a person or a process, for purposes of security, before servicing their transactions.	• Realm used in authentication response headers • Authentication server and scheme • Digest QOP and algorithm to include in a digest challenge • Whether to remove the Proxy-Authorization header before forwarding a request downstream • Whether to allow third-party registration and INVITE requests • RADIUS- authentication skew time (in seconds) for which a challenge is valid • Primary and secondary RADIUS server information • SIP headers
Call Forward	Controls redirection of incoming calls on a per-subscriber basis. Requires a MySQL database.	• Whether to forward calls unconditionally • Whether to forward calls under the following circumstances: – When a call is not answered; if yes, time (in ms) after which to forward – When a busy response is received – When a UAC is unavailable; if yes, time (in ms) after which to forward • Whether to include the Diversion header in SIP messages • Name used for diversion headers generated by this proxy server
Debug and Logs	Sets debug flags and error-log parameters.	• Debug flags for various types of logs • Log location and level (in decreasing order of verbosity) • Custom-log information • Log formats • Whether to log SIP and shared-memory statistics; if yes, logging duration (in seconds) **Note** For more information on debugging and error logs, see Chapter 3, "Operating and Maintaining Cisco SPS."
ENUM	Specifies rules for translating E.164 (or other number-plan) phone numbers into URLs and IP addresses by means of DNS lookup.	• Private search domain (optional) • Global search domain for use when the Request-URI user begins with a plus (+) character (indicating a global domain)
Farming	Specifies farm-member host and port numbers and additional parameters.	• Farm-member information

Table 2-1 Configurable Farm/Proxies Settings (continued)

Setting	Purpose	Configurable Directives
GKTMP	(GateKeeper Transaction Message Protocol) Controls use of GKTMP for communication with external applications. SPS processes use GKTMP to initiate a TCP connection with a network-application-manager (NAM) server.	• Transport type • Master and secondary server information
MySQL	Configures the interface to the subscriber table in the MySQL database. MySQL stores call-forwarding information and authentication passwords.	• Host and secondary-host names or IP addresses for the MySQL server • Username and password to access the MySQL server • Connect timeout (in seconds) when attempting to access the MySQL server

Number Expansion	Expands telephone numbers from extension numbers to full E.164 telephone numbers and back.	• Number-expansion plan
		Examples:
		– To expand 5*xxxx* to +1303555*xxxx*, add this rule:
		`From 5....  To +1303555....`
		– To strip +1888555*xxxx* to *xxxx*, add this rule:
		`From +1888555....  To ....`

Table 2-1 Configurable Farm/Proxies Settings (continued)

Setting	Purpose	Configurable Directives
Privacy	Specifies varieties of calling-line-identity privacy for individual subscribers, based on the trustworthiness of upstream and downstream addresses.	• P-Asserted-Identity (PAI) does the following: – Removes PAI headers from the received INVITE request if the sender is not trusted – Adds its own PAI header to the INVITE request if the sender has been authenticated by SPS – Removes PAI headers before sending if the receiver is not trusted • Remote Party ID (RPID) does the following: – Removes RPID headers from the received INVITE request if the sender is not trusted – Adds its own RPID header to the INVITE request if the sender has been authenticated by SPS – Removes RPID headers before sending if the receiver is not trusted • Diversion does the following: – Removes Diversion headers from the received INVITE request if the sender is not trusted – Validates and potentially rewrites the topmost Diversion header from the received 302 response if the sender is not trusted – Removes identity information from Diversion headers before sending if the receiver is not trusted
RAS	(Registration, Admission, and Status protocol) Enables communication with a H.323 gatekeeper.	• Whether to accept the first or best location-confirm (LCF) message within a specified timeout interval (in ms) • Location-request (LRQ) method and window time (in ms) • Time-to-live (TTL) number of hops • Whether to allow translation—that is, allow the gatekeeper to replace the dialed phone number • Default tech-prefix action to perform to the prefix in an outgoing INVITE request • Transport type for routes specified in LCF responses • Tech-prefix used when the dialed number matches the specified pattern • Gatekeeper clusters; you can group gatekeepers into clusters of up to five gatekeepers and assign a priority to each cluster

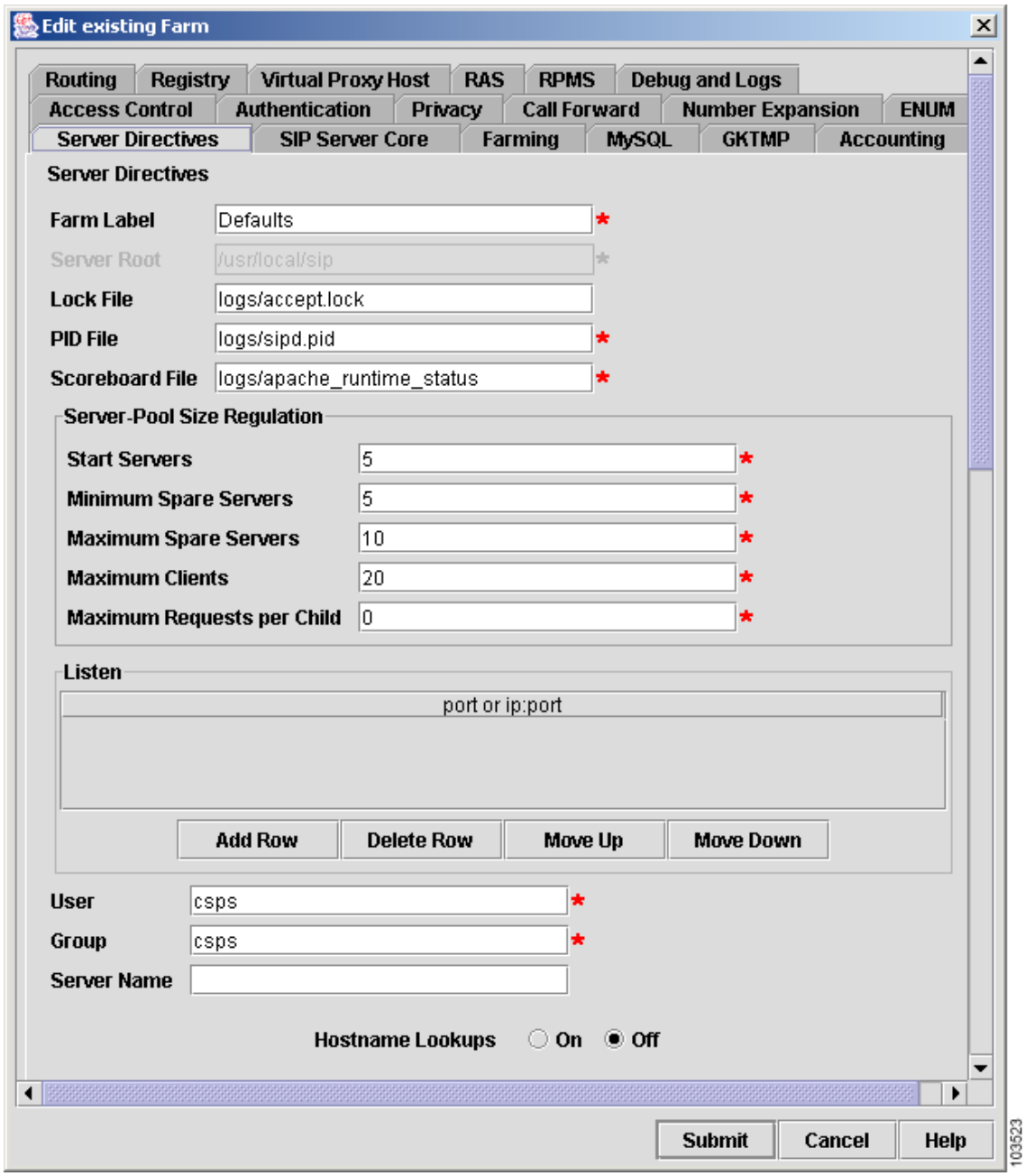
Table 2-1 Configurable Farm/Proxies Settings (continued)

Setting	Purpose	Configurable Directives
Registry	Specifies use of registry services. A registry is the location information for a registered endpoint.	• Shared memory address of the registration table • Rendezvous name and directory location of the registration database • Remote-update port number of the registration-database server for all members of a server farm; must be the same for all farm members • Max database age on boot (in seconds); a file older than this is deleted
Routing	Specifies use of and parameters for next-hop routing.	• Shared memory address of the routing table • Rendezvous name and directory location of the routing database • Remote-update port number of the routing-database server for all members of a server farm; must be the same for all farm members • Max database age on boot (in seconds); a file older than this is deleted • Whether to perform domain next-hop routing. This type of routing uses the host portion of the Request-URI as the key in obtaining the next one or more hops for a request.
RPMS	(Cisco Resource Policy Management System) Specifies use of RPMS policy management for data and voice platform resources, enabling provision of a variety of services to a variety of customers on a single set of gateways. Uses RADIUS messages to communicate with RPMS servers.	• Whether to preauthorize new INVITE requests • Whether the proxy server supports query. Leave as is • RPMS server's IP, port, and secret • Whether to do a preauthorization check if a new INVITE request's previous-hop IP matches any one from this list
Server Directives	Sets server-global (protocol-independent) standard Apache directives that define overall server operation. Examples of such directives include directory in which the Cisco SPS software resides and how child processes function. Directives that you set here reside in the system's configuration file. **Note** For more information on Apache directives, see the Apache Software Foundation website at www.apache.org.	The following, and more: • Farm label • Server root • Various filenames • Server-pool sizes • Listen information • User • Group • Server name

Table 2-1 Configurable Farm/Proxies Settings (continued)

Setting	Purpose	Configurable Directives
SIP Server Core	Sets server-core (protocol-specific) directives that define server and DNS operation. Examples of such directives include whether the server functions as a redirect, registrar, or proxy server and, if as a proxy server, whether it is transaction stateful or stateless. **Note** Use this setting to configure shared memory size, as described in the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.	The following, and more: • Proxy domain • Max forks • Various retransmission times, backoff intervals, wait times, and hold times • Various retransmit counts • Shared memory size • Registry cleanup interval • TOS byte value • Various SIP ports • Various RADIUS directives • SIP over TCP information
Virtual Proxy Host	Specifies use of the proxy server as a registrar for multiple domains (a single domain for fully expanded E.164 numbers and zero or more additional domains within which private name spaces may exist). An example of such usage is two or more companies that share a web server, each with their own domain (www.company1.com and www.company2.com) and access to the web server. **Note** Use this setting to configure multiple-domain handling.	• Whether to use virtual-proxy host

Figure 2-1 Farm/Proxies Window, with Tabs for Configurable Settings



Prerequisites

- Review the "Prerequisites" section on page 2-2.
- Physically install and configure each proxy server that is to become a farm member.
- Set up a proxy-server farm as described in the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Farm/Proxies.**

Step 2 Verify, on the **Server Directives tab window that displays,** that farm label and other related information displays correctly.

Information should display automatically if you installed Cisco SPS with the SPS setup (sps_setup) script. If it does not, enter the information.

Step 3 Make changes as follows:

a. **Click the tab for a configurable setting. (Tabs are listed in Table 2-1 and shown in Figure 2-1.)**

b. Turn the setting On or Off or enter information as needed.

c. If you turn a setting On, select or type new setting details as needed. (Click **Help** to view context-sensitive online help.)

Note If a setting is Off, associated directives are dimmed. You can make changes only if the setting is On.

Step 4 To make additional changes for the same or another tab, repeat Step 3 as needed.

Step 5 Click **Submit**.

Step 6 Do one of the following:

- To restart or gracefully restart and have changes take effect now, click **Go to Proxy Control screen**.
- To continue with configuration and have changes take effect later, click **OK**.

How to Configure Subscribers

You can add, change, or delete information about subscribers. A subscriber is a SIP endpoint—typically (but not necessarily) a phone extension used by an individual—that has static, configurable subscriber information. Configurable subscriber information includes the following:

- User name, domain name, and password
- First name, last name, and middle name (useful for sorting, searching, and filtering)
- Destination URLs for the following:
  - Call Forward No Answer (CFNA)—Forward to this number if the original call is unanswered
  - Call Forward Unconditional (CFUNC)—Forward to this number unconditionally
  - Call Forward Busy (CFB)—Forward to this number if the original call returns busy
  - Call Forward Unavailable (CFUNV)—Forward to this number if this subscriber currently has no active registration or does not respond
- Redirect DN Privacy—Allows individual subscribers to keep their identity private when they use call forwarding.

Prerequisites

- Review the prerequisites in the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Subscribers**.

Step 2 To add a new subscriber, do the following:

a. Click **Add**.

b. Enter subscriber information. (Any field that has a red asterisk must have an entry.)

Note Enter each subscriber's first, middle, and last name as might be useful to you for sorting, searching, and filtering. They are not used by Cisco SPS.

Note You cannot configure a subscriber if the first 50 characters of the username are the same as those of an existing subscriber.

Step 3 **To edit or delete an existing subscriber, do the following:**

a. **Locate** the **subscriber (see tips above) and click to select.**

b. To edit subscriber information, do the following:

1. Click **Edit**.
2. Edit fields as needed.

Note A password displays as a series of asterisks (example: *****). If a system user has forgotten a password, assign a new one.

3. Click **Submit**.

c. **To delete the subscriber, c**lick **Delete > Yes**.

Step 4 When done, confirm your changes by performing a search to display the new or changed information (see tips above)**.**

What to Do Next

You can continue with any additional configuration tasks.

How to Configure Registries

You can add, change, or delete a registry. A registry is the location information for a registered endpoint. Registries reside in the registry database. Configurable registry information includes the following:

- User
- Domain name
- Contact, contact user type, and contact port
- Transport protocol
- Expiration

Note If you need to add a large number of registries, as an alternative to this procedure you can do bulk provisioning as described in the "How to Import and Export Bulk Routing and Registry Data" section on page 2-16.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Registry**.

Step 2 Display existing registries by performing a search with the search tool (see tips above). Both dynamic and static registries display; you can differentiate dynamic from static by their expiration times.

Step 3 To add a new registry, do the following:

a. Click **Add**.

b. Enter registry information. (Any field that has a red asterisk must have an entry.)

Note If you set an expiration time, the registry automatically expires from Cisco SPS at the time you specify. However, the data remains in the system until you manually remove it or until the database is accessed by sysadmin_sps_regroute tool.

Note You cannot use wildcards in a registry. If you want to use wildcards, configure a route instead.

c. Click **Submit**.

Step 4 **To edit or delete an existing registry, do the following:**

a. **Locate the registry (see tips above) and click to select it.**

b. To edit the registry, do the following:

1. Click **Edit**.
2. Edit fields as needed.
3. Click **Submit**.

c. **To delete the registry, c**lick **Delete > Yes**.

Step 5 When done, confirm your changes by performing a search to display the new or changed information (see tips above)**.**

What to Do Next

You can continue with any additional configuration tasks.

How to Configure Routes

You can add, change, or delete dynamic and static routes.

A dynamic route is a path through the network that is automatically calculated according to routing protocols and routing update messages. A static route is a fixed path through the network that you explicitly configure. Static routes take precedence over dynamic routes and are synchronized among farm members. Configurable route information includes the following:

- Destination pattern and type
- Next hop and next-hop port
- Transport protocol
- Priority and weight
- Tech-prefix action
- Allow less-specific route
- Route block
- In service
- Label

Define destination patterns for routes when setting up a static route with Cisco SPS as follows:

- Use user=phone when routing based on the phone number in the user portion.

  Example: `+14085550122@cisco.com;user=phone` (where 14085550122 is an E.164 number)

  Example: `50122@cisco.com;user=phone` (where 50122 is an unambiguous extension within the cisco.com domain)

- Use user=ip when routing based on the domain portion (aka domain routing).

  Example: `bsmith@foo.com;user=ip (`where foo.com is a domain for which a domain route exists)

You can use any of the characters included in the following directive when specifying a destination pattern, with the following caveat:

- NumericUsernameCharacterSet—Set of characters that Cisco SPS uses to determine whether the user-information portion of a Request-URI is in a category that applies to the "NumericUsernameInterpretation" processing step. This set does not apply to any user-information parameters.

  Default is +0123456789.-() (global phone number combinations). For more information on this directive, see the sipd.conf file.

Caution Some characters are treated as visual separators (examples: `( ) . -`). These characters are removed before looking in the route database. Do not include them when defining a route destination pattern.

Special characters for defining a route are as follows:

- `*` indicates a multiple-digit wildcard (example: `9*` matches `911`, `914085551212`)
- `.` indicates a single-digit wildcard (example: `9..` matches `911` but not `9111`)
- `\*` indicates an actual '*' character (example: `\*69` matches `*69`)

Note If you need to add a large number of routes, as an alternative to this procedure you can do bulk provisioning as described in the "How to Import and Export Bulk Routing and Registry Data" section on page 2-16.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Routes**.

Step 2 Display existing routes by performing a search with the search tool (see tips above).

Step 3 To add a new route, do the following:

a. Click **Add**.

b. Enter route information. (Any field that has a red asterisk must have an entry.)

c. Click **Submit**.

Step 4 **To edit or delete an existing route**, do the following:

a. **Locate the route (see tips above) and c**lick to select it.

b. To edit the route, do the following:

1. Click **Edit**.
2. Edit fields as needed.
3. Click **Submit**.

c. **To delete the route, c**lick **Delete > Yes**.

Step 5 When done, confirm your changes by performing a search to display the new or changed information (see tips above)**.**

What to Do Next

You can continue with any additional configuration tasks.

How to Stop and Start a Proxy Server

You can stop or start (restart or gracefully restart) one or more proxy servers.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Proxy Control**.

Step 2 In the Select column, select the one or more proxy servers whose operation you want to stop or restart.

Note You must perform operations listed in the Required Operation column on the corresponding farm member to put into effect any changes you made to Cisco SPS configuration parameters. Red entries require either a Restart or Graceful Restart. Gray entries require no operation.

Step 3 Click Stop, Restart, or Graceful Restart.

How to View Persistent TCP Connections

You can add, change, or delete persistent TCP connections.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Persistent TCP**.

Step 2 View and add or delete entries as needed.

How to Import and Export Bulk Routing and Registry Data

You can import or export bulk routing and registry data in comma-separated-value (csv) form. You can manipulate csv data manually or load it into Microsoft Excel for a more user-friendly table format. Each line should contain one registry or routing entry.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Registry** or **Routes**.

Step 2 Right-click the page.

Step 3 To import data into the GUI-based provisioning system, do the following:

a. Click **Import**.

b. Select the source directory and enter the source filename.

c. Click **Import**.

The data is read into memory and sent to the pserver for parsing and storage.

d. Review status messages and address any errors that are generated during import. Errors can be any of the following:

- Syntax errors: missing quotes, too many or too few elements in a line
- Semantic errors: out-of-range or otherwise invalid values, characters instead of numbers
- Other errors: database overflow

e. Verify that import is successful by refining your search parameters to display the new data. and clicking **Search**.

Step 4 To export data from the GUI-based provisioning system, do the following:

a. Click **Export**.

b. Select a destination directory and enter a destination filename.

c. Click **Export**.

How to Configure Administrator Accounts

You can configure user IDs on your Cisco SPS—that is, control who, in addition to yourself, can access Cisco SPS and, among those with access, who can change each configurable parameter. Configurable account information includes the following:

- User ID
- Optional password
- Various levels of read-write permission, including read only

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 To edit your own account, do the following:

a. From the Cisco SPS main menu (see Tip for how to access), click **My Account**.

b. Edit fields as needed.

c. Click **Submit**.

Step 2 To add a new account, do the following:

a. From the Cisco SPS main menu (see Tip for how to access), click **Administrator Accounts**.

b. Click **Add**.

c. Enter account information. (Any field that has a red asterisk must have an entry.)

d. Click **Submit**.

Step 3 To edit or delete an existing account, do the following:

a. From the Cisco SPS main menu (see Tip for how to access), click **Administrator Accounts**.

b. **Locate the account (see tips above) and click to select it.**

c. To edit the account, do the following:

1. Click **Edit**.

2. Edit fields as needed.

Note A password displays as a series of asterisks (example: *****). If a system user has forgotten a password, assign a new one.

3. Click **Submit**.

d. **To delete the account, c**lick **Delete > Yes**.

Step 4 When done, click **Refresh to redisplay all accounts, unfiltered**.

How to Configure TLS Support

You can configure TLS support on the proxy server.

Note To learn how to set up Transport Layer Security (TLS) certificates, see the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **SIP Server Core**.

Step 2 In the SIP-over-TLS area, set the following directives as needed:

- **Enable TLS**—Enables TLS. Default is Off.
- **Port**—TLS port. Default is 5061.
- **Certificate File**—Location of the privacy-enhanced mail (PEM)-encoded certificate file for the server
- Certificate Key File—Location of the PEM-encoded private key file for the server.
- CA Certificate File—Location of certificates of the certification authorities with whose clients Cisco SPS deals. These certificates are used for client authentication. The file is simply a concatenation of the various PEM-encoded certificate files, in order of preference.
- Mutual Authentication—Directs the server-side TLS to perform mutual authentication when accepting a new connection from TLS clients.
- **Allow SIP TLS Conversion to SIP**—Gives explicit permission for a proxy server to terminate incoming SIPS requests on the SIP contacts. This is a security risk, and should be used very carefully. Use it only if you know in advance that your endpoints and gateways are incapable of receiving sips/TLS connections. Default is Off.
- **SIP TLS Session Timeout**—Server-side session cache timeout value, in seconds. Sessions are not reusable after this timeout expires. Default is 300.

Step 3 Set the following directives as needed. (They indirectly control TLS functionality.)

- **Stateful Server**—Enables TLS functionality only when SPS runs in stateful mode.
- **Add Record Route Header**—If the proxy server is not configured to add record routes, disables translation from SIP to SIPS and vice versa.
- **SipTcpReuseConnection**—If the proxy server is configured not to reuse TCP connections, also prevents TLS from reusing the connections. This might result in poor performance; hence, whenever you intend to use TLS, set this to On.

Step 4 Click **Submit**.

Step 5 Do one of the following:

- To restart or gracefully restart and have changes take effect now, click **Go to Proxy Control screen**.
- To continue with configuration and have changes take effect later, click **OK**.

How to Configure Proxy-Server DNS Behavior

You can configure varying degree of DNS support, depending on your requirement. You configure the proxy server to locate other SIP services and then you set directives.

Prerequisites

- Review the "Prerequisites" section on page 2-2.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), click **Farm/Proxies.**

Step 2 Verify, on the **Server Directives window that displays,** that farm label and other related information displays correctly.

Information should display automatically if you installed Cisco SPS with the SPS setup (sps_setup) script. If it does not, enter the information.

Step 3 **Click SIP Server Core.**

Step 4 Set the following DNS directives as needed:

- **SRV Lookups For FQDN Only**—Enables SRV DNS lookups only on FQDN hosts (an FQDN is a fully qualified domain name). A Request-URIs URL whose host portion is not an IP address and has a period is considered an FQDN. The system normally performs SRV DNS lookup for any host portion that does not contain a target port. Default is Off.
- **Allow NAPTR Lookup**—Enables naming-authority-pointer (NAPTR) lookup logic on the proxy server. Default is On. If this directive is Off, use TransportPrefOrder to select a transport.
- **Transport Preference Order**—Transport preferences for times when NAPTR cannot be used or is unsuccessful. Valid values are the following:
  - For TLS first, the following: TLS_TCP_UDP (default), TLS_UDP_TCP, TLS_TCP, TLS_UDP, TLS (if SipTlsEnable is disabled, TLS is ignored)
  - For TCP first, the following: TCP_TLS_UDP, TCP_UDP_TLS, TCP_TLS, TCP_UDP, and TCP
  - For UDP first, the following: UDP_TLS_TCP, UDP_TCP_TLS, UDP_TLS, UDP_TCP, and UDP

Step 5 Set the following proxy-server directives as needed:

- **Proxy Address Resolution Type**—Type of DNS configuration for SIP services in the proxy-server domain. Valid values are the following:
  - IP—No DNS configuration is available; the proxy server should use IP addresses in the headers. This is the default setting.
  - A—DNS is set up with A records corresponding to the ServerName directive. The proxy server uses this value in headers.
  - SRV—If the ServerName directive is not enabled, the proxy server uses its host-name SRV (which indicates that the proxy server domain has SRV records configured), and hence uses the value of the ProxyDomain directive in headers.

Note To set up DNS records for the proxy-server domain (ProxyDomain directive) and proxy-server farm name (ServerName directive), refer to Appendix D, "DNS Setup."

Step 6 Click **Submit**.

Step 7 Do one of the following:

- To restart or gracefully restart and have changes take effect now, click **Go to Proxy Control screen**.
- To continue with configuration and have changes take effect later, click **OK**.

CHAPTER 3

Operating and Maintaining Cisco SPS

This chapter provides information on how to operate and maintain the Cisco SIP proxy server (Cisco SPS). Operation and maintenance involves starting and stopping the system and database, working with system logs, and backing up and restoring the system.

Contents

Note For additional related information, see the following:

- For background information on concepts relevant to this chapter, see Chapter 1, "Cisco SPS 2.2 Overview."
- For additional information on use of the GUI-based provisioning system, see Chapter 2, "Configuring Cisco SPS."
- For troubleshooting information, see Chapter 5, "Troubleshooting" and Appendix E, "SIP Call-Flow Scenarios."
- For information on Apache directives, see the The Apache Software Foundation website at www.apache.org.

How to Operate Cisco SPS

You can start and stop Cisco SPS in any of three ways:

- Using the GUI-based provisioning system (recommended and described below)
- Using CIAgent and the CIAgent Dr-Web interface (described in Chapter 4, "Monitoring System Status")

- Using the sip script that is generated when you run the SPS setup (sps_setup) script (described in Appendix C, "Manual Operation and Maintenance")

Note The results of any operation that you perform outside of the GUI-based provisioning system (such as using the sip script) are not reflected in the GUI. If such an operation was performed outside of the GUI, you might need to update the required operation manually.

Tip To access the Cisco SPS GUI-based provisioning system, use this procedure:

1. Go to the following (default) directory or your Windows desktop:

   Linux: `/usr/local/sip/gui/`

   Solaris: `/opt/sip/gui/`

2. Enter the **CiscoSPS** command or double-click the CiscoSPS icon to open the Cisco SPS GUI.
3. Enter your password (default is cspsuser).
4. During installation, did you enter the correct value for the pserver location?
   - If yes, click **OK**.
   - If no, click **more>>**, enter the pserver host name and port number, and click **OK** (the default port is 26005).

The Cisco SPS main menu appears. The pserver host name and port number automatically reappear at the next login.

Tip Cisco SPS provides a means not only to stop and restart, but also to gracefully restart the proxy server. Graceful restart causes the system to use a new SIP-directives (sipd.conf) configuration file. The SIP proxy server (sipd) daemon (parent process) remains alive, rereads the sipd.conf file, tears down child processes as they become idle, and spawns new child processes with the new configuration. Call processing is not interrupted as a result. Graceful restart is therefore a useful way to activate a new configuration without dropping calls.

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), choose **Proxy Control**.

The screen displays all system proxy servers and their current running state.

Step 2 Click the Select box next to the one or more proxy servers whose operation you want to control.

Step 3 Click **Stop**, **Restart**, or **Graceful restart** (see tip above).

Step 4 Click **Yes**.

How to Operate MySQL

The MySQL database starts automatically when you run the SPS setup (sps_setup) script. You can also start the database manually as described below.

User and root passwords are set when you run the script. You can, however, change the root password.

Note The start and stop procedures might yield the error message "/etc/init.d/mysql: @HOSTNAME@: not found." Ignore the message; MySQL starts and stops properly.

Detailed Steps

Step 1 Log in to Cisco SPS as root.

```
> su root
```

Step 2 To start MySQL, do either of the following:

- Use the **start** command:

  ```
  # /etc/init.d/mysql start
  ```

- Use the safe MySQL daemon (safe_mysqld) script:

  Linux: `# /usr/local/mysql/bin/safe_mysqld &`

  Solaris: `# /opt/mysql/bin/safe_mysqld &`

Step 3 To stop MySQL, use the **stop** command:

```
# /etc/init.d/mysql stop
```

Step 4 To change the MySQL administrator password, enter the following sequence of commands:

Linux:
```
# /usr/local/mysql/bin/safe_mysqld --user=mysql &
# /usr/local/mysql/bin/mysqladmin -u root -p<old_password> password <new_password>
# /usr/local/mysql/bin/mysqladmin -p reload
```

Solaris:
```
# /opt/mysql/bin/safe_mysqld --user=mysql &
# /opt/mysql/bin/mysqladmin -u root -p<old_password> password <new_password>
# /opt/mysql/bin/mysqladmin -p reload
```

How to Manage Log Files

This section contains the following information:

Information About Log Files

During Cisco SPS operation, each system component writes to one or more log files (see Table 3-1).

			Associated Configuration File[2]

2. Located in the <ServerRoot>/conf/ directory. Where possible, configure debugging using the GUI-based provisioning system rather than by manually editing these files.

Debugs are written to the file <ServerRoot>/logs/error_log.

Log files are text files that you can view with any text editor. They contain detailed event information in hexadecimal code. In most cases, you can configure their content and level of detail by setting the debug level or verbosity. Although you should do so using the GUI-based provisioning system as described below, you can also do so by manually editing the DebugLevel directive in the associated configuration file.

Caution Log-file size is limited to 2 Gb. It is important to ensure that neither the access_log file nor the error_log file exceeds this size.

File Verbosity

Error logs are either lengthy or abbreviated, according to whether or not debug flags are turned on.

- Lengthy format prints when a debug flag is turned on.

  Debug Output Example

  ```
  [Fri Apr 13 22:29:37 2001] sip_protocol.c(4322) Received 291 bytes UDP packets from
  10.80.36.85:50117
  REGISTER sip:64.102.93.77 SIP/2.0
  Via:SIP/2.0/UDP 10.80.36.85:5060
  From:sip:IPphone-2@64.102.93.77
  To:sip:IPphone-2@64.102.93.77
  Call-ID:c3943000-ee2f9c88-23f9821e-382e3031@10.80.36.85
  CSeq:101 REGISTER
  Contact:<sip:IPphone-2@10.80.36.85:5060>
  Expires:3600
  ```

- Abbreviated format prints when a debug flag is turned off (default).

  Debug Output Example

  ```
  [Fri Apr 20 21:44:51 2001] [notice] A new sipd child process (27413) has started.
  ```

File Rotation

On even a moderately busy server, the quantity of information stored in the log files is very large. The access log file typically grows 1 MB or more per 10,000 requests. The error log file can grow even faster, depending on its configured verbosity.

It is consequently necessary to cause the system periodically to cease writing to old log files, move or delete those files, and begin writing to new log files—that is, to rotate log files. You cannot rotate files while the server is running, however, because Cisco SPS continues writing to log files as long as they are open. Instead, you must gracefully restart the server to cause it to continue to write to the old log files while it finishes serving old requests, and then to open new log files. The system must wait for some time after the graceful restart before processing the new log files.

Default names for active and rotated log files are shown in Table 3-2.

Table 3-2 Rotated Log Files

Log	Active Log File	Rotated Log File
Access log	access_log	access_log.*unix-time* where *unix-time* is seconds since 01/01/1971
Error log	error_log	error_log.*unix-time* where *unix-time* is seconds since 01/01/1971

Note

- You can rotate only error-log and access-log files that are written by the SIP proxy server (sipd).
- With periodic log rotation, new log files are not created unless traffic or logging occurs. If there is nothing to write, new log files are not created.

Caution As needed, periodically remove old log files from the local hard disk. This prevents logs from growing until they use up the entire hard disk and cause the proxy server to stop functioning properly. For information on how to do this manually, see the "Configuring Host-Specific Directives" section on page B-5.

Setting Up Debug Logs

Detailed Steps

Step 1 From the Cisco SPS main menu (see Tip for how to access), choose **Farm/Proxies > Debug and Logs**.

Step 2 In the Error Log area, do the following:

a. Click one or more debug-flag check boxes.

Choices include the following: State Machine, GKTMP, Routing, SIP TCP, Privacy, Radius, GKTMP API, Registry, SIP TLS, Parser, Number Expansion, RAS, RPMS, DBMySQL, ENUM, RAS API, and Authentication.

Note To activate overall debugging, click State Machine in addition to other relevant debugs.

b. Select a log level.

Choices include the following, in decreasing order of verbosity: debug (most verbose), info, notice, warn, error, crit, alert, and emerg (least verbose).

Note To activate overall debugging, click debug.

c. Verify or reset the path for the log file (see Table 3-1).

d. Set rotation to On or Off. If On, set a rotation interval and set rotation-interval units to seconds or megabytes.

Step 3 In the Custom Log area, do the following:

a. Verify or reset the path for the custom log file.

Note Where transactions are logged depends on whether you define a custom log file in a <VirtualHost> container. If yes, they are logged in that container. If no, they are logged here. Default is logs/access_log.

b. Set rotation to On or Off. If On, set a rotation interval and set rotation-interval units to seconds or megabytes.

Step 4 Enter settings for the following:

- Set the SIP stats log to On or Off. If On, set the SIP stats interval.
- Set the memory stats log to On or Off. If On, set the shared memory stats interval.

Note TransferLog is not provisionable from the GUI, but you can configure CustomLog to have the same functionality.

Step 5 Click **Submit**.

Step 6 As needed, reset log levels in the following files:

- Provisioning server (ps.conf) file
- SIP provisioning agent (spa.conf) file
- License manager (lm.conf) file

Note To increase log verbosity, in the "DebugLevel" line, change the parameter from LOG_ERR to LOG_DEBUG.

Step 7 As needed, make further manual log-file modifications in the SIP-directives (sipd.conf) configuration file (for information, see the "Configuring Host-Specific Directives" section on page B-5).

Step 8 Restart your proxy servers.

Tip Debugs are written to the file <ServerRoot>/logs/error_log.

How to Replace, Upgrade, or Delete a Cisco SPS License

Cisco SPS licenses are of two types: evaluation and permanent. You can replace one evaluation license with another, upgrade from an evaluation license to a permanent license, or delete a license.

Your license is delivered to you in the form of a license key—that is, a sequence of text characters that the Cisco SPS must read and validate at startup before it can run.

Tip

- To resize a column, place the cursor on the vertical line dividing column headers and drag it to a desired position. To rearrange column order, place the cursor on a header and drag it to a desired position.
- To display only specific licenses, use the search tool (field, operator, search string) at page top.
- To display all licenses, use the search tool with the search string set to *.
- To display licenses in a particular order, use the column-heading sort arrows.

Note You can access your licenses either using the GUI-based provisioning system or manually. The former is presented below. The latter is presented in Appendix B, "Manual Configuration."

Detailed Steps

Step 1 Access the Cisco SPS license window as follows:

a. Go to the following (default) directory or your Windows desktop:

Linux: `/usr/local/sip/gui/`

Solaris: `/opt/sip/gui/`

b. Double-click the license GUI.

c. Enter your password (default is cspsuser) and then do either of the following:

- From the pserver, click **OK**.
- From a server other than the pserver, click **more>>**, enter the pserver host name and port number, and click **OK**.

The license window appears. The pserver host name and port number automatically reappear at login.

Step 2 To replace a license, do the following:

a. Copy the new license key, in preparation for pasting.

b. **Locate** the **license (see the tips above) and click to select it.**

c. Click **Edit**.

d. Double-click and delete the old license key.

e. Paste in the new license key.

f. Edit other fields as needed for the upgraded license.

g. Click **Submit.**

Step 3 To delete a license, do the following:

a. **Locate** the **license (see the tips above) and click to select it.**

b. Click **Delete** > **Yes**.

Troubleshooting Tips

- Do not include quotation marks around the license key.
- Be cautious if you cut and paste the license key from one operating system to another. You might introduce an incorrect end-of-line character sequence that causes the system not to recognize the key.
- License-validation messages that display (Linux) on screen or (Solaris) in the /var/log/messages file are stored in the error log (error_log) file for your later reference.

How to Back Up and Restore Cisco SPS

It is important that you back up Cisco SPS data on a regular basis so that you can recover quickly from catastrophic failures on the part of one or more servers.

The following sections describe procedures for backing up and restoring data:

- Backing Up Data, page 3-8
- Restoring Backed-Up Data, page 3-9

Backing Up Data

Prerequisites

- Make available a separate data-storage system on which to store backed-up data.
- Determine and adhere to a regular backup schedule.

Detailed Steps

Step 1 If MySQL is run for the provisioning system or subscriber features or both, save all the data to a flat file using the following command on the system where MySQL is run:

```
# mysqldump -u guest -p --databases sip > <outside_directory/file>
Enter password: <default password is "nobody">
```

Step 2 Export any registries to a computer-separated value (csv) file as described in the "How to Import and Export Bulk Routing and Registry Data" section on page 2-16.

Step 3 Export any static routes as described in the "How to Import and Export Bulk Routing and Registry Data" section on page 2-16.

Step 4 Copy the license (license.conf), persistent TCP (persistent_tcp.conf), and SIP-directives (sipd.conf) configuration files and store the copies in an alternate location.

Note If you use the GUI-based provisioning-system, do not back up the sipd.conf file. It regenerates from the information stored in MySQL.

Linux:
```
# cp /usr/local/sip/conf/license.conf <outside_directory/file>
# cp /usr/local/sip/conf/persistent_tcp.conf <outside_directory/file>
# cp /usr/local/sip/conf/sipd.conf <outside_directory/file>
```

Solaris:
```
# cp /opt/sip/conf/license.conf <outside_directory/file>
# cp /opt/sip/conf/persistent_tcp.conf <outside_directory/file>
# cp /opt/sip/conf/sipd.conf <outside_directory/file>
```

Restoring Backed-Up Data

Prerequisites

- If Cisco SPS was installed on your system when the system was delivered to you, before restoring Cisco SPS, uninstall Cisco SPS so that the system is in a known state and then reinstall it using the SPS setup (sps_setup) script. For details on uninstalling and installing Cisco SPS, refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.
- During reinstallation, when the license-key prompt appears, refer to the saved license (license.conf) file if the license key from initial installation is not readily available.

Detailed Steps

Step 1 Delete the existing sip database.

```
# mysql -u guest -p
Enter password: <default password is "nobody">
at mysql> prompt type:
    drop database sip;
    quit;
```

Step 2 Restore the previously saved sip database.

```
# mysql -u guest -p < <mysql_backup_file>
Enter password: <default password is "nobody">
```

Step 3 (GUI) Delete any old provisioning-server connection data:

```
# mysql -u guest -p
Enter password: <default password is "nobody">
at mysql> prompt type:
    use sip;
    delete from DBSubscriberTable;
    quit;
```

Step 4 Import any saved registries to shared memory from the backup file (csv format), as described in the "How to Import and Export Bulk Routing and Registry Data" section on page 2-16.

Note If you have a proxy-server farm, perform this operation on the first farm member only. If an active member already exists with a current registry database, skip this step, because members update each other automatically.

Step 5 Import any saved static routes as described above.

Step 6 Restore the following files:

- license (license.conf) file
- persistent TCP (persistent_tcp.conf) file
- SIP-directives (sipd.conf) configuration file

Linux:
```
# cp <license.conf_backup file> /usr/local/sip/conf/license.conf
# cp <persistent_tcp.conf_backup_file> /usr/local/sip/conf/persistent_tcp.conf
# cp <sipd.conf_backup_file> /usr/local/sip/conf/sipd.conf
```

Solaris:
```
# cp <license.conf_backup file> /opt/sip/conf/license.conf
# cp <persistent_tcp.conf_backup_file> /opt/sip/conf/persistent_tcp.conf
# cp <sipd.conf_backup_file> /opt/sip/conf/sipd.conf
```

Note If you used the SPS setup (sps_setup) script to install Cisco SPS, do not restore the license (license.conf) file. It is generated from the information entered during setup.

Note If you use the provisioning-system GUI, do not restore the SIP-directives (sipd.conf) configuration file. It regenerates from the information stored in MySQL at the next start, restart, or graceful restart.

Step 7 Restart Cisco SPS with the new SIP-directives (sipd.conf) configuration file.

Linux:
```
# /usr/local/sip/bin/sip graceful
```

Solaris:
```
# /opt/sip/bin/sip graceful
```

How to Restore a MySQL Database

If you enable MySQL replication when configuring your system—that is, if you use two synchronized MySQL databases—updates to either MySQL server are replicated to the other. The SPS setup (sps_setup) script configures each replicated MySQL server as both master and slave to each other. Replication then works as follows:

1. The master MySQL logs all changes.
2. The slave MySQL reads from these logs and keeps track of where it has read from last.

The location of debug information is shown in Table 3-3.

For more information about MySQL replication, refer to the MySQL website at www.mysql.com.

Note Do not attempt to write to the MySQL server simultaneously using both the GUI-based provisioning system and the sysadmin_mysql_users script. Changes made via one are not seen by the other and there is a potential to both make conflicting changes to the same user, or add the same user, in which case the first change is overwritten by the second. The same would be the case if you used two GUIs.

Tip Loss of connection during an update usually leaves databases out of sync. When the connection is back up, the databases do not automatically synchronize. To cause them to synchronize, stop and restart the MySQL that did not update. You can determine whether your databases are out of sync by running the mysql_sync_check.sh script. You can also configure SNMP to run the script every evening.

Detailed Steps

Step 1 To check for differences between the two MySQL databases, run the MySQL synchronicity check (<server_root>/bin/mysql_sync_check.sh) script (run as cron if needed).

Note For 20,000 users, this script takes about 10 seconds to run, during which it does not lock the databases from writes. A write during this time might fail if the slave database is not yet updated.

Solaris: `# /opt/sip/bin/mysql_sync_check.sh`

Linux: `# /usr/local/sip/bin/mysql_sync_check.sh`

Step 2 To restore a corrupted MySQL database (corrupted, for example, by a hard-drive crash), do one of the following:

- (Recommended) If you can tolerate a stop to call processing and can therefore stop your remaining MySQL server, follow the *Cisco SIP Proxy Server Version 2.2 Installation Guide* instructions for upgrading from a one-member to a two-member farm.

  Note Upgrading involves bringing down the remaining MySQL server. Call processing stops during this upgrade time.

- If you cannot tolerate a stop to call processing and must keep your remaining MySQL server up and running, use the following procedure:

Note The following procedure is long and complex. If possible, follow the recommended procedure above rather than this one.

1. Back up your database (see the "How to Back Up and Restore Cisco SPS" section on page 3-8).

On the Existing MySQL Server...

2. On the existing MySQL server, log in as root:

   Linux: `/usr/bin/mysql -uroot -p<root_password> sip`

   Solaris: `/opt/mysql/bin/mysql -uroot -p<root_password> sip`

3. Stop the slave process. Ignore any error messages.

   ```
   mysql> slave stop;
   ```

4. Lock the table from writes:

   ```
   mysql> flush tables with read lock;
   ```

5. In a new window, change directories:

   Linux: `cd /var/lib/mysql`

   Solaris: `cd /opt/mysql/data`

6. Tar the MySQL data:

   ```
   # tar -cf <tmp_dir>/sip.tar sip
   ```

7. In the original window, enter the following **show** command:

   ```
   mysql> show master status;
   ```

8. Make a note of the file and position information that displays and keep it in a safe place.
9. Unlock the table from writes:

   ```
   mysql> unlock tables;
   ```

On the New MySQL Server...

10. On the new MySQL server, uninstall Cisco SPS (if it is already installed) and then reinstall it (refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide*).

    ```
    rpm -i <CSPS.rpm>
    ```

    or

    ```
    pkg -d <CSPS.pkg>
    ```

11. Install whichever member is corrupted:

    ```
    # <server_root>/bin/sps_setup
    ```

12. Stop MySQL:

    ```
    # /etc/init.d/mysql stop
    ```

13. Open the my.cnf file with a text editor and remove all "master*" lines.

    ```
    # vi /etc/my.cnf
    ```

14. Change directories:

Linux: `cd /var/lib/mysql`

Solaris: `cd /opt/mysql/data`

15. Delete the current database:

```
# rm -rf sip
```

16. If the database that you stored from the existing MySQL in the steps above is not accessible from this machine, copy it to a local temporary directory.
17. Restore the database with that from the existing MySQL server:

```
# tar -xf <tmp_dir>/sip.tar
```

18. Start MySQL:

```
# /etc/init.d/mysql start
```

19. Log in to the MySQL server as root:

Linux: `/usr/bin/mysql -uroot -p<root_password> sip`

Solaris: `/opt/mysql/bin/mysql -uroot -p<root_password> sip`

20. Stop the slave process:

```
mysql> slave stop;
```

21. Update the master information using the information that you wrote down earlier:

```
mysql> change master to master_host='<EXISTING_MySQL_host>';
mysql> change master to master_user='guest';
mysql> change master to master_password='nobody';
mysql> change master to master_port=3306;
mysql> change master to master_log_file='vvs-finland-bin.002';
mysql> change master to master_log_pos=1428;
mysql> show master status;
```

22. Make a note of the file and position information and keep it in a safe place.
23. Start the slave process:

```
mysql> slave start;
```

On the Existing MySQL Server...

24. Log in to the existing MySQL server as root:

Linux: `/usr/bin/mysql -uroot -p<root_password> sip`

Solaris: `/opt/mysql/bin/mysql -uroot -p<root_password> sip`

25. Reverse master and slave, using the information that you wrote down above:

```
mysql> change master to master_log_file='vvs-iceland-bin.002';
mysql> change master to master_log_pos=73;
```

26. If you are using a different host, enter the following:

```
mysql> change master to master_host='<NEW_MySQL_host>';
```

27. Start the new slave process:

```
mysql> slave start;
```

Note Writes are blocked only during the appropriate times for the existing MySQL server. It is assumed that no writes occur to the new MySQL server until the master information is updated on the new MySQL server.

CHAPTER 4

Monitoring System Status

This chapter describes how to use the SNMP-based CIAgent tool to monitor Cisco SIP proxy server (Cisco SPS) system status.

Contents

Prerequisites

- Install CIAgent as described in the *Cisco SIP Proxy Server Version 2.2 Installation Guide*. Be sure that each server in a proxy-server farm has its own CIAgent.
- Make a note of the CIAgent location on each server. When a step instructs you to access CIAgent, go to this directory. Default is as follows:

  Linux: `/usr/local/Snmpri/CIAgent`

  Solaris: `/opt/Snmpri/CIAgent`

- Make a note of the CIAgent Dr-Web location:

  http://<*localhost, machine IP or host name running CIAgent*>:280

  (For example: http://172.16.1.1:280)

Information About CIAgent, Subagents, and Traps

A management information base (MIB) is a database of network management information that is used and maintained by a network management protocol such as Simple Network Management Protocol (SNMP) or Common Management Information Protocol (CMIP).

MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches. You can retrieve or change the value of a MIB object using SNMP or CMIP commands, usually through a GUI network management system.

Functions

CIAgent is an SNMP agent with which you can perform certain Cisco SPS operations, including the following:

- Start, stop, and gracefully restart Cisco SPS
- Monitor proxy-server log file sizes
- Monitor CPU usage
- Check memory size and disk space
- Check system status

For most of these, you can instruct CIAgent to let you know when file size, CPU usage, memory size, or disk space exceeds a certain threshold.

CIAgent has a master agent (snmpdm) that communicates with a few subagents (critagt, smagt, logagt, and more) that serve their respective MIBs. Here is the sequence that is followed:

1. You issue an SNMP request to the master agent.
2. The master agent passes the request to the appropriate subagent.
3. The subagent retrieves or sets a particular MIB object value and sends a response back to the master agent.
4. The master agent passes the response back to you.

Responses might be issued immediately, as for system status, or they might be issued when certain events occur, such as when Cisco SPS goes up or goes down, when the size of a log file exceeds a specified limit, or when the CPU load rises above or falls below a specified threshold.

Architecture

CIAgent runs as a set of processes as root on the same system that runs Cisco SPS.

Each server in a proxy-server farm should have its own CIAgent. A system therefore has multiple CIAgents, each of which controls and monitors a particular farm member.

MIB files are located in the CIAgent installation directory under the mibs directory. They are text files with the extension .my, and are a good source for learning what a MIB or subagent provides.

Standard MIBs and the subagents that serve them are listed in Table 4-1.

Table 4-1 MIBs and Subagents

MIB	Subagent	MIB Function
CIAgent MIBs		
CRITAPP-MIB (critagt)	Critical application	Starts and stops Cisco SPS

Table 4-1 MIBs and Subagents (continued)

MIB	Subagent	MIB Function
LOG-MIB (logagt)	Log file	Monitors error-log and access-log file sizes and sends traps when thresholds are exceeded
Standard MIBs		
DISMAN-SCRIPT-MIB (smagt)	Script MIB	Gracefully restarts Cisco SPS
DISMAN-EVENT-MIB (eventagt)	Event MIB	Monitors CPU load by setting rising and falling thresholds; sends traps when thresholds are exceeded
HOST-RESOURCES-MIB (hostagt)	Host resources	Checks CPU usage, memory size, and disk space
RFC1213-MIB (mib2agt)	MIB-II	Checks link status
SYSAPPL-MIB (sappagt)	System applications	Checks what applications are installed and running on the system

How to Set Up and Use CIAgent

This section contains the following:

Stopping and Restarting CIAgent Manually

Detailed Steps

Step 1 Access CIAgent.

Step 2 To stop CIAgent, use the **stop** command:

```
# csps_ciagent_ctl stop
```

or

```
# ciagent -fullstop
```

Step 3 To restart CIAgent, use the **start** command:

```
# csps_ciagent_ctl start
```

or

```
# ciagent -fullstart
```

Creating a CIAgent Dr-Web User ID

Detailed Steps

Step 1 Access CIAgent.

Step 2 Open the SNMP directives (/etc/srconf/agt/snmpd.cnf) file.

Step 3 Add the following line to the end of the file:

```
httpUserNameEntry <your-login-name> SystemAdmin - nonVolatile <your-login-password>
```

Step 4 If security is an issue, remove the comment markers of, or remove entirely, the other httpUserNameEntries. Doing so limits access to just the system administrator.

Step 5 Change the file permission to read-only by root.

Step 6 Save and close the file.

Step 7 Stop and restart CIAgent.

Starting and Stopping Cisco SPS from CIAgent

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 Click **Critical Application Monitor**.

Step 3 Do any of the following:

- To start Cisco SPS, set the sipd intended operation status to up.
- To stop Cisco SPS, set the sipd intended operation status to down.

Note The following CIAgent utility commands **getone** and **setany** are available in (Linux) /usr/local/Snmpri/CIAgent/bin or (Solaris) /opt/Snmpri/CIAgent/bin. You can use them to get or set object values in MIBs supported by CIAgent. Set the variable <*location*> to the local host or the IP address or host name of the system that runs CIAgent.

- To check Cisco SPS running status, use the following command:

```
# getone -v2c <location> cspsAdmin critAppOperStatus.1
```

- To gracefully restart Cisco SPS, use the following command:

```
# setany -v2c <location> cspsAdmin smLaunchStart.1.67.1.68 0
```

Step 4 If Cisco SPS should be running at this point, reload the page and check the operational status to so verify.

How to Configure Subagents

The section contains the following:

Configuring CIAgent for Subagent Use

If you used the Cisco SPS CIAgent install (csps_ciagent_install) script to install CIAgent, a preset configuration is included in the installation. This configuration is extendable.

To send SNMP requests to CIAgent, use the Dr-Web interface, command-line CIAgent utility, or any other SNMP product. If you use the command-line CIAgent utility, a default community string, cspsAdmin, is provided in the SNMP directives (snmpd.cnf) file.

CIAgent supports SNMP versions 1, 2c, and 3. The examples in this document and the readme file use Version 2c to show how to retrieve and set certain MIB objects. If security is a major concern, use Version 3.

Note

- You can always remove or add logins in the SNMP directives (/etc/srconf/agt/snmpd.cnf) file. For more information, refer to the CIAgent online manual chapter on the emanate master agent.
- For more information on CIAgent and subagent configurations and how to set up users and passwords, refer to the CIAgent Dr-Web online manual.

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 At the CIAgent home page, do the following:

a. Locate, in the CIAgent online manual, information on how to configure subagents.

b. Click the listed subagents to see a sample list of Cisco SPS configurations. Table 4-1 shows a list of sample subagents.

Step 3 As needed, change the following default settings as instructed in the CIAgent Dr-Web online manual:

- sipd's intended operation status—Initialized to down. To start Cisco SPS, change to up.
- Start and stop commands—Initialized into the critical-applications MIB with appropriate path to the Cisco SPS installation directory.
- Graceful restart command—Initialized into the script MIB with appropriate path to the Cisco SPS installation directory.
- Cisco SPS log files monitored by CIAgent—Initialized to error_log and access_log each with a maximum size of 5 MB. If either one grows bigger than 5 MB, a trap is sent.

- CPU average load over the last minute, monitored by the event subagent—Rising threshold is initialized to 75 percent; falling threshold is initialized to 20 percent. When the CPU average load over the last minute falls outside that range, a trap is sent.

Note

- You cannot configure the script MIB subagent through this home page.
- MIB files are located in the CIAgent installation directory under mibs.
- Utility programs in the CIAgent bin directory that can interact with MIB objects include setany, getone, getmany, and traprcv.

Configuring the Critical Application Subagent (critagt)

You can configure critagt to start and stop Cisco SPS. You can also add an entry for sipd, provide the **Start** and **Terminate** commands, and set desired options.

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 Open the critical application subagent (/etc/srconf/agt/critagt.cnf) file with a text editor.

Step 3 Add a critAppProcEntry line for sipd, SIP provisioning agent (spa), provisioning server (pserver), and the license manager (licenseMgr).

Example

The following example is for Linux. For Solaris, change the path to sip from /usr/local/sip/bin/sip to /opt/sip/bin/sip.

```
# Entry type: critAppProcEntry
# Entry format: integer - index number (continuous positive integer)
#   octetString - process name (real process name)
#   octetString - start command (string of characters)
#   octetString - stop command (string of characters)
#   integer - intended operation status (up(1), down(2))
#   integer - restart on exit (true or false)
#   integer - restart interval in centisecond
#   integer - send trap on exit (true or false)
#   integer - send trap on start (true or false)
#   integer - find process on startup (true or false)
critAppProcEntry 1 sipd \
    "/usr/local/sip/bin/sip start" \
    "/usr/local/sip/bin/sip stop" 2 false 3000 \
    true true true

critAppProcEntry 2 spa \
    "/usr/local/sip/bin/sip start" \
    "/usr/local/sip/bin/sip stop" 2 false 3000 \
    true true true

critAppProcEntry 3 pserver \
    "/usr/local/sip/bin/sip start" \
    "/usr/local/sip/bin/sip stop" 2 false 3000 \
```

```
    true true true

critAppProcEntry 4 licenseMgr \
    "/usr/local/sip/bin/sip start" \
    "/usr/local/sip/bin/sip stop" 2 false 3000 \
    true true true

critAppProcEntry  5 mysqld \
    "/etc/init.d/mysql start" \
    "/etc/init.d/mysql stop" 2 false 3000 \
    true true true

# Entry type: critAppTrapWhenNotAllRunning
# Entry format: integer - send trap when some process is down
critAppTrapWhenNotAllRunning false
# Entry type: critAppTrapWhenAllRunning
# Entry format: integer - send trap when all are running
critAppTrapWhenAllRunning false
```

Step 4 Save and close the file.

Step 5 Stop and restart critagt to activate the new configuration.

Linux:
```
# ps -ef|grep critagt
# kill -9 <critagt's PID>
# cd /usr/local/Snmpri/CIAgent/bin
# ./critagt &
```

Solaris:
```
# ps -ef|grep critagt
# kill -9 <critagt's PID>
# cd /opt/Snmpri/CIAgent/bin
# ./critagt &
```

Step 6 As needed, start or stop Cisco SPS using Dr-Web or the following commands:

Note **setany** is an SNMP utility program that you run from CIAgent. You can achieve the same functionality with any other SNMP **set** command. Refer to the MIB files (located in the CIAgent installation directory under mibs) for the object information before setting it.

Note In the following commands, the variable <*location*> indicates the local host name or the IP address or host name of the system that runs CIAgent.

- To start Cisco SPS:

  ```
  # setany -v2c <location> cspsAdmin critAppAdminStatus.1 1
  ```

- To stop Cisco SPS:

  ```
  # setany -v2c <location> cspsAdmin critAppAdminStatus.1 2
  ```

- To learn the Cisco SPS running status:

  ```
  # getone -v2c <location> cspsAdmin critAppOperStatus.1
  ```

- To learn about the Cisco SPS **start** command:

  ```
  # getone -v2c <location> cspsAdmin critAppStartCommand.1
  ```

- To learn about the Cisco SPS **stop** command:

  ```
  # getone -v2c <location> cspsAdmin critAppTerminateCommand.1
  ```

Configuring the Script Subagent (smagt) to Gracefully Restart Cisco SPS

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 In the CIAgent installation-bin directory, open the MIB-population (smPopScript) script with a text editor.

Step 3 Modify the following variables for the environment.

```
Agent="localhost"
Version="-v2c"
User="cspsAdmin"
AuthPassword=""
PrivPassword=""
```

Step 4 Modify the following line as shown to cause the Cisco SPS installation path to do a graceful restart.

Linux:
```
setany $Version $Agent $User \ smCodeRowStatus.$SOwner.$SName.1 createAndGo \
    smCodeText.$SOwner.$SName.1 -D "exec(\"/usr/local/sip/bin/sip graceful\");"
```

Solaris:
```
setany $Version $Agent $User \ smCodeRowStatus.$SOwner.$SName.1 createAndGo \
    smCodeText.$SOwner.$SName.1 -D "exec(\"/opt/sip/bin/sip graceful\");"
```

Step 5 Save and close the file.

Step 6 Change the file permission to be executable by root.

Step 7 Run the MIB-population script.

```
# ./smPopScript
```

Step 8 Do one of the following:

- If Cisco SPS is already running, gracefully restart CIAgent.

  Note In the following command, the variable *<location>* indicates the local host or the IP address or host name of the system that runs CIAgent.

  ```
  # setany -v2c <location> cspsAdmin smLaunchStart.1.67.1.68 0
  ```

- Otherwise, stop and restart CIAgent by running a script to populate the script MIB subagent.

  Note To avoid doing this every time, add a call to the script in the ciagent script file (CIAgent installation directory). See the following example in the ciagent script file.

  ```
  ./critagt
  ./mib2agt
  ./eventagt &
  ./fsagt &
  ./hostagt &
  ./htmlagt
  ./logagt
  ```

```
./sappagt &
./smagt
sleep 5
./smPopScript
```

Note This example assumes that the MIB-population (smPopScript) script is modified and exists in the CIAgent bin directory. The **sleep 5** command runs smPopScript after sleeping 5 seconds, which allows smagt to fully start. An alternative is to use the customized start/stop csps_ciagent_ctl script (Linux: in /usr/local/sip/ciagent; Solaris: in /opt/sip/ciagent). Copy it to your CIAgent installation-bin directory (Linux: /usr/local/Snmpri/CIAgent/bin; Solaris: /opt/Snmpri/CIAgent/bin) and give it execution permission. The customized script starts CIAgent and then calls smPopScript.

Configuring the Log-File Subagent (logagt) File to Monitor Log Sizes

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 Open the log-file subagent (/etc/srconf/agt/logagt.cnf) file with a text editor.

Step 3 Make changes as needed.

Example

The following example is for Linux. For Solaris, change the path for error_log and access_log to /opt/sip/logs/error_log and /opt/sip/logs/access_log respectively.

```
# Entry type: siLogGlobalPollInterval
# Entry format: integer
siLogGlobalPollInterval 60
# Entry type: siLogEntry
# Entry format: integer - index number
# octetString - description of the file to be monitored
# octetString - full path to the file
# octetString - regular expression to match in the file
# integer - leave it as is
# integer - character position to start matching
# integer - character position to stop matching
# integer - number of matches found so far
# octetString - command to run on match
# integer - send trap on match (yes(1), no(2))
# integer - current size of the file in bytes
# integer - maximum file size as threshold
# octetString - command to run when maximum size reached
# integer - send trap on maximum size (yes(1), no(2))
# integer - polling interval in seconds
# integer - leave it as is
# octetString - file owner
# integer - leave it as is
siLogEntry 1 "CSPS error log" \
    /usr/local/sip/logs/error_log - 2 0 0 0 - 2 \
    316687 5000000 - 1 10 2 csps 1 316687 "tent-Length: 0\r\n\r\n\n\n"

siLogEntry 2 "CSPS access log" \
    /usr/local/sip/logs/access_log - 2 0 0 0 - 2 \
```

```
316687 5000000 - 1 10 2 csps 1 316687 "tent-Length: 0\r\n\r\n\n\n"
```

Step 4 Save and close the file.

Step 5 Stop and restart logagt to activate the new configuration.

Linux:
```
# ps -ef|grep logagt
# kill -9 <logagt's PID>
# cd /usr/local/Snmpri/CIAgent/bin
# ./logagt &
```

Solaris:
```
# ps -ef|grep logagt
# kill -9 <logagt's PID>
# cd /opt/Snmpri/CIAgent/bin
# ./logagt &
```

Step 6 View your current log file size from Dr-Web, or use the following commands to retrieve it.

Note In the following command, the variable <*location*> indicates the local host or the IP address or host name of the system that runs CIAgent.

```
# getone -v2c <location> cspsAdmin siLogSize.1

# getone -v2c <location> cspsAdmin siLogSize.2
```

Note In the preset configurations, siLogSize.1 refers to log file size for the file at index 1 (error_log). siLogSize.2 refers to access_log.

Configuring the Event MIB Subagent to Monitor CPU Usage

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 Do one of the following:

- Configure a Trigger-Event-Notification set for the hrProcessorLoad object in Host Resources MIB for CPU rising and falling thresholds.
- Modify the following sample event agent (eventagt.cnf) file accordingly.

  Example

  In this example, the rising threshold is set to 75%, the falling threshold to 20%, and the polling interval to 5 seconds.

```
# Entry type: mteResourceSampleMinimum
# Entry format:
#   mteResourceSampleMinimum (integer) mteResourceSampleMinimum 1
# Entry type: mteResourceSampleInstanceMaximum
# Entry format:
#   mteResourceSampleInstanceMaximum (unsigned)
mteResourceSampleInstanceMaximum 0u
# Entry type: mteTriggerEntry
# Entry format:
```

```
#   mteOwner (text)
#   mteTriggerName (text)
#   mteTriggerComment (text)
#   mteTriggerTest (bits)
#   mteTriggerSampleType (integer)
#   mteTriggerValueID (ObjectID)
#   mteTriggerValueIDWildcard (integer)
#   mteTriggerTargetTag (text)
#   mteTriggerContextName (text)
#   mteTriggerContextNameWildcard (integer)
#   mteTriggerFrequency (unsigned)
#   mteTriggerObjectsOwner (text)
#   mteTriggerObjects (text)
#   mteTriggerEnabled (integer)
#   mteTriggerEntryStatus (integer) mteTriggerEntry 61 loadTrigger " " 20 1
    iso.3.6.1.2.1.25.3.3.1.2.1 2 - - 2 \
    5u - - 1 1
# Entry type: mteTriggerDeltaEntry
# Entry format:
#   mteTriggerDeltaDiscontinuityID (ObjectID)
#   mteTriggerDeltaDiscontinuityIDWildcard (integer)
#   mteTriggerDeltaDiscontinuityIDType (integer)
#   mteOwner (text)
#   mteTriggerName (text)
# Entry type: mteTriggerExistenceEntry
# Entry format:
#   mteTriggerExistenceTest (bits)
#   mteTriggerExistenceStartup (bits)
#   mteTriggerExistenceObjectsOwner (text)
#   mteTriggerExistenceObjects (text)
#   mteTriggerExistenceEventOwner (text)
#   mteTriggerExistenceEvent (text)
#   mteOwner (text)
#   mteTriggerName (text)
# Entry type: mteTriggerBooleanEntry
# Entry format:
#   mteTriggerBooleanComparison (integer)
#   mteTriggerBooleanValue (integer)
#   mteTriggerBooleanStartup (integer)
#   mteTriggerBooleanObjectsOwner (text)
#   mteTriggerBooleanObjects (text)
#   mteTriggerBooleanEventOwner (text)
#   mteTriggerBooleanEvent (text)
#   mteOwner (text)
#   mteTriggerName (text)
# Entry type: mteTriggerThresholdEntry
# Entry format:
#   mteTriggerThresholdStartup (integer)
#   mteTriggerThresholdRising (integer)
#   mteTriggerThresholdFalling (integer)
#   mteTriggerThresholdDeltaRising (integer)
#   mteTriggerThresholdDeltaFalling (integer)
#   mteTriggerThresholdObjectsOwner (text)
#   mteTriggerThresholdObjects (text)
#   mteTriggerThresholdRisingEventOwner (text)
#   mteTriggerThresholdRisingEvent (text)
#   mteTriggerThresholdFallingEventOwner (text)
#   mteTriggerThresholdFallingEvent (text)
#   mteTriggerThresholdDeltaRisingEventOwner (text)
#   mteTriggerThresholdDeltaRisingEvent (text)
#   mteTriggerThresholdDeltaFallingEventOwner (text)
#   mteTriggerThresholdDeltaFallingEvent (text)
#   mteOwner (text)
#   mteTriggerName (text)
```

```
    mteTriggerThresholdEntry 1 75 20 0 0 - - 61 rising 61 falling - - - - 61 \
    loadTrigger
# Entry type: mteObjectsEntry
# Entry format:
#   mteObjectsName (text)
#   mteObjectsIndex (unsigned)
#   mteObjectsID (ObjectID)
#   mteObjectsIDWildcard (integer)
#   mteObjectsEntryStatus (integer)
#   mteOwner (text)
    mteObjectsEntry loadValue 1u iso.3.6.1.2.1.25.3.3.1.2.1 2 1 61
# Entry type: mteEventEntry
# Entry format:
#   mteEventName (text)
#   mteEventComment (text)
#   mteEventActions (bits)
#   mteEventEnabled (integer)
#   mteEventEntryStatus (integer)
#   mteOwner (text)
    mteEventEntry falling "Falling threshold trap" 80 1 1 61
    mteEventEntry rising "Rising threshold trap" 80 1 1 61
# Entry type: mteEventNotificationEntry
# Entry format:
#   mteEventNotification (ObjectID)
#   mteEventNotificationObjectsOwner (text)
#   mteEventNotificationObjects (text)
#   mteOwner (text)
#   mteEventName (text)
    mteEventNotificationEntry 0.0 61 loadValue 61 falling
    mteEventNotificationEntry 0.0 61 loadValue 61 rising
#   Entry type: mteEventSetEntry
#   Entry format:
#   mteEventSetObject (ObjectID)
#   mteEventSetObjectWildcard (integer)
#   mteEventSetValue (integer)
#   mteEventSetTargetTag (text)
#   mteEventSetContextName (text)
#   mteEventSetContextNameWildcard (integer)
#   mteOwner (text)
#   mteEventName (text)
```

Note The hrProcessorLoad object represents the average CPU usage over the last minute. (This is not the same as the CPU usage output in the Unix program "top," which shows CPU usage in the sampling moment.) The hrProcessorLoad value rises and drops slowly, because it is an average value over a minute.

Step 3 Save and close the file.

Step 4 Stop and restart eventag to activate the new configuration.

Linux:
```
# ps -ef|grep eventagt
# kill -9 <eventag's PID>
# cd /usr/local/Snmpri/CIAgent/bin
# ./eventagt &
```

Solaris:
```
# ps -ef|grep eventagt
# kill -9 <eventag's PID>
# cd /opt/Snmpri/CIAgent/bin
# ./eventagt &
```

Step 5 Check the current hrProcessorLoad object value.

Note In the following command, the variable <*location*> indicates the local host or the IP address or host name of the system that runs CIAgent.

```
# getone -v2c <location> cspsAdmin hrProcessorLoad.1
```

How to Configure Traps

This section contains the following:

Configuring SNMP and Trap Target Addresses

Instructions on SNMP and trap target address configuration are in the CIAgent online manual (see Table 4-2).

Table 4-2 Location of Instructions for SNMP and Trap Target Address Configuration

Topic	CIAgent Online Manual Section
Add or configure SNMP v2c community strings for Cisco SPS administrators	emanate master agent
Configure trap target addresses	
Modify the system environment	SNMP Community/userName configuration and trap configuration

Configuration Examples

The following are examples from the SNMP directives (/etc/srconf/agt/snmpd.cnf) file:

- To add community string cspsAdmin with security level proxySec:

```
snmpCommunityEntry t0000001 cspsAdmin proxySec localSnmpID - - nonVolatile
```

- To add a new group, proxyGroup, with snmpv2c access permissions:

```
vacmAccessEntry proxyGroup - snmpv2c noAuthNoPriv exact All All All nonVolatile
```

- To associate security level proxySec with proxyGroup:

```
vacmSecurityToGroupEntry snmpv2c proxySec proxyGroup nonVolatile
```

- To specify an IP address, 172.17.140.131, to which to send traps:

```
snmpTargetAddrEntry 40 snmpUDPDomain 172.17.140.131:0 100 3 Console v2cExampleParams
nonVolatile 255.255.255.255:0
```

- To specify a community string, cspsAdmin, for use in v2c traps:

```
snmpTargetParamsEntry v2cExampleParams 1 snmpv2c proxySec noAuthNoPriv nonVolatile
```

Configuring Trap Sinks for CIAgent Traps

Detailed Steps

Step 1 Access CIAgent.

Step 2 Run the CIAgent utility traprcv as root.

Step 3 Configure the SNMP directives (snmpd.cnf) file where the trap sink is located.

Step 4 Restart CIAgent.

Note Usually the loopback address (127.0.0.1) is one of the default trap sink addresses. When a trap-triggering event occurs, such as Cisco SPS going up or down, a trap message appears in the traprcv program window. A trap message also appears when log files exceed limits and to any CPU load that is over or under threshold.

How to Monitor System Status and Components

This section contains the following:

Changing Cisco SPS System Information

Detailed Steps

Step 1 Access CIAgent.

Step 2 Open the SNMP directives (/etc/srconf/agt/snmpd.cnf) file.

Step 3 Change the default information in any of the following:

- sysLocation—Physical location of this managed system (for example, 2nd rack, 3rd floor)
- sysContact—Contact person for this managed system
- sysName—fully qualified domain name (FQDN) of this managed system

Step 4 Save and close the file.

Step 5 Stop and restart CIAgent.

Checking System Status and Components

Detailed Steps

Step 1 Log in to CIAgent Dr-Web.

Step 2 Click **System Applications Monitor**.

Step 3 Do any of the following.

Note In the following commands, the variable <*location*> indicates the local host or the IP address or host name of the system that runs CIAgent.

- To check memory size, use the following **getmany** command:

```
# getmany -v2c <location> cspsAdmin hrMemorySize
```

- To check disk space, use the following **getmany** command:

```
# getmany -v2c <location> cspsAdmin hrStorageEntry
```

Note Refer to the HrStorageEntry in the Host Resources MIB for more detail.

- To check link status from MIB-2, use the following **getmany** command:

```
# getmany -v2c <location> cspsAdmin ifTable
```

Link Description Output Example

```
ifDescr.1 = lo0
ifDescr.2 = hme0
```

Link Up/Down Status Output Example

```
ifOperStatus.1 = up(1)
ifOperStatus.2 = up(1)
```

Link Type Output Example

```
ifType.1 = softwareLoopback(24)
ifType.2 = ethernet_csmacd(6)
```

Link MTU Output Example

```
ifMtu.1 = 8232
ifMtu.2 = 1500
```

Note Refer to MIB-2 for more detail.

- To check system components, use the following **getmany** commands.

Note In the following commands, the variable <*location*> indicates the local host or the IP address or host name of the system that runs CIAgent.

```
# getmany -v2c <location> cspsAdmin sysApplInstalled

# getmany -v2c <location> cspsAdmin sysAppRun
```

```
# getmany -v2c <location> cspsAdmin sysApplInstallPkgProductName

# getmany -v2c <location> cspsAdmin sysApplInstallPkgDate

# getmany -v2c <location> cspsAdmin sysApplInstallPkgLocation
```

Step 4 Reload the page and check the operational status to verify that Cisco SPS is running.

CHAPTER 5

Troubleshooting

This chapter contains a list of trouble symptoms and error messages that might arise as you configure and use Cisco SIP proxy server (Cisco SPS), with possible causes and recommended actions. It also describes an SPS application.

Contents

Note

- The same symptom appears under all relevant headings.
- Also helpful for troubleshooting problems are the call flows in Appendix E, "SIP Call-Flow Scenarios."

Setup

Symptom Cisco SPS does not install or run.

Possible Cause Cisco SPS runs only on the following operating systems: Red Hat Enterprise Linux AS, ES, or WS 3.0; or Sun Solaris 8. It is critical that you use only a supported operating system. Cisco SPS does not run on other than these.

Recommended Action Check your operating system and reinstall if necessary.

Symptom When you attempt to run the SPS setup (sps_setup) script after successful installation of Cisco SPS on a Solaris machine, Cisco SPS cannot find the script.

Possible Cause Bash needs to be installed.

Recommended Action Install Bash.

Symptom System messages warn you of invalid URL or subscriber-ID syntax.

Recommended Action Check the call-forwarding destination (dest_url_cfna) or other URL and reenter it with valid syntax. The following are examples:

```
sip:5000@cisco.com
sip:bob@192.168.1.2
```

Recommended Action Check the subscriber ID (user_id) and reenter it with valid syntax.

Symptom Endpoints do not register. Error messages suggest that a proxy-server member is not in the farm.

Possible Cause The system cannot resolve its hostname.

Recommended Action Review, in the *Cisco SIP Proxy Server Version 2.2 Installation Guide* section on prerequisites, information about hostname resolution.

Symptom Due to typing errors and subsequent corrections during installation, the SPS GUI now comes up with two IP addresses.

Recommended Action The IP address is stored in several places, one of which is a MySQL database. On Cisco SPS 2.2, rerun the sps_setup script to change the IP address.

Startup

Symptom Cisco SPS does not start.

Recommended Action

1. Verify that the ./sip directory has the proper read-write permissions so that you can start sipd.
2. Verify that an old version of SIP proxy server (sipd) is not still running (enter **ps -ef | grep sps**). If necessary, kill the old-version processes.
3. Verify that Cisco SPS can use Domain Name System (DNS) to resolve its hostname.
4. Verify that your system has enough shared memory (at least the amount specified during installation).

Symptom System messages warn that you have an invalid license.

Possible Cause Your license may have expired. You can check this in either of two ways:

- Start the license GUI, log in, and check the Status field of the license. It should say something like this:

```
Level:        Evaluation, valid until GMT: Fri Dec  3 21:02:21 2004
  MajorVersion: 2, matches software version, 2.2.x.x
  Type:         Infrastructure
  TRIP:         Disabled
```

- Use the sysadmin_lic_val utility:

```
/usr/local/sip/bin/sysadmin_lic_val -l <your_license>
```

Recommended Action Renew your license.

Symptom Your Linux system boots and displays the Red Hat background desktop with the Cisco SPS login screen, but does not allow you to log into SPS.

Possible Cause You may be using the wrong Linux root password.

Recommended Action Try booting in single-user mode, as described on the Red Hat Linux website at http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/s1-rescuemode-booting-single.html.

Symptom The SIP proxy server (sipd) process stops and cannot be restarted. Or, when you start the server, the provisioning GUI works fine and then gets blocked after an update (particularly an update of a subscriber profile) or after waiting for awhile.

Possible Cause You might have changed the IP address for one of your proxy servers. The new information displays in the GUI, but the old information remains in MySQL in addition to the new information, causing an IP address mismatch between what you set with the GUI (which updates sipd.conf) and what appears in the spa.conf file. (The GUI changes only certain things such as

sipd.conf, subscriber data, static routes, and static registries; it does not change other things such as spa.conf, ps.conf, and dynamic registries.) The provisioning server (pserver) tries to contact a SIP provisioning agent (spa) that no longer exists, and then hangs.

Recommended Action Check the SPS setup log (sps_setup_log) to see which IP address you set up during installation. Use the **ifconfig -a** command to see what the IP address is now. If a mismatch exists, update both the GUI and the <ServerRoot>/sip/conf/spa.conf file and do a graceful restart.

Recommended Action Check that any directives that contain IP addresses are set correctly. As necessary, modify them with the GUI. IP addresses appear in the following places:

- sipd.conf farm directives
- sipd.conf MySQL host directive on the secondary server
- spa.conf pointer to PServerHost on the secondary server
- spa.conf (if your system runs on a server with two NIC cards)
- The following MySQL tables: Seeding, License, TempLicense, and DBSubscriberTable

Symptom When you start the provisioning server (pserver) and enter the default password, Cisco SPS rejects you.

Recommended Action

1. Verify that pserver is running.
2. Verify that you are running the GUI-based provisioning system on the same box where pserver is running (use the **ps -ef | grep pserver** command).
3. Check the ./logs/pserver_log file for helpful information. If there is none, change the log level of pserver by editing ./conf/ps.conf. Change the log level to LOG_DEBUG and then perform a ./sip restart. Afterwards, try opening the GUI again and see if more useful debugging information appears in the pserver_log file.

Symptom Cisco SPS takes a long time to start.

Possible Cause A DNS is used before local files. Cisco SPS times out attempting to resolve DNS.

Recommended Action To override this behavior, create /etc/irs.conf:

```
##############################################################
# /etc/irs.conf
##############################################################
# Get password entries from local file
passwd local
# Build group membership from local file
group local
# Services comes from just the local file
services local
protocols local
# Hosts comes first from local, failing that, DNS
hosts local continue
hosts dns
# Networks comes first from the local file, and failing # that the, irp daemon
 networks local continue
 networks irp
netgroup local
```

Symptom The csps_provision provisioning application does not start.

Recommended Action Verify that MySQL is up and running (use the **ps -ef | grep mysql** command). If command output indicates that MySQL is running, do the following:

1. Raise the debug level on the GUI and pserver by going to the sip/conf/www directory and editing the log configuration (logConfig) file to uncomment the following line:

   ```
   log4j.category.org.vovida.prov.psLib.HttpPSConnection=DEBUG
   ```

2. Change the log level in sip/conf/ps.conf to LOG_DEBUG.
3. Perform a sip restart.
4. Run the Cisco SPS script and check the contents of GUI and pserver logs for helpful information.

Hostname Resolution

Symptom Cisco SPS does not start.

Recommended Action

1. Verify that the ./sip directory has the proper read-write permissions so that you can start sipd.
2. Verify that an old version of SIP proxy server (sipd) is not still running (enter **ps -ef | grep sps**). If necessary, kill the old-version processes.
3. Verify that Cisco SPS can use Domain Name System (DNS) to resolve its hostname.
4. Verify that your system has enough shared memory (at least the amount specified during installation).

Symptom Cisco SPS takes a long time to start.

Possible Cause A DNS is used before local files. Cisco SPS times out attempting to resolve DNS.

Recommended Action To override this behavior, create /etc/irs.conf:

```
##############################################################
# /etc/irs.conf
##############################################################
# Get password entries from local file
passwd local
# Build group membership from local file
group local
# Services comes from just the local file
services local
protocols local
# Hosts comes first from local, failing that, DNS
hosts local continue
hosts dns
# Networks comes first from the local file, and failing # that the, irp daemon
 networks local continue
 networks irp
netgroup local
```

Symptom Cisco SPS routes calls improperly.

Recommended Action

- Verify that num-expansion statements, if present, are configured correctly.
- Verify that the various translation mechanisms (Call Forward Unconditional, Static Registry, Dynamic Registry, ENUM, GKTMP) are configured correctly and are populated correctly.
- Verify that the Static Routing table shows correct routes.
- Verify that the DNS server is provisioned correctly for DNS NAPTR, DNS SRV, and DNS A records of the devices to be routed.
- Check the error_log to see what errors are reported (example: bad SIP messages, process errors).

Symptom Cisco SPS static domain routes seem unable to use raw IP addresses.

Possible Cause Cisco SPS static domain (IP-based) routes do not support raw IP addresses.

For example, say that you configure a route to `.com.ibm.*` with a next hop of `proxy-1.isp.com`. A request is received for A.B.C.D that happens to be a host in the IBM domain. SPS forwards directly to A.B.C.D. It does a reverse DNS lookup and forwards to `proxy-1.isp.com` only if the Request URI contains `ibm.com` as in the following examples:

```
INVITE sip:user@us.ibm.com
INVITE sip:user@bldg1.sjc.ca.us.ibm.com
```

Symptom Endpoints do not register. Error messages suggest that a proxy-server member is not in the farm.

Possible Cause The system cannot resolve its hostname.

Recommended Action Review, in the *Cisco SIP Proxy Server Version 2.2 Installation Guide* section on prerequisites, information about hostname resolution.

Symptom You experience a delay of several seconds from when Cisco SPS sends a 100 trying message to when it processes an INVITE message. If you disable the access-control list (ACL), the delay goes away.

Possible Cause The delay might be caused by the Cisco SPS trying to match the caller's IP address with a hostname listed in the ACL.

Recommended Action Use IP addresses rather than hostnames in the ACL.

Connections

Symptom You cannot connect to your local MySQL server (Error 2002).

Possible Cause The MySQL database is not installed.

Recommended Action Install the database before running the tool.

Symptom The GUI-based provisioning system cannot connect to the provisioning server (pserver).

Recommended Action Change the debug level in <ServerRoot>/sip/conf/ps.conf to LOG_DEBUG. Then do the following.

- Verify that the pserver process is running. If it is not, restart it with ./sip restart.
- Verify that the pserver port is not used by another process.
- Verify that your session has not timed out. Log back in to the GUI and reauthenticate.
- Verify that the database information in ps.conf is correct. It should look like the following.

```
############################################################
#  Provisioning Server (ps) Configuration File
############################################################

DebugLevel              string          LOG_ERR
LogFilename             string          <ServerRoot>/logs/pserver_log
LocalPort               int             26005
DaemonMode              bool            True
Authentication          bool            True
UserAccountName         string          csps
GroupAccountName        string          csps
SIPDFile                string          <ServerRoot>/conf/sipd.conf
#
#DB info
#

#MySql info sample
database                string          sip
primaryDBHost           string          192.168.1.2
secondaryDBHost         string          192.168.1.3
user                    string          guest
password                string          nobody
```

Symptom The provisioning server (pserver) cannot connect to MySQL.

Recommended Action

1. Manually attempt to connect to MySQL:

```
Login:  mysql –h<host> -u<user> -p<password>
show databases;
use sip; (sip should be listed as an option)
show tables;
select * from <table>;
```

2. If you can log in manually, verify that the correct MySQL IP address is configured.
3. Verify the permission (that is, the MySQL password).

Symptom SIP provisioning agent (spa) does not receive updates from the provisioning server (pserver).

Recommended Action

1. Check error logs for any issues.
2. Log in to MySQL manually to confirm that the changes are present.
3. If changes are in MySQL, do the following:

   a. Change the log level in <ServerRoot>/sip/conf/spa.conf to LOG_DEBUG.

   b. Stop and restart spa.

   c. Tail –f spa_log.

   d.Verify that spa receives the update.

   e. If spa receives the update but problems persist, contact the Cisco Technical Assistance Center.
4. If changes are not in MySQL, verify that the tables are set up correctly and that the data is not corrupted.

Symptom SIP provisioning agent (spa) cannot write to the sipd.conf file.

Possible Cause The sipd.conf file may have been modified by a different user (such as root) so that, when spa tries to write to the file, it lacks the correct permissions.

Recommended Action

1. Save the sipd.conf file under a different name.
2. Use the GUI-based provisioning system to apply the changes again so that spa writes a new sipd.conf file.
3. Restart Cisco SPS so that it picks up the changes.

Symptom You cannot access MySQL (Error 1045).

Possible Cause Your MySQL username and password are invalid or have insufficient permission to access the database.

Recommended Action Enter the correct or properly enabled username and password. If you have forgotten your password, assign a new one.

Symptom You experience a delay of several seconds from when Cisco SPS sends a 100 trying message to when it processes an INVITE message. If you disable the access-control list (ACL), the delay goes away.

Possible Cause The delay might be caused by the Cisco SPS trying to match the caller's IP address with a hostname listed in the ACL.

Recommended Action Use IP addresses rather than hostnames in the ACL.

Subscribers and Registrations

Symptom Users can no longer register. The error_log file contains the following line:

ACE_INET_Addr::ACE_INET_Addr: localhost: Invalid argument send_n failed: Broken pipe, expected len = 4, actual len = -1.

Possible Cause A farm member is configured whose name or IP address cannot be resolved.

Recommended Action Determine why the machine cannot resolve the name or address and fix that problem.

Symptom Devices can no longer register.

Possible Cause Registration might be disabled in the GUI or the sipd.conf file.

Possible Cause If authentication is required, one of the following might be the problem:

- The SIP-UA and password might not be defined in MySQL or the RADIUS server.
- Cisco SPS might be unable to connect to the MySQL database or RADIUS server.

Possible Cause The type of authentication that the SIP-UA is trying to use might be set incorrectly. Cisco SPS supports only HTTP basic and digest authentication locally for SIP user agents.

Note Due to its weak security, basic authentication has been deprecated. This is a change from RFC 2543. It is not disabled or removed from Cisco SPS, but will no longer be supported or extended to interwork with new or modified functionality. We strongly discourage the use of basic authentication.

Possible Cause Access control lists prevent SIP user agents from registering.

Symptom Endpoints do not register. Error messages suggest that a proxy-server member is not in the farm.

Possible Cause The system cannot resolve its hostname.

Recommended Action Review, in the *Cisco SIP Proxy Server Version 2.2 Installation Guide* section on prerequisites, information about hostname resolution.

Symptom Call forwarding does not configure properly.

Possible Cause You might need to define the corresponding subscribers.

Recommended Action Rather than editing an existing subscriber, add a new one:

```
user --> 5100
domain --> cisco.com
```

Add a call-forwarding URL for the new user and enable the corresponding call-forwarding feature in the sipd.conf file.

Symptom When you try to use the registrar feature on customer-premises-equipment (CPE) routers, requests come to Cisco SPS in the format <*pots-dial-peer-number*>@CPE_A.Sip.Vodafone.com. In response, CPE_A sends 02109187.@CPE_A.vodafone.com to SPS. Registration occurs but the number that is displayed in the routing table is 02109187 without the trailing period, so calls that would match 02109187. are discarded. If you specify all the numbers—for example, CPE_A sends 021091870—then everything works.

Possible Cause The period (.) is a visual separator within a phone number (for example, 408.902.3126 is equivalent to 408-902-3126 is equivalent to 4089023126). SPS strips out all visual separators as part of normalizing numbers before storing them in the registry.

Note The "*" and "." wildcards are not supported for registrations. They are defined for routing only.

MySQL Database

Symptom You cannot access MySQL (Error 1045).

Possible Cause Your MySQL username and password are invalid or have insufficient permission to access the database.

Recommended Action Enter the correct or properly enabled username and password. If you have forgotten your password, assign a new one.

Symptom You cannot access the sip.subscriber subscriber table (Error 1116).

Possible Cause The database whose name you entered does not exist.

Recommended Action Enter a valid name or reinstall the database.

Symptom Bulk import of data into MySQL stops before anything is written to the database.

Possible Cause The provisioning server (pserver), upon checking the imported information, finds fields or values other than what it expects. When you import data through the GUI-based provisioning system, the pserver checks the number of fields and some of the values. For example, it checks whether the port is within the correct range of values. If it finds a problem, it stops the import before writing anything to the database.

Recommended Action Fix any problems and then import the data again.

Routing

Symptom Cisco SPS routes calls improperly.

Recommended Action

- Verify that num-expansion statements, if present, are configured correctly.
- Verify that the various translation mechanisms (Call Forward Unconditional, Static Registry, Dynamic Registry, ENUM, GKTMP) are configured correctly and are populated correctly.
- Verify that the Static Routing table shows correct routes.
- Verify that the DNS server is provisioned correctly for DNS NAPTR, DNS SRV, and DNS A records of the devices to be routed.
- Check the error_log to see what errors are reported (example: bad SIP messages, process errors).

Symptom Cisco SPS seems not to understand [ ] characters in the routing table. For example, the entry 21091878[1-8] is not matched, and the call cannot be established.

Possible Cause SPS does not support the [1-8] notation. You can use . to indicate [0-9]. Or you can enter several routes—in this case eight, one for each permutation. Note that this notation is supported for routes only, not for registration. Wildcard registrations are not defined for SIP and are not supported by SPS.

Symptom Cisco SPS static domain routes seem unable to use raw IP addresses.

Possible Cause Cisco SPS static domain (IP-based) routes do not support raw IP addresses.

For example, say that you configure a route to `.com.ibm.*` with a next hop of `proxy-1.isp.com`. A request is received for A.B.C.D that happens to be a host in the IBM domain. SPS forwards directly to A.B.C.D. It does a reverse DNS lookup and forwards to `proxy-1.isp.com` only if the Request URI contains `ibm.com` as in the following examples:

```
INVITE sip:user@us.ibm.com
INVITE sip:user@bldg1.sjc.ca.us.ibm.com
```

Farming

Symptom Cisco SPS farming works improperly.

Recommended Action

- Verify that farm members have the same GUI or sipd.conf file configuration.
- Verify that each farm member contains an entry for the other farm member in its GUI or sipd.conf file configuration.
- Verify that both farm members are running the same version of Cisco SPS.
- Verify, using Network Time Protocol (NTP), that both farm members are synchronized to the same clock source.

Symptom Endpoints do not register. Error messages suggest that a proxy-server member is not in the farm.

Possible Cause The system cannot resolve its hostname.

Recommended Action Review, in the *Cisco SIP Proxy Server Version 2.2 Installation Guide* section on prerequisites, information about hostname resolution.

Regroute Tool

Symptom When you use the sysadmin_sps_regroute tool, writes to MySQL fail.

Possible Cause The sysadmin_sps_regroute tool and the MySQL database might be from different Cisco SPS releases.

Recommended Action Upgrade the earlier-release component.

Symptom When you use the sysadmin_sps_regroute tool to list everything in the registry database, you get a segmentation fault.

Possible Cause Your registration database might contain some garbage that the sysadmin_sps_regroute tool cannot parse, causing the tool to crash, possibly due to a power outage or some other hard reset of the machine.

Recommended Action If you can tolerate losing the data in your registration and route databases, create a new (empty but not corrupted) database by entering the following commands on each Cisco SPS in the farm:

1. `./bin/sip stop`
2. `rm ..data/*_db`
3. `./bin/sip start`

Then refresh registrations and add routes again.

Symptom The sysadmin_sps_regroute tool cannot push a registration into MySQL.

Possible Cause The sysadmin_sps_regroute tool works under the assumption that the subscriber table is named “subscriber.”

Recommended Action Rename the subscriber table to the default “subscriber.”

Applications

Q. For a prepaid service using Cisco SPS, how does one arrange for SPS to tear down a call when the amount (minutes) reaches zero for the call?

A. SPS cannot time active calls or tear them down. However, it does provide RADIUS authentication and accounting interfaces that you may find helpful in doing just this. See the *Cisco SPS RADIUS Interface Specification* at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/index.htm

Using these interfaces, you can have SPS authenticate all call attempts as well as send accounting starts and stops for all calls.

One way to use this along with your own RADIUS application to do prepaid calling is to require that all endpoints use a session timer. In this case, SPS sends authentication credentials for all calls to your application not only for the original INVITE request but for all re-INVITEs to refresh the session timer as well. Once a user goes over their credit limit, you can start rejecting their authentication credentials. This causes the call to be torn down when the session timer expires.

This scheme requires support for session timer as described in *Session Timers in the Session Initiation Protocol (SIP)* at http://www.ietf.org/internet-drafts/draft-ietf-sip-session-timer-13.txt

APPENDIX A

SIP Compliance

This appendix describes Cisco SIP proxy server (Cisco SPS) compliance with the Internet Engineering Task Force (IETF) definition of Session Initiation Protocol (SIP) as described in the following RFCs.

RFC[1]	Title
2543	*SIP: Session Initiation Protocol (March 1999)*
3261	*SIP: Session Initiation Protocol (June 2002)*
3263	*SIP: Locating SIP Servers (June 2002)*

1. Not all supported RFCs are listed.

Contents

RFC 2543 and RFC 3261

SIP Functions

Table A-1 SIP Functions

Function	Supported?
Proxy server	Yes (transaction stateful, parallel forking, and recursive)
Redirect server	Yes
Registrar server	Yes

SIP Methods

Cisco SPS supports five of the six methods used by SIP. It handles unknown methods such as NEWMETHOD in the same manner as known methods such as OPTIONS and REFER.

Table A-2 SIP Methods

Method	Supported?	Cisco SPS Action
ACK	Yes	Forwards ACK requests.[1]
BYE	Yes	Forwards BYE requests.
CANCEL	Yes	Forwards CANCEL requests.[2]
INFO	Yes	Forwards INFO requests.
INVITE	Yes	Forwards INVITE requests.
NOTIFY	Yes	Forwards NOTIFY requests.
OPTIONS	Yes	Responds to OPTIONS requests.
REFER	Yes	Forwards REFER requests.
REGISTER	Yes	Supports both user and device registration.
SUBSCRIBE	Yes	Forwards SUBSCRIBE requests.
UPDATE	Yes	Forwards UPDATE requests.

1. The SPS can generate a local ACK for a non-200 OK final response to an INVITE request.
2. The SPS can generate a local CANCEL for a pending branch when it receives a 200 OK or 6xx response from the branch.

SIP Responses

Cisco SPS supports the following SIP responses:

- 1xx Response—Information Responses
- 2xx Response—Successful Responses
- 3xx Response—Redirection Responses
- 4xx Response—Request Failure Responses

- 5xx Response—Server Failure Responses
- 6xx Response—Global Responses

Table A-3 SIP Responses

SIP Response		Meaning	Supported?	Cisco SPS Action
1xx Response—Information Responses				
100	Trying	Action is being taken on behalf of the caller, but the called party is not yet located.	Yes	Generates and forwards this response for an incoming INVITE. Upon receiving this response, waits for a 180 Ringing, 183 Session progress, or 200 OK response.
180	Ringing	Called party is located and is being notified of the call.	Yes	Forwards this response.
181	Call is being forwarded	Call is being rerouted to another destination.	Yes	
182	Queued	Called party is not currently available or elects to queue the call rather than reject it.	Yes	
183	Session progress	System performs inband alerting for the caller.	Yes	
2xx Response—Successful Responses				
200	OK	Request has been successfully processed. The action taken depends on the request made.	Yes	Generates this response to a REGISTER or CANCEL request. Otherwise forwards this response.
3xx Response—Redirection Responses				
300	Multiple choices	Address resolves to more than one location. All locations are provided and the user or UA can select which location to use.	Yes	Does not generate this response. If recursive is enabled, recurses on all contacts; otherwise forwards this response.
301	Moved permanently	User is no longer available at the specified location. An alternate location is included in the header.	Yes	
302	Moved temporarily	User is temporarily unavailable at the specified location. An alternate location is included in the header.	Yes	In redirect mode, generates this response when it locates one or more contacts. In proxy mode, if recursive is enabled, recurses on all contacts; otherwise forwards this response.
305	Use proxy	Caller must use a proxy to contact the called party.	Yes	Does not generate this response. If recursive is enabled, recurses on all contacts; otherwise forwards this response.
380	Alternative service	Call is unsuccessful, but alternative services are available.	Yes	

Table A-3 SIP Responses (continued)

SIP Response		Meaning	Supported?	Cisco SPS Action
4xx Response—Request Failure Responses				
400	Bad Request	Request can not be understood because of an illegal format.	Yes	Generates and forwards this response.
401	Unauthorized	Request requires user authentication.	Yes	Forwards this response. If it is configured as a registrar and authentication is enabled, generates this response.
402	Payment required	Payment is required for server to complete the call.	See Cisco SPS action	In registrar mode and if proxied by the proxy server, generates this response.
403	Forbidden	Server has received and understood the request but will not provide the service.		
404	Not found	Server has definite information that the user does not exist in the specified domain.	Yes	Generates and forwards this response.
405	Method not allowed	Method specified in the request is not allowed. The response contains a list of allowed methods.	See Cisco SPS action	Forwards this response.
406	Not acceptable	Requested resource can generate only responses that have unacceptable content as specified in the accept header of the request.		
407	Proxy authentication required	Similar to the 401 Unauthorized response, but client must first authenticate itself with the proxy.	Yes	Forwards this response. If authentication is enabled, generates this response.
408	Request timeout	Server could not produce a response before the expiration timeout.	See Cisco SPS action	Generates and forwards this response.
409	Conflict	Request cannot be processed because of a conflict with the current state of the resource.		
410	Gone	A resource is no longer available at the server and no forwarding address is known.	See Cisco SPS action	Forwards this response.
411	Length required	User refuses to accept the request without a defined content length.	Yes	
413	Request entity too large	Server refuses to process the request because it is larger than the server is willing or able to process. If a retry after header field is contained in this response, the user can attempt the call once again in the retry time provided.	See Cisco SPS action	
414	Request-URI too long	Server refuses to process the request because the Request-URI is too long for the server to interpret.	Yes	Generates and forwards this response.

Table A-3 SIP Responses (continued)

SIP Response		Meaning	Supported?	Cisco SPS Action
415	Unsupported media	Server refuses to process the request because the format of the body is not supported by the destination endpoint.	Yes	Forwards this response.
420	Bad extension	Server cannot understand the protocol extension indicated in the Require header.	Yes	Generates and forwards this response.
480	Temporarily unavailable	Called party was contacted but is temporarily unavailable.	Yes	Forwards this response. If preauthentication is enabled and fails, generates this response.
481	Call leg/transaction does not exist	Server ignores the request because it is either a BYE for which there is no matching leg ID or a CANCEL for which there is no matching transaction.	Yes	Generates and forwards this response.
482	Loop detected	Server received a request that includes itself in the path.		
483	Too many hops	Server received a request that requires more hops than allowed by the Max-Forwards header.		
484	Address incomplete	Server received a request that contains an incomplete address.	See Cisco SPS action	Forwards this response.
485	Ambiguous	Server received a request that contains an ambiguous called-party address. It can provide possible alternative addresses.		
486	Busy here	Called party was contacted but his or her system is unable to take additional calls.		
487	Busy here; request cancelled	Request was terminated by a BYE or CANCEL request.	Yes	
488	Not acceptable media	An error in handling the request occurred.	See Cisco SPS action	

Table A-3 SIP Responses (continued)

SIP Response		Meaning	Supported?	Cisco SPS Action
5xx Response—Server Failure Responses				
500	Server internal error	Server or gateway encountered an unexpected error that prevents it from processing the request.	Yes	Generates and forwards this response.
501	Not implemented	Server or gateway does not support the functions required to complete the request.		
502	Bad gateway	Server or gateway received an invalid response from a downstream server.	See Cisco SPS action	Forwards this response.
503	Service unavailable	Server or gateway is unable to process the request due to an overload or maintenance problem.	Yes	Generates and forwards this response.
504	Gateway timeout	Server or gateway did not receive a timely response from another server (such as a location server).	See Cisco SPS action	Forwards this response.
505	Version not supported	Server or gateway does not support the version of the SIP protocol used in the request.	Yes	
6xx Response—Global Responses				
600	Busy everywhere	Called party was contacted but is busy and cannot take the call at this time.	See Cisco SPS action	Forwards this response.
603	Decline	Called party was contacted but cannot or does not want to participate in the call.		
604	Does not exist anywhere	Server has authoritative information that the called party does not exist in the network.		
606	Not acceptable	Called party was contacted, but some aspect of the session description was unacceptable.		

SIP Header Fields

Note All SIP header fields that concern the Cisco SPS are correctly handled and parsed except for the Hide and Encryption header fields. Header fields that do not directly affect the Cisco SPS or that are unknown to it are passed unaltered in the SIP request.

SIP Transport Layer Protocols

Table A-5 SIP Transport Layer Protocols

Transport Layer Protocol	Supported?
Unicast UDP	Yes
Multicast UDP	No
TCP	Yes
TLS	Yes

SIP Security

Table A-6 SIP Security

Mode	Supported?
Encryption Mode	
End-to-end	No
Hop-by-Hop	Yes
Authentication Mode	
Basic[1]	Yes
Digest	Yes
Proxy	Yes

1. Due to its weak security, basic authentication has been deprecated. This is a change from RFC 2543. It is not disabled or removed from Cisco SPS, but will no longer be supported or extended to interwork with new or modified functionality. We strongly discourage the use of basic authentication.

RFC 3263

SIP DNS Records Usage

Table A-7 SIN DNS Records Usage

DNS Resource Record Type	Supported?
A	Yes
NAPTR	Yes
SRV	Yes

APPENDIX **B**

Manual Configuration

You can configure Cisco SIP proxy server (SPS) in either of two ways:

- Using the GUI-based provisioning system
- Manually editing text-based files

This appendix describes how to configure Cisco SPS manually. Unless your situation is highly unusual, do not perform manual configuration. Use the GUI-based provisioning system, as described in Chapter 2, "Configuring Cisco SPS" and Chapter 3, "Operating and Maintaining Cisco SPS."

Caution All Cisco SPS 1.*x* versions require manual configuration. Therefore, for backward compatibility, Cisco SPS supports manual editing of all configuration files. However, if you use the GUI-based provisioning system, do not attempt manual editing. Manual changes to any configuration file written by the GUI are overwritten when the GUI is used again.

Contents

Prerequisites

- Install Cisco SPS and activate the license as described in the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.
- Change to the directory in which the Cisco SPS configuration file (sipd.conf file) is located and open the file using a text editor such as vi. A default sipd.conf configuration file is copied at installation to the following location:

  Linux:
  ```
  # /usr/local/sip/conf/
  # vi sipd.conf
  ```

  Solaris:
  ```
  # /opt/sip/conf
  # vi sipd.conf
  ```

Note Do not use a Microsoft Windows DOS text editor. When you save the sipd.conf file using a DOS editor, the <eol> (end of line) characters are changed and Cisco SPS has trouble reading them.

How to Replace, Upgrade, or Delete a Cisco SPS License

Cisco SPS licenses are of two types: evaluation and permanent. You can replace one evaluation license with another, upgrade from an evaluation license to a permanent license, or delete a license.

Your license is delivered to you in the form of a license key—that is, a sequence of text characters that the Cisco SPS must read and validate at startup before it can run.

Detailed Steps

Step 1 Copy the new license key, in preparation for pasting.

Step 2 Open the license (license.conf) file:

Linux: `/usr/local/sip/conf/license.conf`

Solaris: `/opt/sip/conf/license.conf`

Step 3 Comment out the existing “LicenseKey” line and leave it in the file for backup.

Step 4 Add a new LicenseKey line with the new key value.

Step 5 Save and close the file.

How to Define the Cisco SPS Configuration File

This section describes the following:

Information About the Cisco SPS Configuration File and Directives

The Cisco SPS configuration file is sipd.conf. The file contains directives that define Cisco SPS operation, functionality, and interfaces. A default sipd.conf configuration file is copied at installation into the following:

Linux: `/usr/local/sip/conf/`

Solaris: `/opt/sip/conf/`

In most cases, you can use the default configuration for starting Cisco SPS and placing some test registrations and calls through it, but you might need to customize the defaults for your particular environment. If you make changes, you must restart the server in order for the changes to take effect. In general, a graceful restart is sufficient; however, for some changes, a complete restart is required. The directives for which changes require a complete restart are identified below.

Cisco SPS directives are similar to Apache server directives. Directives are grouped into four categories:

- Server-global directives—Define the overall operation of Cisco SPS. See the "Configuring Server-Global Directives" section on page B-4.
- Host-specific directives—Define basic Cisco SPS operations that are specific to a particular proxy server (rather than to a virtual proxy host that represents more than one server on one machine, as might be the case for companies that share a web server and yet each have their own domain (www.company1.com and www.company2.com) and access to the web server). See the "Configuring Host-Specific Directives" section on page B-5.
- Server-core directives—Define the primary SIP functionality of Cisco SPS. See the "Configuring Server-Core Directives" section on page B-8.
- Standard directives—Define Cisco SPS interfaces and additional functionality on a per-module basis. See the "Configuring Standard Directives" section on page B-13.

Directives are listed below in the order in which they appear in the Apache server (on which Cisco SPS is based). That is also the order in which the directives are written in the sipd.conf file and in which they appear under the various tabs in the GUI-based provisioning system.

Information About Log Files

See the discussion of log files in the "Information About Log Files" section on page 3-3.

It is particularly important to develop a plan for rotating logs and removing old logs from the hard drive. A typical scenario that does so is as follows:

```
# cd <server_root>/logs
# mv access_log access_log.old
# mv error_log error_log.old
# sip graceful
# sleep 600
# mv access_log.old <some remote location>
# mv error_log.old <some remote location>
```

However, rather than rotate and move logs as just described, we recommend that you use pipes to set up periodic rotation.

In addition to writing directly to a file, you can direct Cisco SPS to write error and access log files through a pipe to another process, which allows you to rotate logs without restarting the server. Cisco SPS includes a simple program called rotatelogs for this purpose.

To write logs to a pipe, simply replace the filename with the pipe character | followed by the name of the executable that is to accept log entries on its standard input.

Configuring Server-Global Directives

Server-global directives are generic server directives that define the overall operation of the server. This does not include directives that configure protocol-specific (HTTP or SIP) details.

Note Cisco SPS uses standard Apache directives to configure the SPS global environment. If the default for an Apache directive differs for Cisco SPS, the SPS default is listed below. For more detail on Apache directives, see the Apache website at http://www.apache.org.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf/ directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **<ServerRoot>**—Directory in which Cisco SPS scripts and executables (bin/), configuration (conf/), and error-log (logs/) files reside. Do not add a forward slash (/) to the end of the directory path. Default is as follows:

  Linux: `/usr/local/sip`

  Solaris: `/opt/sip`

- **LockFile**—Path to the lockfile used when Cisco SPS is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or USE_FLOCK_SERIALIZED_ACCEPT. (A lockfile is a type of file that Apache uses to let multiple processes access the same network socket.)

  You should normally leave this directive at its default value. Change the value, though, if the logs directory is NFS mounted, because the lockfile must be stored on a local disk. For more information on lockfile location, refer to Apache's documentation on this directive at http://www.apache.org/docs/mod/core.html#lockfile.

  The protocol identifier (PID) of the main server process is automatically appended to the filename. Default is logs/accept.lock.

- **PidFile**—Path and file where Cisco SPS records its process ID upon startup. A filename that does not begin with a forward slash (/) is assumed to be relative to <ServerRoot>. Default is logs/sipd.pid.

- **ScoreBoardFile**—Memory-mapped file in which internal server process information is stored. This file is automatically created if your architecture requires it. If the file is automatically created, ensure that no two servers share the same file. Default is logs/apache_runtime_status.

- **prefork MPM module**—Module that implements a nonthreaded, preforking web server for handling requests. This directive causes Cisco SPS to monitor child processes and, when necessary, spawn additional child processes to handle incoming SIP requests and responses. If too few requests and responses are being created, Cisco SPS tears down some idle child processes.

  Maximum and minimum values for the following prefork-MPM-module directives depend on available platform resources. Cisco SPS ignores prefork-MPM-module directives if the server runs in single-process mode (/sipd -DONE_PROCESS) for debugging purposes.

  – **StartServers**—Number of child processes that Cisco SPS creates upon startup. Default is 5.

  – **MinSpareServers**—Minimum number of idle child processes (that is, processes that do not handle requests). Default is 5.

  - **MaxSpareServers**—Maximum number of idle child processes. Such processes that exceed this number are torn down. Do not set this parameter to a large number. Default is 10.
  - **MaxClients**—Number of simultaneous requests that Cisco SPS can support; this number must be no greater than the number of child processes to be created. Default is 20.
  - **MaxRequestsPerChild**—Maximum number of requests that an individual child process (process that handles UDP traffic, IPC traffic, and timeouts) can handle during its life. If this number is exceeded, the child process is torn down and replaced by a new child process. This directive limits the amount of memory that processes can consume by accidental memory leakage. Timeouts occur every 50 milliseconds even in the absence of SIP traffic, and the counter is updated. Commonly used value is on the order of hours or days, (100000) or days (1000000). Default is 0 (child process is never torn down).

- **Listen**—List of ports or port and IP address combinations that the server listens to. This directive binds the server to specific IP addresses and specifies whether the server should listen to more than one IP address or port. If you specify only a port, the server responds to requests on all IP interfaces on that port. Valid entries include port and IP:port, but not IP only. Default is all requests on all IP interfaces. The following are examples of proper syntax:

```
Listen 3000
Listen 10.23.56.78:5060
```

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Host-Specific Directives

Host-specific directives define the basic configuration of Cisco SPS. They define server access, error logs, and the frequency with which logs rotate.

Note Cisco SPS uses standard Apache directives to configure the SPS basic configuration. If the default for an Apache directive differs for Cisco SPS, the SPS default is listed below. For more detail on Apache directives, see the Apache website at http://www.apache.org.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf/ directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Port**—Port on which Cisco SPS listens. Default is SIP port 5060. If this setting is less than 1023, Cisco SPS (sipd daemon) initially must be run as root. This is true even if sipd is to run as a different user or group.
- **User/Group**—Name or number of the user and group to run the sipd process as when sipd is started by the root user. Default is csps.
- **ServerName**—Hostname that clients use to create request URIs. This differs from the hostname that the server normally recognizes as its own. For example, the server name might be sip-proxy.company.com rather than the host's real name. This directive is useful for building a server

farm and publishing a single hostname for the farm. This directive is optional. Use it only if you have multiple servers in a farm. The ServerName, if defined, must be a valid Domain Name System (DNS) name for the host.

- **HostnameLookups**—Log client DNS hostnames rather than of IP addresses. Valid values are On and Off. Default is Off.
- **ErrorLog**—Location of the error log file that contains Cisco SPS logs debug and error messages. Default is logs/error_log without rotation.

  To automatically rotate error and debug logs daily without having to tear down and gracefully restart the Cisco SPS (sipd daemon), enter text in the following format, being sure to specify the full path to both the rotatelogs script and the log file that you want to rotate:

  ```
  ErrorLog "|<full path to rotatelogs script> <full path to error-log file>"
  ```

  Examples:

  Linux: `ErrorLog "|/usr/local/sip/bin/rotatelogs /usr/local/sip/logs/error_log 86400"`

  Solaris: `ErrorLog "|/opt/sip/bin/rotatelogs opt/sip/logs/error_log 86400"`

  Rotation-time default is 86400 seconds (24 hours).

  Note See the Tip below for an alternate way to achieve the same result by adding and removing comment markers from existing commented text rather than typing in new text.

- **LogLevel**—Verbosity of messages recorded in the error logs. Valid values are the following:
  - emerg—Emergencies and when system is unusable
  - alert—When action must be taken immediately
  - crit—Critical conditions
  - error—Error conditions
  - warn—Warning conditions (default)
  - notice—Normal but significant conditions
  - info—Informational
  - debug—Debugging messages
- **LogFormat**—Format nicknames of the log file, for use with CustomLog.

  ```
  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
  LogFormat "%h %l %u %t \"%r\" %>s %b" common
  LogFormat "%{Referer}i -> %U" referer
  LogFormat "%{User-agent}i" agent
  ```

- **CustomLog**—Location and format of the access-log file. Default is logs/access_log common.

  To automatically rotate the access log daily without having to tear down and gracefully restart the Cisco SPS (sipd daemon), enter text in the following format, being sure to specify the full path to both the rotatelogs script and the log file that you want to rotate:

  ```
  CommonLog "|<full path to rotatelogs script> <full path to access-log file>" <log
  format>
  ```

  Examples:

  Linux: `CommonLog "|/usr/local/sip/bin/rotatelogs /usr/local/sip/logs/access_log 86400" combined`

Solaris: `CommonLog "|/opt/sip/bin/rotatelogs /opt/sip/logs/access_log 86400" combined`

Rotation-time default is 86400 seconds (24 hours).

Remove any other existing CommonLog entries, because only one common log can be configured at one time.

Note See the Tip below for an alternate way to achieve the same result by adding and removing comment markers from existing commented text rather than typing in new text.

Step 3 Develop a plan to periodically remove old log files from the local hard disk (see the "Information About Log Files" section on page B-3).

Note Rotated logs do not automatically remove themselves. If you use log rotation, simply move inactive logs off the hard disk without renaming, sleeping, or gracefully restarting the server as follows:

```
# cd <server_root>/logs
# mv access_log.<all except the active one> <some remote location>
# mv error_log.<all except the active one> <some remote location>
```

Step 4 Save and close the file, and restart Cisco SPS.

Tip Rather than typing in new text, you can achieve the same result by adding and removing comment markers from existing commented text as follows:

1. Locate the commented line that mentions rotatelogs.
2. Locate the corresponding default line that does not mention rotatelogs.
3. Add and remove comment markers so that just one such line is active.
4. Verify that the path and format in the active line is as you want it.

The following examples show the result of adding and removing comment markers so as to leave just one active line:

Correct:
```
#ErrorLog logs/error_log
ErrorLog "|/opt/sip/bin/rotatelogs /opt/sip/logs/error_log 86400"
```

Correct:
```
ErrorLog logs/error_log
#ErrorLog "|/opt/sip/bin/rotatelogs /opt/sip/logs/error_log 86400"
```

Incorrect:
```
ErrorLog logs/error_log
ErrorLog "|/opt/sip/bin/rotatelogs /opt/sip/logs/error_log 86400"
```

Incorrect:
```
#ErrorLog logs/error_log
#ErrorLog "|/opt/sip/bin/rotatelogs /opt/sip/logs/error_log 86400"
```

Configuring Server-Core Directives

Server-core directives govern how Cisco SPS functions as a redirect, registrar, or proxy server, either transaction stateful or stateless.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **CSPSVersion**—Version of Cisco SPS that matches this configuration file. This directive is read-only; do not manually change it.
- **ProxyDomain**—Default domain to which Cisco SPS belongs. Valid value is a Domain Name System (DNS) domain suffix in standard fully qualified domain name (FQDN) format. There is no default.

  Examples

  ```
  mydomain.com
  company.mydomain.com
  ```

- **StatefulServer**—Whether Cisco SPS is a transaction-stateful or transaction-stateless server. A transaction includes the following: received request, request or requests (if forked) forwarded downstream, responses received from downstream hosts, and best response returned upstream. Valid values are the following:
  - On (stateful)—Remember incoming and outgoing requests, provide reliable retransmission of proxied requests, and return the best final responses.
  - Off (stateless)—Forget all information once a request or response has been processed. Merely forward requests and responses.

  Default is On. If you change the value of this directive, you must restart the server.
- **SipResolveLocalContactsInRedirectMode**—If Cisco SPS is configured as a redirect server, return next-hop routing information and update contact information before returning the SIP 3xx response. Valid values are On and Off. Default is Off.
- **UseCallerPreferences**—Allow user-defined user-agent client (UAC) preferences to override administrator-defined preferences during request handling. Preferences include decisions such as whether to proxy or redirect a request, whether to fork a request (sequential or parallel), whether to recursively search, and to which URI to proxy or redirect a request. Valid values are On (use user-defined preferences) and Off (use administrator-defined preferences; ignore user-defined preferences). Default is On.
- **ServerType**—Whether Cisco SPS functions as a proxy server or as a redirect server. A proxy server processes and routes SIP requests. A redirect server provides contact information by means of SIP redirect (3xx) responses. Valid values are Proxy and Redirect. Default is Proxy.
- **Recursive**—Recursively try addresses returned in a SIP 3xx redirection response. Valid values are On (try addresses) and Off (return the lowest-numbered or best response). Default is On.
- **MaxForks**—Maximum number of branches that can be forked when Cisco SPS functions as a stateful server. Range is 1 to 6. Default is 5.

- **NumericUsernameInterpretation**—Lookup order for numeric user information in the Request-URI header field when the ";user=IP/PHONE" parameter is missing. Valid values are the following:
  - IP_164—Process as URLs first and then as E.164 numbers.
  - E164_IP—Process as E.164 numbers first and then as URLs (default).
  - IP—Process as URLs only.
  - E164—Process as E.164 numbers.
- **NumericUsernameCharacterSet**—Set of characters that Cisco SPS uses to determine whether the user-information portion of a Request-URI is in a category that applies to the "NumericUsernameInterpretation" processing step. This set does not apply to any user-information parameters. Default is +0123456789.-() (global phone number combinations). For more information on this directive, see the sipd.conf file.
- **OrigUserNameSource**—Origin of the UserName attribute in the accounting request message. Valid values are the following:
  - From—The user part of the URL in the From SIP header is used for authentication and to populate standard RADIUS accounting attribute #1 (UserName).
  - Auth—The user provided for authentication in the authorization or proxy-authorization header is used for authentication and billing. If no proxy-authorization header is present, the user is taken from the From header in the billing records.

  Default is Auth.
- **NumExpandAuthUserName**—Apply number-expansion rules to the UserName received in the Authorization or Proxy-Authorization header. Valid values are On and Off. Default is On.
- **SrvForFqdnOnly**—Perform DNS server (SRV) lookups only for hosts that are FQDNs. If the host portion of the Request-URI header field does not contain an IP address but does contain a period, determine the host to be an FQDN. Valid values are On (perform lookups only on FQDN hosts) and Off (perform lookups for any host that does not contain a target port). Default is Off.
- **SipT1InMs**—Time (in milliseconds) after which a request is first retransmitted if no response is received. Default is 500 (0.5 second).
- **SipT2InMs**—Time (in milliseconds) after which the backoff interval for non-INVITE requests does not increase exponentially. Default is 4000 (4 seconds).
- **SipT3InMs**—Default time (in milliseconds) that Cisco SPS waits after receiving a provisional response when processing an INVITE request. If a client does not include an Expires value in the INVITE, this value is used. Default is 60000 (60 seconds).
- **SipMaxT3InMs**—Maximum time (in milliseconds) that Cisco SPS waits after receiving a provisional response when processing an INVITE request. If a client includes an Expires value greater than this value in an INVITE, this value is used instead. Default is 180000 (180 seconds).
- **SipT4InMs**—Time (in milliseconds) that Cisco SPS maintains the transaction control block (TCB) after proxying a final response to a SIP INVITE request. Default is 32000 (32 seconds).
- **MaxInviteRetxCount**—Number of times that Cisco SPS can retransmit a SIP INVITE request. Range is 0 to 6. Default is 6.
- **MaxNonInviteRetxCount**—Number of times that Cisco SPS can retransmit a SIP request other than an INVITE. Range is 0 to 10. Default is 10.
- **SharedMemorySize**—Size (in bytes) of shared memory to be allocated for TCB. Range is 32000000 to 512000000. Default is 128000000. Recommended size is 128000000 (128 MB).

- **RegistryCleanupRate**—Time (in milliseconds) after which expired or deleted entries are removed from the registry. Default is 180000 (180 seconds).
- **AddRecordRoute**—Add the Record-Route header to an initial SIP INVITE request. The Record-Route header field contains a globally reachable Request-URI that identifies the proxy server. When the proxy server adds the Request-URI to the Record-Route field in SIP messages, the server is kept in the path of subsequent requests for the same call leg. Valid values are On (add) and Off (do not add). Default is Off. ServerType must be set to Proxy for this directive to apply.
- **AddTransportInRecordRoute**—Force use of a transport parameter in the Record-Route header. Doing so is useful for when the proxy server does not use the domain name in the path headers (that is, when ProxyAddressResolutionType is NOT SRV). Enabling this directive explicitly allows the proxy server to work with old equipment that does not yet support NAPTR and SRV. Valid values are On (force use) and Off (do not force use). Default is Off.
- **SipRouteHeaderTransportType**—Transport type for routes specified in Route headers of SIP requests handled by Cisco SPS. If the route contains an explicit transport parameter, this directive is ignored and the transport identified in the route header is used. Valid values are TCP, TLS, and UDP. Default is UDP.
- **AllowNaptrLookup**—Enable NAPTR lookup logic on the proxy server. Valid values are On (enable) and Off (disable; use TransportPrefOrder to select transport). Default is On.
- **TransportPrefOrder**—Transport preferences for times when NAPTR cannot be used or is unsuccessful. Valid values are the following: TLS_TCP_UDP, TLS_UDP_TCP, TCP_TLS_UDP, TCP_UDP_TLS, UDP_TLS_TCP, UDP_TCP_TLS, TLS_TCP, TLS_UDP, TCP_TLS, TCP_UDP, UDP_TLS, UDP_TCP, TLS, TCP, UDP. If SipTlsEnable is disabled, a transport preference of TLS is ignored. Default is TLS_TCP_UDP.
- **DiffServValue**—Value (in hex) to mark the type-of-service (TOS) byte of the IP header field of the transmitted SIP packets. Default is 0x60.

  Values and their meanings are specified in RFC2474, RFC2475, RFC2597, and RFC3246. Valid values are the following:

  - Expedited Forwarding (EF) queue (RFC3246) value: 0xB8
  - Assured Forwarding (AF) queue (RFC2597) values:

	Class 1	Class 2	Class 3	Class 4
Low drop	0x28	0x48	0x68	0x88
Medium drop	0x30	0x50	0x70	0x90
High drop	0x38	0x58	0x78	0x98

  - IP routing (Class 6) value: 0xC0
  - Streaming video (Class 4) value: 0x80
  - Telephony signaling (voice & video) Class 3) value: 0x60
  - Network management (Class 2) value: 0x40
  - Scavenger (Class 1) value: 0x20
  - Other (default, Class 0) value: 0x00

  Some networks might alternatively recognize the type-of-service (RFC1349, RFC1812) bitmasks as follows.

  - Minimize delay: 0x10

  - Maximize throughput: 0x08
  - Maximize reliability: 0x04
  - Minimize cost: 0x02

Note This directive marks IP packets to a specified value. Marked packets receive special treatment only if network IP routers and switches are configured to provide it.

- **Sip_Token_Port**—Port that the synchronization server uses. This port must be the same for all servers in a farm. Default is 22794. If you change the value of this directive, you must restart the server.
- **Sip_Services_Port**—Port on the synchronization server. Default is 52931. If you change the value of this directive, you must restart the server.
- **RadiusUserNameAttrAddDomain**—Append the domain in the From header to the username in the RADIUS UserName attribute (user@domain format). Domains other than ProxyDomain (default) or a domain in Virtual_Proxy_Domain are not appended. Valid values are On (append) and Off (do not append). Default is On.
- **RadiusRetransmissionInterval**—Time (in milliseconds) between retransmissions to the RADIUS server. Default is 2000.
- **RadiusRetransmissionCount**—Number of times to retransmit before deciding that the RADIUS server is unreachable. Default is 2.
- **RadiusRetransmissionAfterFailure**—Number of times to retransmit the current RADIUS request if all attempts to send the previous request fail. Default is 0.
- **RadiusRetryTime**—Time (in seconds) before retrying the primary RADIUS server, if it is out of service. Default is 300 (5 minutes).
- **ProxyAddressResolutionType**—Type of DNS configuration that is set up for SIP services in the proxy-server domain. Valid values are the following:
  - IP—No DNS configuration is available. The proxy server uses IP addresses in headers.
  - A—DNS is set up with A records corresponding to the ServerName directive. Hence, the proxy server uses the value of this directive in headers. If the ServerName directive is not set, then the proxy server uses its hostname SRV, which denotes that the proxy-server domain has SRV records set, and hence the proxy server uses the value of the ProxyDomain directive in headers.

  Set this directive in conjunction with the ProxyDomain directive. Default is IP.
- **IpAddrInPathHeaders**—IP address to use in the Via and Record-Route path headers when ProxyAddressResolutionType is set to IP. Use this directive to control which address is used on multihomed servers. Default is to use the first value returned from gethostbyname.
- **IgnoreProxyRequire**—Behave as if ProxyRequire headers are not present in a request. For example, suppose that an INVITE request contains ProxyRequire:extension-foo, but the proxy has no formal logic to understand extension-foo. RFC 2543 requires that a 420 response be returned, but with IgnoreProxyRequire configured, the INVITE is processed as if that particular header were not present. Valid values are strings of text that the proxy server is to ignore in headers.
- **SIPStatsLog**—Print statistics to the stats_log file. Valid values are On and Off. Default is On.
- **SIPStatsInterval**—Time (in seconds) for which statistics are logged. Default is 3600.
- **SharedMemoryStatsLog**—Enable debugging for shared memory. Valid values are On and Off. Default is Off.

- **SharedMemoryStatsInterval**—Time (in minutes) for which the log is written to the sharedmem_stats_log file in the logs directory. Default is 5.
- **SipTcpMaxTCPConnections**—Number of SIP TCP connections that can be open at any time. Default is 128.

  Note This setting is ignored in favor of using limits enforced by the operating system. For both Linux and Solaris, this limit is 1024 (as set by FD_SETSIZE) by default. Increasing this limit might degrade performance.

- **SipTcpMaxConnectTimeout**—Time (in milliseconds) that the server waits to connect to the client. Range is 150 to 10000 (10 seconds). Default is 1000 (1 second).
- **SipTcpReuseConnection**—Reuse the TCP connection for subsequent transactions with the same entity. Valid values are On (reuse) and Off (do not reuse). Default is Off.

  All SIP entities using TCP for transport to one another should share the same setting. This prevents performance degradation and potential call failures. Otherwise, a proxy server that has this flag set to On continuously tries to reuse the same connection, even while another hop where this flag is set to Off is being torn down. For best performance, set to On only when all elements support that setting. To interwork with Cisco IOS gateways, set this to Off. If other entities in the network reuse connections, define persistent connections to those entities in the conf/persistent_tcp.conf file.
- **SipTlsEnable**—Enable TLS. Valid values are On and Off. Default is Off. This directive is read-only during start/restart.
- **AllowSipTlsConversionToSip**—Terminate incoming SIPS requests on SIP contacts. Doing so presents a security risk; do so with caution and only if you know in advance that your endpoints and gateways are incapable of handling SIPS and TLS connections.
- **SipTlsPort**—TLS port. Default is 5061. This directive is read-only during start/restart.
- **SipTlsSessionTimeout**—Server-side-session cache timeout (in seconds). Sessions are not reusable after this timeout. Default is 300.
- **SipTlsCertificateFile**—Location of the PEM-encoded certificate file for the server. This directive is read-only during start/restart.
- **SipTlsCertificateKeyFile**—Location of the PEM-encoded private key file for the server. This directive is read-only during start/restart.
- **SipTlsCACertificateFile**—Location of certificates of certification authorities (CAs) with whose clients Cisco SPS deals. Cis co SPS uses this information for client authentication. This directive is read-only during start/restart.
- **SipTlsMutualAuthentication**—Perform mutual authentication with the client. Valid values are On and Off. Default is Off.
- **DebugFlag StateMachine**—Log information on operation of the per-sipd child SIP state machine to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.
- **DebugFlag Radius**—Log information for RADIUS messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.
- **DebugFlag Parser**—Log information on operation of the per-sipd child SIP parser to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.
- **DebugFlag SipTcp**—Log information on TCP transport of SIP messages by TCP services to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

- **DebugFlag SipTls**—Log information about the TLS transport of SIP messages by TCP services to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Standard Directives

Cisco SPS contains a number of modules that you can use to configure a variety of interfaces, services, and features. This section includes the following information:

Configuring the MySQL Database Subscriber-Table Interface

You can configure an interface to a MySQL database subscriber table to maintain subscriber records for user authentication, authorization, accounting, and per-user call-forwarding. You can also map field names used by Cisco SPS to an existing MySQL subscriber table.

Prerequisites

- If a MySQL database subscriber table exists in the network, use directives in the MySQL module to map the field names used by Cisco SPS to those used in the MySQL database subscriber table.
- If a MySQL subscriber table does not exist in the network, create one, using the install_mysql_db script (refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide*).

Note

- If you use the GUI-based provisioning system, the MySQL database tables are created during provisioning-system installation, and you cannot modify the field names in the tables. Refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide* for details.
- For information on working with MySQL databases, see the MySQL website at http://www.mysql.com.

- You can conduct a MySQL query at any time. Use the following information:
  - For terminating features such as the call forwarding: the “user” portion of the Request-URI
  - For originating features such as Authentication: the UserName from the Authorization, Proxy-Authorization Header, or From header

  In either case, you may expand the key to a full E.164 number as needed before the MySQL query.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **DB_MySQL**—Connect by means of TCP with the MySQL database. Valid values are On (connect) and Off (do not connect). Default is Off.
- **DB_MySQL_HostName**—Hostname or IP address of the system on which the primary MySQL database resides.
- **DB_MySQL_Secondary_HostName**—Hostname or IP address of the system on which the secondary MySQL database resides.
- **DB_MySQL_DB**—Name of the database in which the subscriber table is stored and maintained.
- **DB_MySQL_Username**—Login username for the database account.
- **DB_MySQL_Password**—Login password for the database account.
- **DB_MySQL_SubscriberTable**—Name of the table in which the subscriber entries are stored.
- **DB_MySQL_Connect_Timeout**—Timeout value (in seconds) for when Cisco SPS attempts to connect to the MySQL database server. After expiration of this time, SPS marks the connection as bad to prevent more child processes from blocking and resets the connection flag as soon as the server returns online. Adjust this value according to the traffic load on the server. A large value blocks more child processes than does a small value. Default is 3.
- **DB_MySQL_XXX_Field**—Name equivalent in an existing MySQL database subscriber table. Use this directive to map Cisco SPS field names to their equivalent MySQL names.
- **DebugFlag DBMySQL**—Log mod-sip-db-mysql debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring the GKTMP Interface

You can configure an interface to translate SIP protocol data units (PDUs) to the GateKeeper Transaction Message Protocol (GKTMP) protocol for local-number-portability (LNP) lookups, 1-800 and 1-900 number translations, and endpoint resolutions. A newly started Cisco SPS process initiates a TCP connection with a network-application-manager (NAM) server via the GKTMP interface.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **GktmpConnection**—Connect to the GKTMP interface. Valid values are On (connect) and Off (do not connect). Default is Off.
- **MasterServerHostname**—Hostname of the primary NAM server.
- **MasterServerIpAddress**—IP address of the primary NAM server.
- **MasterServerPort**—Destination port number of the primary NAM server and LNP lookup services.
- **SecondaryServerHostname**—Hostname of the secondary NAM server.
- **SecondaryServerIpAddress**—IP address of the secondary NAM server.
- **SecondaryServerPort**—Destination port number of the secondary NAM server.
- **GktmpTransportType**— Transport type for routes specified in GKTMP responses received by Cisco SPS. Valid values are TCP, TLS, and UDP. Default is UDP.
- **Debug Flag GKTMP**—Log mod_sip_gktmp debug messages to logs/error_log. Valid values are On and Off. Default is Off.
- **DebugFlag GktmpAPI** —Log mod_sip_gktmp API debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Accounting Services

You can configure Cisco SPS to perform accounting services—that is, to generate and forward transaction or call information to a RADIUS server. This information is in the form of a RADIUS accounting-request message that contains standard billing information such as username, IP address of the proxy server that set up the call, message status type, type of port, session time, ID of the endpoint that is called, and ID of the endpoint that calls.

When accounting service is enabled and the interface to the RADIUS server is configured, Cisco SPS creates and sends accounting records to the RADIUS server according to how you set relevant directives (see Table B-1 and Table B-2).

Table B-1 AccountingServerSide Directive

Condition 1	Condition 2	Returned Message	Sent Record
AccountingServerSide Directive is On	—	200 for INVITE upstream	Server-side START record
	—	Final response for BYE upstream (call is successful)	Server-side STOP record
	AccountingUnsuccessful directive is On	Non-200 final response for an INVITE upstream (call is unsuccessful)	

When Cisco SPS receives an INVITE request from itself or a member of its registry or routing farm, server-side accounting is disabled for that INVITE, and no server-side START or STOP records are sent to the RADIUS server. Similarly, when Cisco SPS sends an INVITE to itself or a member of its registry or routing farm, client-side accounting is disabled for that INVITE, and no client-side START or STOP records are sent to the RADIUS server.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Accounting**—Log accounting information on a RADIUS server. Valid values are On and Off. Default is Off.
- **AccountingServerSide**—Send server-side accounting records for successful calls. Valid values are On and Off. Default is On.
- **AccountingClientSide**—Send client-side accounting records for successful calls. Valid values are On and Off. Default is Off.
- **AccountingUnsuccessful**—Send accounting records for unsuccessful calls. Valid values are On and Off. Default is Off. Operates as follows:
  - On—This directive is interpreted in conjunction with the AccountingServerSide and AccountingClientSide directives. If the AccountingServerSide and this flag are both On, accounting records are sent for server-side unsuccessful calls. If the AccountingClientSide and this flag are both On, accounting records are sent for client-side unsuccessful calls.
  - Off—No accounting records are sent for unsuccessful calls, regardless of the setting for the AccountingServerSide or AccountingClientSide flags.
- **AccountingRecordFormat**—Record format used for accounting. Currently, RADIUS is the only valid option.
- **AccountingTimeFormat**—Whether timestamps are in local or GMT time.
- **PrimaryRadiusAcctIp**—IP address or hostname of the primary RADIUS server to use for accounting.
- **PrimaryRadiusAcctPort**—Destination port number of the primary RADIUS server to use for accounting.

- **PrimaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the primary RADIUS server to use for accounting.
- **SecondaryRadiusAcctIp**—IP address or hostname of the secondary RADIUS server to use for accounting.
- **SecondaryRadiusAcctPort**—Destination port number of the secondary RADIUS server to use for accounting.
- **SecondaryRadiusAcctSecret**—Secret text string shared between Cisco SPS and the secondary RADIUS server to use for accounting.
- **AcctIncludeSIPHeader**—Send the SIP header in VSA #1 (AVPair) within RADIUS accounting messages as they are received by the proxy server—that is, with complete header line (from the 200 OK for the start request and from the BYE for the stop request). You can have a maximum of 50 headers in the sipd.conf file. For RADIUS, this directive is included as the value of Cisco AVPair 1 and attribute name sip-hdr.

Note For information on vendor-specific attributes (VSAs), see the following:

Cisco SPS RADIUS Interface Specification at http://www.cisco.com/univercd/cc/td/doc/product/voice/sipproxy/index.htm or http://www.cisco.com/en/US/products/sw/voicesw/ps2157/index.html

RADIUS VSA Voice Implementation Guide at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Authentication and Authorization

You can configure Cisco SPS to authenticate users or endpoints before it processes a transaction. You can specify that authentication be provided by SPS or a RADIUS server, and that it be done by means of HTTP Digest Authentication or HTTP Basic Authentication.

Note Due to its weak security, basic authentication has been deprecated. This is a change from RFC 2543. It is not disabled or removed from Cisco SPS, but will no longer be supported or extended to interwork with new or modified functionality. We strongly discourage the use of basic authentication.

Authentication is based on the username that Cisco SPS extracts from the From, Authorization, or Proxy-Authorization header, regardless of where authentication takes place.

Cisco SPS expands the name to a full E.164 number before authentication according to the header type and to the rules in the relevant directive:

A Username from This Type of Header...	Expands According to Rules in This Directive...
From	User type and NumericUserNameInterpretation
Authorization or Proxy-Authorization	NumExpandAuthUserName

You can also configure Cisco SPS to ask for the domain of the user (as determined by the host portion of the header) as part of an authentication request.

You can specify the domain in the authentication header in either of two ways:

- Specify a user as <user>@<domain> rather than just user.
- Add the optional uri parameter and specify the domain as the host portion of it.

Specifying the domain in either of these ways causes SPS to look for the domain in the From header. For this to work, the From URL needs to be something like anonymous@<actual-domain> rather than anonymous@anonymous.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi.

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Authentication**—Whether users must be authenticated before their transactions are processed. Valid values are On (authentication required) and Off (authentication not required). Default is Off.

  Note User authentication does not occur if the following three conditions are true, because access control is already satisfied:
  1. Access control is being used.
  2. Hostname or IP address of the sender is covered by a corresponding Allow directive.
  3. The Satisfy directive is set to Any instead of All.

- **AuthRealm**—Realm used in authentication response headers. Default is CISCO.
- **AuthServer**—Server on which user authentication takes place. Valid values are Radius and Proxy. Default is Proxy.
- **AuthScheme**—Authentication method to use when users require authentication before receiving service. Valid values are HTTP_Digest and HTTP_Basic. Default is HTTP_Digest.

  Note Due to its weak security, basic authentication has been deprecated. This is a change from RFC 2543. It is not disabled or removed from Cisco SPS, but will no longer be supported or extended to interwork with new or modified functionality. We strongly discourage the use of basic authentication.

- **AuthDigestQop**—Quality-of-protection (QoP) value for a digest-authentication challenge. Indicates the quality of protection supported. Valid values are auth (authentication only), auth-int (authentication and integrity), and both (allow the client to choose). Default is auth. For backward compatibility, none (from previous releases) is treated as auth.
- **AuthDigestAlgorithm**—Value of the algorithm to be included in a Digest Challenge to the user and used in Authentication Response headers. Valid values are MD5 (algorithm="MD5") and MD5-sess (algorithm="MD5-sess"). Default is MD5.

- **AuthConsumeProxyAuth**—Consume (that is, strips off) the proxy-authorization header before forwarding a request downstream. Valid values are On (consume) and Off (pass downstream). Default is On. When a downstream device needs the header to identify the originator of the request, set to Off.
- **AuthAllow3rdPartyRegistration**—Check unauthorized redirection of calls by a third-party registration. If set to Off, the username in the To header is matched with the username in the From or Authorization header; if these usernames do not match, registration is rejected, regardless of the form of authentication (Basic or Digest). Default is Off.
- **AuthAllow3rdPartyInvite**—Allow third-party INVITE requests for all forms of authentication (Basic or Digest). Valid values are On (user in the From header can differ from user used for authentication) and Off (user in the From header must match user used for authentication). Default is On.
- **RadiusAuthSkew**—Time (in seconds) for which a challenge is valid. Default is 30.
- **PrimaryRadiusAuthIp**—IP address or hostname of the primary RADIUS server to use for authentication.
- **PrimaryRadiusAuthPort**—Destination port number of the primary RADIUS server to use for authentication.
- **PrimaryRadiusAuthSecret**—Secret text string shared between Cisco SPS and the primary RADIUS server to use for authentication.
- **SecondaryRadiusAuthIp**—IP address or hostname of the secondary RADIUS server to use for authentication.
- **SecondaryRadiusAuthPort**—Destination port number of the secondary RADIUS server to use for authentication.
- **SecondaryRadiusAuthSecret**—Secret text string shared between Cisco SPS and the secondary RADIUS server to use for authentication.
- **AuthIncludeSIPHeader**—Send the SIP header in VSA #1 (AVPair) within RADIUS accounting messages as they are received by the proxy server—that is, with complete header line (from the 200 OK for the start request and from the BYE for the stop request). You can have a maximum of 50 headers in the sipd.conf file. For RADIUS, this directive is included as the value of Cisco AVPair 1 and attribute name sip-hdr.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring SIP Access Control and Trust Lists

You can configure access to Cisco SPS.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Order**—Default access state and the order in which Allow and Deny directives are evaluated. In all cases, every Allow and Deny statement is evaluated. There are two valid values:

  - Deny,Allow—Evaluate Deny directives before Allow directives. Allow access by default. Allow access to any client that matches an Allow directive or does not match a Deny directive.
  - Allow,Deny—Evaluate Allow directives before Deny directives. Deny access by default. Deny access to any client that does not match an Allow directive or matches a Deny directive.

Note Separate keywords only by a comma. Do not use blank spaces.

In the following example, all hosts in the company.com domain are allowed access and all other hosts are denied access.

```
Order Deny,Allow
Deny from all
Allow from company.com
```

In the following example, all hosts in the company.com domain are allowed access, except for hosts that are in the foo.company.com subdomain. All hosts not in the company.com domain are denied access because the default state is to deny access to the server.

```
Order Allow,Deny
Allow from company.com
Deny from foo.company.com
```

If the order in the last example is changed to Deny,Allow, all hosts are allowed access. Regardless of the ordering of the directives in the configuration file, the Allow from company.com is evaluated last and overrides the Deny from foo.company.com. All hosts not in the company.com domain are also allowed access because the default state changes to Allow.

- **Allow from**—Which hosts are granted access to an area of the server. Access can be controlled by hostname, IP address, IP address range, or some other characteristic of the client request captured in an environment variable.

  The first argument to this directive is always the *from* hostname. Subsequent arguments can take two different forms: all and host. If Allow from all is specified, all hosts are allowed access, subject to the Deny and Order settings (see below). To allow only particular hosts or groups of hosts to access the server, specify the host in any of the following formats:

  - Partial domain name (example: `Allow from company.com`)
  - Full IP address (example: `Allow from 10.1.2.3`)
  - Partial IP address (example: `Allow from 10.1`)
  - Network/netmask pair (example: `Allow from 10.1.0.0/255.255.0.0`)
  - Network/nnn CIDR specification (example: `Allow from 10.1.0.0/16`)
- **Deny from**—Which hosts are denied access to an area of the server. Valid values are as for the Allow-from directive.
- **Satisfy**—Access policy for both types of access control (Allow and Deny) and authentication checks. Valid values are All (allow and authenticate the sending host) and Any (grant access to the sending host if it passes an access-control-allow or authentication check). With either value, you must turn the authentication module on to ensure that an authentication check is performed.

Note Cisco SPS also uses the Allow-from and Deny-from clauses in the SIP access-control list to determine whether an upstream address is trusted or not when it receives a privacy-related SIP header in an INVITE message. For a downstream address, it extends the syntax to use Allow-to and Deny-to to define a downstream trust list for privacy processing. The same order clause is shared by the SIP access-control list (that is, the upstream trust list) and downstream trust list to define the order of checking between the Allow and Deny lists in their own categories. The satisfy clause affects only access-control processing.

- **Allow to**—Which hosts can receive privacy-related headers such as P-Asserted-Identity, Remote-Party-ID, and Diversion headers with the caller's or redirecting user's true identity. Cisco SPS uses this directive to determine whether a downstream address to which it sends INVITE requests is trusted or untrusted.
- **Deny to**—Which hosts cannot receive privacy-related headers. If a downstream address matches any in the Deny-to list, it is considered as an untrusted address that should not receive any privacy-related headers.

Note On the GUI, Allow-to and Deny-to lists are merged with the access-control list but marked as downstream. Their use is limited to privacy processing.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Privacy

You can configure privacy-related processing of P-Asserted-Identity, Remote-Party-ID, and Diversion headers.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Privacy**—Honor requests for privacy. Reuse the current access-control list (ACL) to determine whether an upstream address is trusted or untrusted (if it passes ACL checking, it is trusted; otherwise, it is untrusted). Use the Allow-to and Deny-to lists with the same ACL "Order" to determine whether a downstream address is trusted or untrusted (if it passes checking from the "Allow to" and "Deny to" with the "Order," it is trusted; otherwise, it is untrusted). Directives that specify further processing detail are listed below. Valid values are On (honor privacy requests) and Off (ignore privacy requests). Default is Off.
- **DebugFlag Privacy**—Log information about processing of PAI, RPID, and Diversion headers in relation to privacy requests. Valid values are On and Off. Default is Off.
- **PrivacyWithPAI**—Accept or pass P-Asserted-Identify (PAI) headers in INVITE requests. Valid values are the following:

  - On—Accept PAI headers in requests received from trusted addresses and remove them from those received from untrusted addresses. Add its own PAI header to authenticated requests, possibly replacing any existing PAI header received from a trusted address. If the From header of the request has a display name “Anonymous” (case-insensitive), add a “Privacy: id” header if one is not already present. In an added PAI header, the user name is the one used for request authentication; the host is the domain name used for request authentication. Before sending a request to an untrusted address, remove the PAI header if there is a “Privacy: id” header or if the From header has a display name “Anonymous.”
  - Off—Pass PAI headers along as is without any processing.

  Default is Off.

- **PrivacyWithRPID**—Add and remove Remote-Party-ID (RPID) headers in INVITE requests. Valid values are the following:
  - On—Accept RPID headers in requests received from trusted addresses and remove them from those received from untrusted addresses. Add its own RPID header to authenticated requests, possibly replacing any existing RPID header received from a trusted address. If the From header of the request has a display name “Anonymous” (case-insensitive), add a “privacy=full” token. In an added RPID header, the user name is the one used for request authentication; the host is the domain name used for request authentication. Before sending a request to an untrusted address, remove the RPID header if it contains a “privacy=full” token or if the From header has a display name “Anonymous.”
  - Off—Pass RPID headers along as is without any processing.

  Default is Off.

- **PrivacyWithDiversion**—Add and remove Diversion headers in INVITE requests. Valid values are the following:
  - On—Accept Diversion headers in requests received from trusted addresses and remove them from those received from untrusted addresses. When it receives a 3xx redirect response, validate the name-addr in the topmost Diversion header against the name-addr in the request URL of the request received from upstream. If they differ, rewrite the former to match the latter.

    Before sending an INVITE to an untrusted address, as in forking or as a result of receiving a 3xx response in recursive mode or call forward for a subscriber, the proxy server rewrites the name-addr to "Anonymous@Anonymous" in Diversion headers that contain "privacy=full" token.
  - Off—Pass Diversion headers in INVITE requests and 3xx responses along as is without any processing.

  Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Preauthentication Query

You can configure Cisco SPS to send a preauthorization query for a new INVITE request to a resource-pool-manager server (RPMS).

You start by creating a list of previous hops and a list of RPMSs that Cisco SPS is to check against. Then, during system operation, Cisco SPS checks a new INVITE request’s previous hop against those in your list of previous hops. If it finds a match, it sends a preauthorization-query RADIUS message on behalf of the INVITE to an RPMS in your list of RPMSs. Cisco SPS does the following:

- If it receives an Accept response from the RPMS, it processes the INVITE normally.
- If it receives a Reject response, it returns a 408 (temporarily unavailable) message to the caller.
- If it receives no response within a specified wait time, it tries the next RPMS in the list. If all RPMSs in the list fail to respond, it processes the INVITE normally, as if the module is Off.
- It designates an RPMS to handle the next call as follows:
  - If any RPMS responds (whether the response is Accept or Reject) within a specified retry time (the counter for which starts when the first RPMS fails to respond), it designates that RPMS.
  - If all RPMSs fail to respond or if the retry time expires, it designates the first RPMS in the list.

Since preauthorization query messages are RADIUS messages, the new RPMS uses the RADIUS module to build and send preauthorization query RADIUS messages.

To see debug messages related to these events, turn on the following debug flags:

- DebugFlag StateMachine
- DebugFlag Radius
- DebugFlag RPMS

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **PreAuthorization**—Preauthorize new INVITE requests. Valid values are On and Off. Default is Off.
- **PreAuthRequestType**—Preauthorization request type. Only valid value with this release is Query.
- **RPMS_ServerIpPortSecret**—List of RPMS IP addresses, port numbers, and secrets (passwords) for up to 10 servers.
- **PreAuthPreviousHop**—IP address, hostname, or domain for up to 100 different hops, or the keyword ALL. The format of an entry is the same as that for an access-list entry.
- **DebugFlag RPMS**—Log RPMS debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Call Forwarding

You can configure Cisco SPS to perform call forwarding, providing that you have a MySQL database.

Note For call forwarding, you need to define the corresponding subscribers. Rather than editing an existing subscriber, add a new one:

```
user --> 5100
domain --> cisco.com
```

Then add a call-forwarding URL for that user. Enable the corresponding call-forwarding feature in sipd.conf as well.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **CallForwardUnconditional**—Forward calls unconditionally. Valid values are On (forward) and Off (do not forward).
- **CallForwardNoAnswer**—Forward calls when a call is not answered within a designated amount of time. Valid values are On (forward) and Off (do not forward).
- **CallForwardBusy**—Forward calls when when the called party is busy (a SIP 486 Busy Here response is received). Valid values are On (forward) and Off (do not forward).
- **CallForwardUnavailable**—Forward calls when a user-agent client (UAC) is unavailable. Calls for users who are listed in the subscriber database but lack a valid registration, or who have a valid registration but do not respond within a designated time, are forwarded to the call-forward-unavailable location. Valid values are On (forward) and Off (do not forward).
- **CallForwardNoAnswerTimer**—Time (in milliseconds) after which to forward an unanswered call. Default is 24000 (24 seconds). This directive requires that CallForwardNoAnswer be set to On.
- **CallForwardUnavailableTimer**—Time (in milliseconds) after which to forward a call when a UAC is unavailable. Default is 24000 (24 seconds). This directive requires that CallForwardUnavailable be set to On.
- **AddDiversionHeader**—Include the CC-Diversion header in SIP messages. Inclusion enables conveyance of call-redirection information during call setup. Valid values are On (include) and Off (exclude). Default is On if call forwarding is enabled.
- **DiversionHeaderName**—Name used for diversion headers generated by this proxy server. Valid values are Diversion, CC-Diversion, and CC-Redirect. Default is Diversion.

Note When the Call Forward feature is invoked for a subscriber whose RedirectDNPrivacy is set to yes, the proxy server adds a "privacy=full" token to the Diversion header added for Call Forwarding when AddDiversionHeader is On and DiversionHeaderName is Diversion. For this implication, see information on the PrivacyWithDiversion directive in the "Configuring Privacy" section on page B-21.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Number Expansion

You can configure support for global number-expansion plans.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Cisco_Numexpand**—Use number expansion. Valid values are On and Off. Default is On.
- **DebugFlag Numexpand**—Log number expansion-related debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file.

Step 4 Start the number plan by assigning it a unique identifier:

```
<NumberPlan ID>
```

where *ID* is the unique identifier of the number plan (for example, global).

Step 5 Specify the number plan, using one or more number-expansion directives:

```
NumExp <unexpanded-pattern> <expanded-pattern>
</NumberPlan>
```

Note You can use a period (.) as a wildcard to represent any digit.

Example:

```
<NumberPlan Global>
    NumExp      2....   +1919392....
    NumExp      7....   +1408527....
    NumExp      8...    5555...
</NumberPlan>
```

Configuring E.164 to Request-URI Address Translation

You can configure Domain Name System (DNS) lookup for translation of an E.164 number (or any number in a number plan) into a list of Request-URIs.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Cisco_Enum**—Translate E.164 numbers to Request-URI s. Valid values are On and Off. Default is Off.
- **Cisco_Enum_Domain**—Private search domain for a private ENUM number plan. This directive is ignored if a Request-URI user begins with the plus (+) character, because the number is part of a global ENUM number plan (e164.arpa).

- **Cisco_Enum_Global_Domain**—Domain to use in either of the following cases:
  - The Request-URI user begins with a plus (+) character (indicating a global domain)
  - A value is not specified for the Cisco_Enum_Domain directive

  Default is e164.arpa.
- **DebugFlag Enum**—Log mod_sip_enum API debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Next-Hop Routing

You can configure Cisco SPS to perform next-hop route lookups for final Request-URIs by means of static route entries. SPS determines static routes by parsing directives such as the destination pattern, transport protocol, and target address.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Cisco_Routing**—Perform next-hop routing. Valid values are On and Off. Default is On.
- **Cisco_Routing_Shared_Memory_Address**—Memory location of the routing table. Default is 0x35000000.
- **Cisco_Routing_Rendezvous_Name**—Rendezvous name of the database containing routing information. Default is routing_db.
- **Cisco_Routing_Rendezvous_Directory**—Location of the routing database. Default is <ServerRoot>/data.
- **Cisco_Routing_Remote_Update_Port**—Port number of the routing database server for all members of a server farm. The value for this directive must be the same for all members of a farm. Default is 22913. If you change the value of this directive, you must restart the server.
- **Cisco_Routing_Use_Domain_Routing**—Perform domain next-hop routing. This type of routing uses the host portion of the Request-URI as the key in obtaining the next one or more hops for a request. Valid values are On and Off. Default is Off.
- **Cisco_Routing_Farm_Members**—Names of Cisco SPS proxy-server farm members, excluding the local host. Specify this list on all SPS farm members, being sure to exclude the local host in its own list because it is included implicitly. This list is used by the sysadmin_sps_regroute tool. If you change the value of this directive, you must restart the server.
- **Cisco_Routing_Max_DB_Age_on_Boot**—Maximum age (in seconds) of the database-backing store file at system startup. A file whose age exceeds this value is deleted. This value must be greater than the registry ageout value. Default is 86400 (24 hours). If you change the value of this directive, you must restart the server.

The value of this directive is particularly important if you have an external routing process for keeping the database current; if you set the directive unwisely, the externally populated routes might be deleted on system startup.

- **Cisco_Routing_Global_Less_Specific_Route_Search**—Search the route database using less-specific patterns when all previously returned routes have been tried without final response or only a 5xx response. If any previously tried route has its AllowLessSpecificRoute field set to Off, Cisco SPS still stops the less-specific route search when those routes have been tried without final response or only a 5xx response. If this directive is set to Off, the value of the AllowLessSpecificRoute directive for an individual route has no effect. Valid values are On and Off. Default is Off. If you are configuring TLS, set to Off.
- **DebugFlag Routing**—Log mod-sip-routing debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Registry Services

You can configure Cisco SPS to process requests from user-agent clients (UACs) that register their location. When registry services are configured, Cisco SPS can do the following:

- Add a new registration
- Delete an existing registration
- Update an existing registration
- Delete all registrations of a user
- Return a current list of registrations of a user
- Periodically purge dated or expired registrations

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **Cisco_Registry**—Perform registry services. Valid values are On and Off. Default is On.
- **Cisco_Registry_Use_Virtual_Proxy_Host**—Function as a virtual-proxy-server host. Valid values are On and Off. Default is On.
- **Cisco_Registry_Shared_Memory_Address**—Memory location of the registration table. Default is 0x30000000.
- **Cisco_Registry_Rendezvous_Name**—Rendezvous name of the database containing registration information. Default is registry_db.
- **Cisco_Registry_Rendezvous_Directory**—Location of the registration database. Default is <ServerRoot>/data.
- **Cisco_Registry_Remote_Update_Port**—Port number of the registration database server for all server-farm members. The value for this directive must be the same for all farm members. Default is 22913. If you change the value of this directive, you must restart the server.

- **Cisco_Registry_Farm_Members**—Names or IP addresses of all remote proxy servers (other than this proxy server) that are contained within the same farm as this proxy server. This list is used by the sysadmin_sps_regroute tool.

Note You must synchronize the system clock between farm members. We recommend that you use Network Time Protocol (NTP) to do so. It provides accuracy within a millisecond on LANs and up to a few tens of milliseconds on WANs. For more information on NTP, refer to the Network Time Protocol Project website at http://www.ntp.org/.

- **Cisco_Registry_Max_DB_Age_on_Boot**—Maximum age (in seconds) of the database-backing store file at system startup. A file whose age exceeds this value is deleted. This value must be greater than the registry ageout value. Default is 86400 (24 hours). If you change this value of this directive, you must restart the server.
- **DebugFlag Registry**—Log mod_sip_registry debug messages to <ServerRoot>/logs/error_log. Valid values are On and Off. Default is Off.
- **Virtual_Proxy_Domain**—Unique DNS domain for this VirtualProxyHost. This value must be different from the value for ProxyDomain. (For information on the ProxyDomain directive, see the "Configuring Server-Core Directives" section on page B-8.) This directive is required. Examples of appropriate values for this directive are somedomain.org and foo.com.
- **Virtual_Proxy_Server_Name**—Unique name for the virtual-proxy-server host. This value must be different from the actual name of the host. This directive is optional. Examples of appropriate values for this directive are usa.somedomain.org and usa.foo.com.
- **Virtual_Proxy_Server_IP**—Unique IP address for this virtual-proxy-server host. This value must be different from the actual IP address of the host. This directive is optional. Examples of appropriate values for this directive are 10.23.2.2 and 192.168.2.2.

Step 3 Save and close the file, and restart Cisco SPS.

Configuring Virtual-Proxy-Server Hosts

You can configure a virtual-proxy-server host and define up to 10 virtual-proxy-host entries.

Detailed Steps

Step 1 Assign a unique identifier to the virtual proxy host:

```
<VirtualProxyHost ID>
```

where *ID* is the unique identifier of this VirtualProxyHost configuration.

Step 2 Set the following directives:

- **Virtual_Proxy_Domain**—Unique domain that Cisco SPS handles as VirtualProxyHost. Specify a value other than the actual domain of the proxy server.
- **Virtual_Proxy_Server_Name**—Unique server name that Cisco SPS handles as VirtualProxyHost. Specify a value other than the actual server name of the proxy server.
- **Virtual_Proxy_Server_IP**—Unique IP address that Cisco SPS handles as VirtualProxyHost. Specify a value other than actual IP address of the proxy server.

Step 3 At the end of the entry, specify the following:

```
</VirtualProxyHost>
```

Step 4 Save and close the file, and restart Cisco SPS.

Configuration Example

The following configuration example shows an entry for a virtual proxy host:

```
<VirtualProxyHost 1.1>
Virtual_Proxy_Domain foo.bar
Virtual_Proxy_Server_Name usa.foo.bar
Virtual_Proxy_Server_IP 61.12.1.1
</VirtualProxyHost>
```

Configuring H.323 RAS

You can configure communication between Cisco SPS and a H.323 gatekeeper. Communication involves sending ASN.1 encoded Registration, Admission, and Status Protocol (RAS) LRQ messages to a provisioned H.323 gatekeeper and receiving LCF, LR,J or RIP messages from the gatekeeper.

Detailed Steps

Step 1 In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

Step 2 Set the following directives as needed:

- **RASModule**—Enable the RAS module. Valid values are On and Off. Default is Off.

- **RASAcceptLCF**—LCF message to accept. Valid values are First (take the first valid LCF message from any gatekeeper) and Best (wait for the best LCF message within the LRQ time window; "best" means from the gateway that has the lowest cost and highest priority).
- **RASTimeoutInterval**—Time (in milliseconds) to wait for a single response from a gatekeeper. If the timeout expires, Cisco SPS tries another gatekeeper within the same cluster. You must set this directive if you set the RASLRQMethod directive to Sequential. Default is 300.
- **RASLRQMethod**—Method used to send LRQ messages to a gatekeeper. Valid values are Sequential (Cisco SPS sends LRQs sequentially and waits for a response in the RASTimeoutInterval time) and Blast (SPS sends LRQs in parallel before detecting any response).
- **RASLRQWindow**—Maximum time (in milliseconds) to wait for responses from the gatekeepers contacted. Default is 3000. The RIP message can override this value.
- **RASTimeToLive**—Time-to-live (TTL) value (in hops) in the RAS LRQ nonstandard message body. Default is 6.
- **RASAllowTranslation**—Set the canMapAlias field in the LRQ message. Valid values are the following:
  - On—Set to True. The gatekeeper replaces the dialed phone number or destinationInfo field.
  - Off—Set to False. The gatekeeper does not replace address information.

  If the gatekeeper replaces address information and the value of the canMapAlias field is False, the gatekeeper rejects the LRQ.
- **RASGateKeeperCluster**—Set priorities for gatekeeper clusters so that Cisco SPS can query clusters in priority order. For each gatekeeper, you must specify the IP address and port number. Valid priority values are from 1 to 65535.

  Example

  The following example sets up two clusters, each containing two gatekeepers that use port 1719. Priority values are 1 and 2 respectively.

```
<RASGatekeeperCluster 1>
RASGatekeeper gatekeeper1.company.com 1719
RASGatekeeper gatekeeper2.company.com 1719
</RASGatekeeperCluster>

<RASGatekeeperCluster 2>
RASGatekeeper gatekeeper3.company.com 1719
RASGatekeeper gatekeeper4.company.com 1719
</RASGatekeeperCluster>
```

- **RASGateKeeper**—IP address (or FQDN) and port number of each individual gatekeeper in a gatekeeper cluster. Maximum number of gatekeepers in a cluster is 5.
- **RASDefaultTechPrefixAction**—Default action to take to an outgoing INVITE request or 302 message when a technology prefix exists and no specific local rule applies. Valid values are Strip (remove the prefix) and Include (include the prefix).
- **RASTechPrefix**—Technology prefix to use when the dialed number matches the specified number pattern.

  Example

```
RASTechPrefix 1919321... 001# INCLUDE
RASTechPrefix 1919456... 002# STRIP
```

- **RASTransportType**—Transport type to use in the route entry that Cisco SPS learns from a gatekeeper via RAS. Valid values are TCP, TLS, and UDP. Default is UDP.

- **DebugFlag RasAPI**—Log RasAPI debug messages to <ServerRoot>/logs/error_log. These messages explain how Cisco SPS handles incoming and outgoing RasAPI messages that concern RAS encoding/decoding and sending/receiving. Valid values are On and Off. Default is Off.
- **DebugFlag RAS**—Log RAS debug messages to <ServerRoot>/logs/error_log. These messages explain how Cisco SPS handles incoming and outgoing RAS messages that concern module-specific tasks such as module configuration, socket creation, message conversion, and routes insertion. Valid values are On and Off. Default is Off.

Step 3 Save and close the file, and restart Cisco SPS.

How to Configure the SIP Proxy Server in a Farm

You can configure registry and routing farms to contain different sets of members, as long as you ensure that configuration is consistent across all farm members.

For example, suppose that your registry contains two members (host1, host2) and the routing farm contains only the local member. You configure the Cisco_Registry_Farm_Members directive on each member and remove the comment marker for the Cisco_Routing_Farm_Members directive on each member.

Note

- Designate one farm member as master and the other as slave. Use the sysadmin_ csps_regroute tool on the master to load any initial registry or routing seed files for the farm and to perform any registry or routing updates for the farm. Do not use the tool on a slave to seed or update registry or routing information. Initially, the master synchronizes any slaves. After that, members synchronize with each other whenever an update occurs.
- Synchronize the system clock between farm members. We recommend that you use Network Time Protocol (NTP) to do so. It provides accuracy within a millisecond on LANs and up to a few tens of milliseconds on WANs. For more information on NTP, refer to the Network Time Protocol Project website at http://www.ntp.org/.

Detailed Steps

Step 1 Ensure that the IP network connecting farm members is working.

Step 2 Edit the configuration file as follows:

a. In the (Linux) /usr/local/sip/conf/ or (Solaris) /opt/sip/conf directory, open the sipd.conf file using a text editor such as vi:

```
# vi sipd.conf
```

b. Set the following directives:

- **Cisco_Registry_Farm_Members**—Hostnames of all servers (excluding the local server)
- **Cisco_Routing_Farm_Members**—Hostnames of all servers (excluding the local server)

Example

The following shows directive settings for each of two servers in a farm (host1.cisco.com and host2.cisco.com}.

Member	Directive Setting
Registry Farm	
host1.cisco.com	`Cisco_Registry_Farm_Members "host2.cisco.com"`
host2.cisco.com	`Cisco_Registry_Farm_Members "host1.cisco.com"`
Routing Farm	
host1.cisco.com	`Cisco_Routing_Farm_Members "host2.cisco.com"`
host2.cisco.com	`Cisco_Routing_Farm_Members "host1.cisco.com"`

c. Save and close the file.

Step 3 Configure the master server:

a. Prepare two seed data files for registry and routing separately for Cisco SPS.

Note For additional information, see the template files sip_registry.conf-dist and sip_routing.conf-dist.

For upgrades from Cisco SPS 1.0 or Cisco SPS 1.1, copy static entries from sipd.conf to two separate files, one each for registry and routing.

b. Use the sysadmin_sps_regroute tool to import the files, or use the provisioning-system GUI to seed the registry and routing entries.

Note Use seed data only to start the master member for the first time or when existing registry or routing databases are invalid. Seed data is not required between normal shutdown and start. A database is invalid when it contains corrupted data or when it has expired. If a database becomes get corrupted, do the following: 1. Stop Cisco SPS. 2. Remove the database files (registry_db or routing_db in the (Linux) usr/local/sip/logs or (Solaris) opt/sip/logs directory). 3. Start Cisco SPS. 4. Seed the database. If the database has expired, the farm member automatically clears its old database and gets the most current database from another farm member.

Step 4 Configure slave servers as needed.

Note Do not place any seed entries in the default template files or sipd.conf, and do not use the sysadmin_sps_regroute tool on a slave server to update a slave database. For more information on slave servers, see the sysadmin_sps_regroute tool.

Step 5 Synchronize the system clock among all farm members.

Step 6 Periodically back up the registry and routing database files (use the sysadmin_sps_regroute tool to export their contents). You can then use the database files as seed data files if the databases later become corrupted.

Note For more information on system backups and restores, see the "How to Back Up and Restore Cisco SPS" section on page 3-8.

How to Configure IPSec

You can configure IP security (IPSec) on your Cisco SPS.

Note You can configure IPSec only if you are a system administrator.

IPSec is a suite of security protocols that secure and encrypt communication channels and ensure that only authorized parties can communicate on those channels. It enables you to restrict inbound and outbound communication on a port-by-port basis and to offer authenticated hosts different levels of access.

IPSec provides the following optional network-security services. In general, your local security policy determines which services you should use:

- Data confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data-origin authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service depends on the data integrity service.
- Anti-replay—The IPSec receiver can detect and reject replayed packets.

Note Data-origin authentication (also sometimes called data authentication or data integrity) is understood in this document to include anti-replay services, unless otherwise specified.

Cisco SPS IPSec is supported on network configurations as shown in Table B-3.

Table B-3 Network Configurations That Support Cisco SPS IPSec

Network Configuration	Means of Support
Solaris system to Solaris system	Manual keying
Linux system to Linux system	Any of the following: • Manual keying • IKE via configuration files • IKE via mod_ipsec_auto.c
Solaris system to Linux system	Manual keying

Cisco SPS IPSec requires authentication and data encryption as follows:

- Authentication is installed as part of the Solaris Operating Environment 2.8.
- Data encryption is available on a Solaris supplemental CD or by download without charge from the Sun Solaris website at http://wwws.sun.com/software/solaris/encryption/.

Detailed Steps

Step 1 Verify that data encryption is installed by checking to see if the following two files exist:

- /kernel/strmod/encrdes
- /kernel/strmod/encr3des

Step 2 Configure IPSec for Cisco SPS on a Solaris platform:

a. Configure the system security policy.

b. Install the authentication and data-encryption security keys.

APPENDIX C

Manual Operation and Maintenance

This appendix describes how to manually operate and maintain the Cisco SIP proxy server (Cisco SPS).

Caution You can perform all of the operations described in this appendix using the Cisco SPS GUI-based provisioning system as described in Chapter 3, "Operating and Maintaining Cisco SPS." Unless your situation is highly unusual, you should do so.

Contents

How to Manage Cisco SPS Licenses

Cisco SPS licenses are of two types: evaluation and permanent. An evaluation license has an expiration date. A permanent license has no expiration date.

Your license is delivered to you as a license key. A license key is a sequence of text characters that Cisco SPS must read and validate at startup before it can run.

Note To add a license, refer to the *Cisco SIP Proxy Server Version 2.2 Installation Guide*.

Detailed Steps

Step 1 Open the Cisco SPS licensing window.

Step 2 Run the license GUI.

Note If, during installation, you chose the "Complete" option, the license GUI "License" was installed along with the Cisco SPS GUI. If, instead, you chose the "Minimal" option, run the installation again and choose "Complete." Install the license GUI in whatever directory you choose; the default location is (Linux) /usr/local/sip/gui or (Solaris) /opt/sip/gui.

Step 3 Enter the password (default is cspsuser) and do one of the following:

- From the pserver, click **OK**.
- From other than the pserver, click **more>>**, enter the pserver host name and port number, and click **OK**.

The licensing window appears. The pserver host-name and port-number entries are automatically saved and reappear at each login.

Step 4 Reformat the licensing window as needed:

- To hide a column: Click **Column** on top of the window and follow instructions.
- To change a column width: Place the cursor on a vertical line dividing column headers and drag the cursor.
- To change column order: Place the cursor on a header and drag it to a new position.

Step 5 To display a license, do the following:

a. In the top righthand side of the licensing window, click the down arrows to select search query and operators.

b. Enter a search string. Do not include quotation marks.

c. Click **Search**.

Step 6 To upgrade a license, do the following:

a. Copy the new license key, in preparation for pasting in a subsequent step. Do not include quotation marks around the license key.

b. Select the license that you wish to upgrade.

c. Click **Edit**.

d. Paste the new license key over the old license key.

e. Edit license information and settings for the new license.

f. Click **Submit > OK.**

Step 7 To delete a license, do the following:

a. Select the license.

b. Click **Delete**.

Step 8 When done, click **Operations > Load All.**

Tip

- Troubleshoot any problems by checking for license-validation debug messages in the error-log (error_log) file.

- Be cautious if you cut and paste a license key from one system to another. You might introduce an incorrect end-of-line character sequence that prevents the system from recognizing the key.

How to Start and Stop Cisco SPS

This section contains information about the following:

You can start, stop, restart, and gracefully restart Cisco SPS by means of the sip script. This script is created when you install Cisco SPS using the SPS setup (sps_setup) script.

Caution Scripts to start and stop each individual component exist in the same directory. Do not use the component scripts individually. Rather, use the sip script to invoke component scripts in the appropriate order and with the necessary precondition checking.

Note

- Before your start, make a note of the directory that contains the sip script. When a step instructs you to access the sip script, this is the directory to which you go to do so. The default location is as follows:

  Linux: `/usr/local/sip/bin`

  Solaris: `/opt/sip/bin`

- If errors occur when Cisco SPS starts, stops, restarts, or gracefully restarts, error messages display (Linux) in /var/log/messages or (Solaris) onscreen. For details about the log files for each process, see Chapter 3, "Operating and Maintaining Cisco SPS."
- The sip script provides the functions in the sipdctl script in Cisco SPS 1.*x*.

Starting Cisco SPS

Detailed Steps

Tip For sample screen and log outputs for these steps, see the "Screen and Log Output: Starting Cisco SPS" section on page C-7.

Step 1 Run the sip script with the start argument:

Linux: `# /usr/local/sip/bin sip start`

Solaris: `# /opt/sip/bin sip start`

Step 2 Verify that Cisco SPS is running properly by viewing all Cisco SPS processes:

```
# ps -ef | grep sps
```

You should have the number of processes shown in Table C-1.

Table C-1 Cisco SPS Processes

Type of Process	Number of Processes	
	Linux	Solaris
Provisioning server (pserver)	1	1
License manager (licenseMgr)	1	1
SIP provisioning agent (spa)	1	1
sipd	7 (default)	7 (default)

Stopping Cisco SPS

Detailed Steps

Tip For sample screen and log outputs for these steps, see the "Screen and Log Output: Stopping Cisco SPS" section on page C-7.

Step 1 Run the sip script with the stop argument:

Linux: `# /usr/local/sip/bin sip stop`

Solaris: `# /opt/sip/bin sip stop`

All Cisco SPS processes stop and the server can no longer process calls.

Step 2 Verify that Cisco SPS processes are stopped:

Linux: `# /usr/local/sip/bin ps -ef | grep sps`

Solaris: `# /opt/sip/bin ps -ef | grep sps`

Step 3 If any of the following processes still are running, stop them manually by using the UNIX command **kill**:

- SIP proxy server (sipd)
- SIP provisioning agent (spa)
- License manager (licenseMgr)
- Provisioning server (pserver)

Restarting Cisco SPS

Detailed Steps

Tip For sample screen and log outputs for these steps, see the "Screen and Log Output: Restarting Cisco SPS" section on page C-8.

Step 1 Run the sip script with the restart argument:

Linux: `# /usr/local/sip/bin sip restart`

Solaris: `# /opt/sip/bin sip restart`

Step 2 Verify that Cisco SPS is running properly by viewing all Cisco SPS processes:

```
# ps -ef | grep sps
```

You should have the number of processes shown in Table C-1.

Note Process IDs that display upon system startup differ from those in effect beforehand. During the time between stop and start, the server cannot process calls.

Gracefully Restarting Cisco SPS

Graceful restart provides a mechanism to prompt spa to write a new sipd.conf file. During a graceful restart, the sipd daemon (parent process) remains alive, rereads the configuration file (sipd.conf), tears down child sipd processes as they become idle, and spawns new child processes with the new configuration. Call processing is not interrupted as a result.

If a server configuration changes, perform a graceful restart to activate the new configuration without dropping calls.

If a Cisco SPS TCP I/O process becomes unresponsive, the parent sipd performs its own graceful restart (up to five times) to activate the process.

Detailed Steps

Tip For sample screen and log outputs for these steps, see the "Screen and Log Output: Gracefully Restarting Cisco SPS" section on page C-9.

Step 1 Run the sip script with the graceful argument:

Linux: `# /usr/local/sip/bin sip graceful`

Solaris: `# /opt/sip/bin sip graceful`

Step 2 Verify that Cisco SPS is running properly by viewing all Cisco SPS processes (the number of processes should be as in Table C-1):

```
# ps -ef | grep sps
```

Step 3 If the TCP I/O process fails to activate, wait 1 minute and perform a graceful restart manually.

Transaction Gracefully Restarting Cisco SPS

Transaction Graceful restart provides a mechanism to restart SPS with minimal impact on the ongoing transactions. During a transaction graceful restart, the sipd daemon (parent process) waits for some specific time (specified while executing the command) to allow any ongoing transactions to complete. Any new INVITE or REGISTER requests during this time (when the command is fired till the SPS is brought down, as part of the processing for this command) is responded in either of the following two manners, depending upon the options chosen while executing the command.

- “503 Service Unavailable" is sent for each new INVITE or REGISTER request.
- No response, not even "100 Trying" is sent for each new INVITE or REGISTER request.

Detailed Steps

Tip For sample screen and log outputs for these steps, see the “Screen and Log Output: Transaction Gracefully Restarting CISCO SPS” section on page C-10

Step 1 Run the SIP script with the transaction graceful argument:

```
Solaris: # /opt/sip/bin/sip txngraceful-restart txngraceful-restart-timer
txngraceful-restart-behavior
txngraceful-restart-timer in seconds e.g 30 for 30 seconds
```

txngraceful-restart-behavior can have value 1 to start sending '503 Service Unavailable' or 2 to stop sending '100 Trying'

When you execute this commad ,CSPS goes down after txngraceful-restart-timer seconds and automatically comes back again. Until CSPS goes down, any new INVITE or REGISTER requests either respond with '503 Service Unavailable' or do not send '100 Trying' depending on the value of txngraceful-restart-behavior .

Step 2 Verify that Cisco SPS is running properly after the txngraceful-restart-timer by viewing all Cisco SPS processes (the number of processes should be as in Table C-1):

```
# ps -ef | grep sps
```

Note When value 1 is used as txngraceful-restart-behavior, the 503 'Service Unavailable' responses contain the 'retry-after' header, with value 60 seconds. You can change this default value by changing the following:

Filename : /opt/sip/bin/sipdctl file.
Parameter: RESTART_TIMER

The Transaction Graceful Restarting applies only when the Add Record Route Header option is switched off.
The services 'Call Forward Busy (CFB)' and 'Call Forward No Answer' will not work after the 'txngreaceful-restart' command is fired until the CSPS automatically comes back again.

Output Examples

This section contains the following output examples:

Screen and Log Output: Starting Cisco SPS

Start Screen Output

Upon startup, the screen displays output similar to the following:

```
Starting pserver:                                          [  OK  ]
Starting license manager:                                  [  OK  ]
Starting spa:                                              [  OK  ]
Starting sipd:                                             [  OK  ]
```

Start-Verification Screen Output

Upon system-start verification, the screen displays output similar to the following. In this example, the first sipd process, with parent process ID 1, is the parent sipd. The other sipd processes are the TCP I/O process and five child processes. Sip_Services is an additional process required to maintain synchronization among local and remote farm members.

```
csps      4040     1  0 17:38 ?        00:00:00 /usr/local/sip/bin/pserver -c /u
csps      4054     1  0 17:38 ?        00:00:00 /usr/local/sip/bin/licenseMgr /u
csps      4068     1  0 17:38 ?        00:00:00 /usr/local/sip/bin/spa /usr/loca
csps      4074     1  0 17:38 ?        00:00:00 /usr/local/sip/bin/Sip_Services
csps      4092     1  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4094  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4096  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4097  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4100  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4101  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4102  4092  0 17:38 pts/1    00:00:00 /usr/local/sip/bin/sipd
root      4107  1387  0 17:39 pts/1    00:00:00 grep sps
```

Screen and Log Output: Stopping Cisco SPS

Stop Screen Output

Upon system stop, the screen displays output similar to the following:

```
Stopping sipd:                                             [  OK  ]
Stopping spa:                                              [  OK  ]
Stopping license manager:                                  [  OK  ]
Stopping pserver:                                          [  OK  ]
```

Stop Log Output

Upon system stop, the log (Linux /var/log/messages file or Solaris screen) displays output similar to the following:

```
sipdctl: Waiting for process to stop.
sipdctl: .
sipdctl: /usr/local/sip/bin/sipdctl stop: sipd stopped
sipdctl: Waiting for process to stop.
sipdctl: .
sipdctl: /usr/local/sip/bin/sipdctl stop: Sip_Services stopped
sip: Stopping sipd:  succeeded
spactl: Waiting for process to stop.
spactl: .
spactl: /usr/local/sip/bin/spactl stop: spa stopped
sip: Stopping spa:  succeeded
lmctl: Waiting for process to stop.
lmctl: .
lmctl: /usr/local/sip/bin/lmctl stop: licenseMgr stopped
sip: Stopping license manager:  succeeded
pserverctl: Waiting for process to stop.
pserverctl: .
pserverctl: /usr/local/sip/bin/pserverctl stop: pserver stopped
sip: Stopping pserver:  succeeded
```

Stop-Verification Screen Output

Upon system-stop verification, the screen displays output similar to the following:

```
csps 16421 15876 0 09:47 pts/0 00:00:00 grep sps
```

Screen and Log Output: Restarting Cisco SPS

Restart Screen Output

Upon system restart, the screen displays output similar to the following:

```
Stopping sipd:                                             [  OK  ]
Stopping spa:                                              [  OK  ]
Stopping license manager:                                  [  OK  ]
Stopping pserver:                                          [  OK  ]
Starting pserver:                                          [  OK  ]
Starting license manager:                                  [  OK  ]
Starting spa:                                              [  OK  ]
Starting sipd:                                             [  OK  ]
```

Restart Log Output

Upon system restart, the log (Linux /var/log/messages file or Solaris screen) displays output similar to the following:

```
sipdctl: Waiting for process to stop.
sipdctl: .
sipdctl: /usr/local/sip/bin/sipdctl stop: sipd stopped
sipdctl: Waiting for process to stop.
sipdctl: .
sipdctl: /usr/local/sip/bin/sipdctl stop: Sip_Services stopped
sip: Stopping sipd:  succeeded
spactl: Waiting for process to stop.
spactl: .
spactl: /usr/local/sip/bin/spactl stop: spa stopped
sip: Stopping spa:  succeeded
lmctl: Waiting for process to stop.
lmctl: .
```

```
lmctl: /usr/local/sip/bin/lmctl stop: licenseMgr stopped
sip: Stopping license manager:  succeeded
pserverctl: Waiting for process to stop.
pserverctl: .
pserverctl: /usr/local/sip/bin/pserverctl stop: pserver stopped
sip: Stopping pserver:  succeeded
pserverctl: /usr/local/sip/bin/pserverctl start: pserver started
sip: Starting pserver:  succeeded
lmctl: /usr/local/sip/bin/lmctl start: licenseMgr started
sip: Starting license manager:  succeeded
spactl: /usr/local/sip/bin/spactl start: spa started
spactl: /usr/local/sip/bin/spactl start: Waiting for sipd.conf from spa..
spactl: .
spactl: /usr/local/sip/bin/spactl start: sipd.conf written
sip: Starting spa:  succeeded
sipdctl: Version of CSPS        : 2.2.x.x
sipdctl: Version in Config file : 2.2.x.x
sipdctl: Software release version of CSPS validated successfully with your license
sipdctl: License validated successfully
sipdctl: This is Permanent license, with Infrastructure functionality
sipdctl: /usr/local/sip/bin/sipdctl start: sipd started
sip: Starting sipd:  succeeded
```

Restart-Verification Screen Output

Upon system-restart verification, the screen displays output similar to the following:

```
[/usr/local/sip/bin]# ps -ef | grep sps
csps      4216     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/pserver -c /u
csps      4225     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/licenseMgr /u
csps      4241     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/spa /usr/loca
csps      4245     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/Sip_Services
csps      4264     1  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4266  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4268  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4269  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4270  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4271  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4276  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
root      4279  1387  0 18:00 pts/1    00:00:00 grep sps
```

Screen and Log Output: Gracefully Restarting Cisco SPS

Graceful-Restart Screen Output

Upon graceful restart, the screen displays output similar to the following:

```
Gracefully restarting pserver:                              [  OK  ]
Gracefully restarting license manager:                      [  OK  ]
Gracefully restarting spa:                                  [  OK  ]
Gracefully restarting sipd:                                 [  OK  ]
```

Graceful-Restart Log Output

Upon graceful restart, the log (Linux /var/log/messages file or Solaris screen) displays output similar to the following:

```
pserverctl: /usr/local/sip/bin/pserverctl graceful: pserver (pid 3749 3769 3770 3775)
already running
sip: Gracefully restarting pserver:  succeeded
lmctl: /usr/local/sip/bin/lmctl graceful: licenseMgr (pid 3764 3766 3767) already running
sip: Gracefully restarting license manager:  succeeded
spactl: Waiting for process to stop.
spactl: .
```

```
spactl: /usr/local/sip/bin/spactl stop: spa stopped
spactl: Wait 3 seconds before restarting the application...
spactl: /usr/local/sip/bin/spactl start: spa started
spactl: /usr/local/sip/bin/spactl start: Waiting for sipd.conf from spa..
spactl: .
spactl: /usr/local/sip/bin/spactl start: sipd.conf written
sip: Gracefully restarting spa:  succeeded
sipdctl: /usr/local/sip/bin/sipdctl graceful: sipd gracefully restarted
sip: Gracefully restarting sipd:  succeeded
```

Graceful-Restart-Verification Screen Output

Upon graceful-restart verification, the screen displays output similar to the following. In this example, the original pserver and licenseMgr processes are not affected. The spa processes are restarted to force the writing of a new SIP directives (sipd.conf) file. The parent sipd process, the original Sip_Services, and the TCP I/O sipd process remain the same as for the previous start of the server. All other sipd child processes have been restarted and have new process IDs.

```
csps      4216     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/pserver -c /u
csps      4225     1  0 17:58 ?        00:00:00 /usr/local/sip/bin/licenseMgr /u
csps      4264     1  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4266  4264  0 17:58 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4334     1  0 18:02 ?        00:00:00 /usr/local/sip/bin/spa /usr/loca
csps      4337     1  0 18:02 ?        00:00:00 /usr/local/sip/bin/Sip_Services
csps      4357  4264  0 18:02 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4358  4264  0 18:02 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4359  4264  0 18:02 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4360  4264  0 18:02 pts/1    00:00:00 /usr/local/sip/bin/sipd
csps      4361  4264  0 18:02 pts/1    00:00:00 /usr/local/sip/bin/sipd
root      4370  1387  0 18:13 pts/1    00:00:00 grep sps
```

Screen and Log Output: Transaction Gracefully Restarting CISCO SPS

```
./sip txngraceful-restart  45 2
Gracefully Stopping sipd...
/opt/sip/bin/sipdctl txngraceful-restart: sipd  signalled to gracefully stop after 45
seconds
/opt/sip/bin/sipdctl txngraceful-restart: Going to sleep for 45 seconds ,SPS will shutdown
after this
Stopping spa...
Waiting for process to stop.....
/opt/sip/bin/spactl stop: spa stopped
Stopping license manager...
Waiting for process to stop....
/opt/sip/bin/lmctl stop: licenseMgr stopped
Stopping pserver...
Waiting for process to stop.
/opt/sip/bin/pserverctl stop: pserver stopped
 Now starting CSPS
Starting pserver...
/opt/sip/bin/pserverctl start: pserver started
Starting license manager...
/opt/sip/bin/lmctl start: licenseMgr started
Starting spa...

/opt/sip/bin/spactl start: spa started
/opt/sip/bin/spactl start: Waiting for sipd.conf from spa...
/opt/sip/bin/spactl start: sipd.conf written
Starting sipd...
```

Transaction Graceful-Restart Log Output:

Upon txn-graceful restart, the log (Linux /var/log/messages file or Solaris screen) displays output similar to the following:

```
Now starting CSPS
Starting pserver...
/opt/sip/bin/pserverctl start: pserver started
Starting license manager...
/opt/sip/bin/lmctl start: licenseMgr started
Starting spa...

/opt/sip/bin/spactl start: spa started
/opt/sip/bin/spactl start: Waiting for sipd.conf from spa...
/opt/sip/bin/spactl start: sipd.conf written
Starting sipd...
```

Transaction Graceful-Restart-Verification Screen Output

Upon transaction graceful-restart verification, the screen displays output similar to the following.

```
csps 17950 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
    csps 17948 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
    csps 17911     1  0 04:02:35 ?        0:00 /opt/sip/bin/licenseMgr
/opt/sip/conf/lm.conf
    csps 17947     1  3 04:02:38 pts/1    0:01 /opt/sip/bin/sipd
    csps  1568     1  2   Aug 01 ?        2:34 /opt/sip/bin/Sip_Services -z -y 52931
    csps 17903     1  2 04:02:34 ?        0:01 /opt/sip/bin/pserver -c
/opt/sip/conf/ps.conf
    csps 17952 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
    csps 17953 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
    csps 17951 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
    csps 17921     1  1 04:02:35 ?        0:00 /opt/sip/bin/spa /opt/sip/conf/spa.conf
    csps 17949 17947  0 04:02:39 pts/1    0:00 /opt/sip/bin/sipd
```

How to Change the MySQL Password

User and root passwords are set when you run the SPS setup (sps_setup) script. You can, however, change the root password. If a use has forgotten a password, assign a new one.

Detailed Steps

Step 1 Log in to the MySQL database.

Step 2 Enter the following commands:

Linus:
```
/usr/local/mysql/bin/safe_mysqld --user=mysql &
/usr/local/mysql/bin/mysqladmin -u root -p<old_password> password <new_password>
/usr/local/mysql/bin/mysqladmin -p reload
```

Solaris:
```
/opt/mysql/bin/safe_mysqld --user=mysql &
/opt/mysql/bin/mysqladmin -u root -p<old_password> password <new_password>
/opt/mysql/bin/mysqladmin -p reload
```

How to Manage Cisco SPS Databases

Two Cisco SIP proxy server (Cisco SPS) database administration tools exist, as described in Table C-2.

Table C-2 Cisco SPS Database Administration Tools

Database	Tool
Registry database	Registry and routing (regroute) databases tool
Routing database	
MySQL server database	MySQL database tool

This section contains the following information:

Using the Regroute Databases Tool

The regroute tool allows you to add, delete, or modify data in the registry and routing databases without interrupting proxy-server operation. The following tasks are possible:

Caution Do not use multiple copies of this tool.

Note

- If you use the GUI-based provisioning system, do not use this tool to add, modify, or delete data. You can use the tool to view data such as dynamic registrations that the GUI cannot provide or to import files containing static registration or routing entries.
- Make a note of the tool location. When a step instructs you to access the tool, this is the directory to which you go in order to do so. The default location is as follows:

  Linux: `/usr/local/sip/bin/sysadmin_sps_regroute`

  Solaris: `/opt/sip/bin/sysadmin_sps_regroute`

- This tool operates by means of a series of menus. To select a menu option, type the letter that precedes it—in either uppercase or lowercase letters—and press **Enter**.

  Some options require additional information, such as a subscriber ID or a URL. The specific type of entry required is shown within brackets <>. For example, the option I, to import a configuration file, contains the notation “configuration <file>” to indicate that you must follow the selection I with the configuration filename.

Most options indicate a default value. You can select the default by typing either the indicated value or the wildcard character *. For example, to add data to the registry database, you must enter a value for User Type. The default value is Phone. You can type Phone or simply *.

Activating the Regroute Tool

Command Summary

sysadmin_sps_regroute [-m {**routing** | **registry** | **both**}**] [-l] [-i** *file*] **[-l** *file*] **[-x** *file*] **[-p** *file*] **[-h** *primary-host*] **[-j** *secondary-host*] **[-H** *primary-port*] **[-J** *secondary-port*] **[-a] [-A** *routing-database-address*] **[-B** *registry-database-address*] **[-S** *routing-database-name*] **[-R** *registry-database-name*] **[-T** *token-port*] **[-L** *directory*] **[-U** *routing-port*] **[-V** *registry-port*] **[-D]**

Note The tool automatically ends after you enter the **-l**, **-i**, **-l**, or **-x** keyword.

Detailed Steps

Step 1 Navigate to the directory where the regroute tool resides.

Step 2 Type **sysadmin_sps_regroute**.

Step 3 (Optional) Append one or more of the following keywords and press **Enter**:

General Keywords	
-m {**routing** | **registry** | **both**}	Type of data: routing data, registry data, or both (you can also enter 1, 2, or 3 respectively). Use in conjunction with **-l**, **-i**, **-l**, and **-x**. Default is both (3).
-l	List all data entries specified by **-m**.
-i *file*	Import comma-separated data (of type specified by **-m**) from the specified configuration file.
-l *file*	Import Cisco SPS 2.0 data (of type specified by **-m**) from the specified configuration file.
-x *file*	Export comma-separated data (of type specified by **-m**) to the specified configuration file. File format is the standard stanza format of a SIP directives (sipd) configuration file for routing and registry data.
-p *file*	Absolute path and filename of the sipd.conf configuration file.
-h *primary-host*	Hostname or IP address of the primary provisioning server (pserver).
-j *secondary-host*	Hostname or IP address of the secondary pserver.
-H *primary-port*	Port to use for the primary pserver.
-J *secondary-port*	Port to use for the secondary pserver.
Expert Keywords (use with care)	
-a	Connect to shared memory rather than the pserver.
-A *routing-database-address*	Routing-database shared-memory address.
-B *registry-database-address*	Registry-database shared-memory address.

-S *routing-database-name*	Routing-database name.
-R *registry-database-name*	Registry-database name.
-T *token-port*	Token port.
-L *directory*	Shared-memory database directory.
-U *routing-port*	Routing port.
-V *registry-port*	Registry port.
-D	Enable debugging.

Examples

```
sysadmin_sps_regroute -i xyz.conf

sysadmin_sps_regroute -x xyz.conf -l

sysadmin_sps_regroute -i abc.conf -x xyz.conf -m registry

sysadmin_sps_regroute -p /opt/sip/xyz.conf
```

Managing Databases

Detailed Steps

Step 1 Open the regroute tool Routing and Registry Databases main menu (see the "Activating the Regroute Tool" section on page C-13).

Step 2 To select a database, do the following:

a. Select **S** (select registry or routing database).

b. Select a database:

- **Y** (registry database)
- **Z** (routing database)

The text line under the menu title indicates which database is selected.

c. Select **M** to return to the main menu.

Step 3 To add or delete a database, do the following:

a. Select **D** (query, add to, or delete from the database).

b. Select one of the following:

- **A** (add an entry to the database)
- **D** (delete an entry from the database)

c. Select **E** (enter the registry user ID) and enter required data as prompted.

Step 4 To search a database, do the following:

a. Select **D** (query, add to, or delete from the database).

b. Select **S** (search the database).

c. Select **E** (enter the destination pattern) and enter the required data as prompted.

Step 5 To display the contents of a database, do the following:

a. Select **D** (query, add to, or delete from the database).

b. Select **L** (list everything in the database).

Step 6 To display database-memory information, do the following:

a. Select **S** (select registry or routing database).

b. Select **D** (display the shared memory and database information). A status message appears.

Step 7 To exit, select one of the following:

- **M** (return to the main menu)
- **P** (return to the previous menu)
- **Q** (exit from the tool)

Importing and Exporting Configuration Files and Databases

Importing the content of a configuration file to a database allows you to update both the registry and routing databases with the configuration file content as long as the system finds matching entries.

Exporting the content of a database to a configuration file appends the database to the specified file. Entries are exported from the database (routing or registry) as specified by the **-m** option.

Detailed Steps

Step 1 Open the regroute tool Routing and Registry Databases main menu (see the "Activating the Regroute Tool" section on page C-13).

Step 2 Select one of the following options:

- **I** (import a configuration with route/registry entries)
- **X** (export current database entries to a configuration)

Step 3 Enter the configuration filename. When import or export is complete, a status message appears.

Step 4 To exit, select one of the following options:

- **M** (return to the main menu)
- **P** (return to the previous menu)
- **Q** (exit from the tool)

Using the MySQL Database Tool

The MySQL database tool allows you to modify a MySQL server on a local or remote system. The following tasks are possible:

- Changing Information About Subscribers, page C-17
- Removing Subscribers, page C-17

Note

- If you run this tool remotely, the remote MySQL server need not be a Linux or Solaris system. This tool has been successfully tested on Redhat 7.1 and Solaris 2.8 platforms.
- This tool operates by means of a series of menus. To select a menu option, type the letter that precedes it—in either uppercase or lowercase letters—and press **Enter**.

  Some options require additional information, such as a subscriber ID or a URL. The specific type of entry required is shown within brackets <>. For example, the option I, to import a configuration file, contains the notation "configuration <file>" to indicate that you must follow the selection I with the configuration filename.

  Most options indicate a default value. You can select the default by typing either the indicated value or the wildcard character *. For example, to add data to the registry database, you must enter a value for User Type. The default value is Phone. You can type Phone or simply *.

Activating the MySQL Database Tool

Detailed Steps

Step 1 Log in to Cisco SPS as root.

Step 2 Run the sysadmin MySQL user script:

Linux: `# /usr/local/sip/bin/sysadmin_mysql_user`

Solaris: `# /opt/sip/bin/sysadmin_mysql_user`

Step 3 Enter the host name, username, and password as directed.

Step 4 Enter the database name and table name. The MySQL Database main menu appears.

Displaying Information About Subscribers

Detailed Steps

Step 1 Open the MySQL Database main menu (see the "Activating the MySQL Database Tool" section on page C-16).

Step 2 To display information about a single subscriber, select **S** (show subscriber) and enter a subscriber ID as prompted.

Step 3 To display a list of all subscribers with or without details on each, do the following:

a. Select **L** (list all subscribers).

b. Select the desired level of detail:

- **Y** (details for each subscriber ID)
- **N** (summary list of subscriber IDs)

Step 4 Repeat from Step 2 as needed.

Step 5 To exit, select **X**.

Adding Subscribers

Detailed Steps

Step 1 Open the MySQL Database main menu (see the Activating the MySQL Database Tool, page C-16).

Step 2 Select **A** (add subscriber) and enter a subscriber ID and domain name as prompted.

Step 3 To assign a password, select **P** (password) and enter a password for the subscriber as prompted.

Step 4 To assign call-forwarding, select one of the following:

- **B** (call forward busy)
- **N** (call forward no answer)
- **U** (call forward unconditional)
- **V** (call forward unavailable)

Step 5 Enter the appropriate URL as prompted.

Step 6 Repeat from Step 2 as needed.

Step 7 To exit, select **X**.

Changing Information About Subscribers

Detailed Steps

Step 1 Open the MySQL Database main menu (see the "Activating the MySQL Database Tool" section on page C-16).

Step 2 Select **M** (modify subscriber) and enter a subscriber ID as prompted.

Step 3 Modify the attributes by selecting the appropriate options in the menu and enter new data as prompted. If a system user has forgotten a password, assign a new one.

Step 4 Repeat from Step 2 as needed.

Step 5 To exit, select **X**.

Removing Subscribers

Detailed Steps

Step 1 Open the MySQL Database main menu (see the "Activating the MySQL Database Tool" section on page C-16).

Step 2 Select **R** (remove subscriber) and enter a subscriber ID and domain name as prompted.

When removal is complete, a status message appears. The list of subscribers appears without the record that you just removed.

Repeat from Step 2 as needed.

Step 3 To exit, select **X**.

Sample Error Messages

Note These messages, plus additional troubleshooting information, appear in Chapter 5, "Troubleshooting."

Error Message `Error 2002: Can't connect to local MySQL server through socket....`

Possible Cause The MySQL database is not installed.

Recommended Action Install the database before running the tool.

Error Message `Error 1045: Access denied for user.... Operation failed.`

Possible Cause Your MySQL username and password are invalid.

Possible Cause Your MySQL username and password have insufficient permission to access the database.

Recommended Action Enter the correct or properly enabled username and password. If a system user has forgotten a password, assign a new one.

Error Message `Error 1116: Table 'sip.subscriber...' doesn't exist. Operation failed.`

Possible Cause The database whose name you entered does not exist.

Recommended Action Enter a valid name or reinstall the database.

Error Message `ERROR: Invalid user_id syntax.`

Possible Cause Your subscriber ID has invalid syntax.

Recommended Action Enter a valid subscriber ID.

Error Message `ERROR: Invalid dest_url_cfna syntax.`

Possible Cause Your call-forwarding destination URL has invalid syntax.

Recommended Action Enter a valid URL.

APPENDIX D

DNS Setup

Domain Name System (DNS) is a system used in the Internet for translating names of network nodes into addresses. This appendix describes how to set up DNS services for use with your Cisco SIP proxy server (Cisco SPS).

Starting with Cisco SPS 2.1, you should set up your DNS services with naming-authority pointer record (NAPTR) records and server (SRV) records based on RFC 3263 guidelines. Doing so ensures high scalability, availability, security, and interoperability of your service deployments.

Different levels of DNS records are used as shown in Figure D-1.

Figure D-1 DNS Record Levels

```
|
+-service-1 [SRV] gives an ordered list of targets to contact
|             |   for service-1 in the domain. We'll call this
|             |   list the "SRV records list."
|             |
|             +-server-1 [A] gives a list of network addresses for the target.
|             |           |  We'll call this list the "A records list."
|             |           |
|             |           +-IP-1 1st network address to contact for service-1
|             |           |      in domain
|             |           +-IP-2 2nd network address to contact for service-1
|             |           |      in domain
|             |           :
|             |           :
|             |
|             +-server-2
|             |
|             :
|             :
|
+---service-2
|
:
:...service-N
:
```

Use DNS records for the following purposes:

- Naming-authority pointer (NAPTR) records—Use to set up different services in the domain. RFC 3263 defines each transport support on Session Initialization Protocol (SIP) as a different service. Hence, SIP over Transmission Control Protocol (TCP), SIP over User Datagram Protocol (UDP), and SIP over Transport Layer Security Protocol (TLS) are three different services, with three different NAPTR records.

- Server resource (SRV) records—Use to provide contacts for the specific domain services. An SRV lookup for a specific service results in an ordered list of SRV records. You can therefore assign your preferred contacts for the service the highest priority and your backups a lower priority.

  Let each SRV record correspond to an individual farm (not farm member). This helps in the smart-failover mechanism, as suggested in RFC 3263. There should be a unique and single fully qualified domain name (FQDN) for each farm, which should be returned in the query results.

- A records—Use to provide IP addresses for specific contacts or individual farm members. Multiple A records can point to the IP address of each proxy server. For example, the FQDN for a farm in SRV records can point to a list of IP addresses, each of which also points to individual host names of each farm member.

Sample Configuration: NAPTR Records

Following is an example of a DNS configuration for domain cisco.com for a setup that has two Cisco SPS farms. The primary farm is farm1.cisco.com; the backup farm is farm2.cisco.com.

```
;;;; NAPTR records for sip services
   ;          order pref flags service      regexp  replacement
      IN NAPTR 50   50   "s"   "SIPS+D2T"     ""   _sips._tcp.cisco.com.
      IN NAPTR 90   50   "s"   "SIP+D2T"      ""   _sip._tcp.cisco.com.
      IN NAPTR 100  50   "s"   "SIP+D2U"      ""   _sip._udp.cisco.com.

;;;; SRV records for each sip service
  ;;                           Priority Weight Port Target
  _sips._tcp.cisco.com SRV      10       1      5061 farm1.cisco.com.
                       SRV      20       1      5061 farm2.cisco.com.
  _sip._tcp.cisco.com  SRV      10       1      5060 farm1.cisco.com.
                       SRV      20       1      5060 farm2.cisco.com.
  _sip._udp.cisco.com  SRV      10       1      5060 farm1.cisco.com.
                       SRV      20       1      5060 farm2.cisco.com.

;;;; A records for the contacts mentioned in SRV records
    farm1          IN  A            10.4.175.126  ;; proxy1
                   IN  A            10.4.175.127  ;; proxy2
    farm2          IN  A            10.4.175.128  ;; proxy3
                   IN  A            10.4.175.129  ;; proxy4

;;;; A records for the well known names of the hosts
    proxy1         IN  A            10.4.175.126
    proxy2         IN  A            10.4.175.127
    proxy3         IN  A            10.4.175.128
    proxy4         IN  A            10.4.175.129
```

In this example, the proxy-server domain is cisco.com, and therefore the proxy-server configuration directive ProxyDomain is also set to cisco.com. Similarly, the proxy-server farm FQDNs are farm1.cisco.com and farm2.cisco.com, and therefore the directive ServerName is also set to one of the two values, as appropriate.

You can use Cisco SPS to support multiple virtual domains. In such deployments, the virtual domain owners might configure their NAPTR DNS records such that the domain suffix in the NAPTR replacement field points to a DNS SRV entry in the actual server domain instead of their own domain. The domain suffixes in the NAPTR replacement field need not match the domain of the original query. However, for backward compatibility with RFC 2543, such domains must maintain SRV records for the domain of the original query, even if the NAPTR record points to a different domain.

As an example, if the SIP+D2T service field above contained the TCP SRV replacement value _sip._tcp.example.com, an SRV record must also exist at the domain cisco.com. The SRV query string _sip._tcp.cisco.com should return the contact as the actual FQDN in the example.com domain. Since RFC 2543 clients may not support NAPTR lookup, they look up the SRV records for the domain

cisco.com directly. Clients whose queries are not answered fail. Again, to maintain maximum compatibility with upstream stateless proxies, we recommend that you assign different weights to SRV records with equal priority.

For NAPTR records with SIPS protocol fields, if Cisco SPS uses a site certificate, the domain name of the NAPTR query and the domain name in the replacement field must both be valid according to the site certificate handed out by Cisco SPS in the TLS exchange. Similarly, the domain name in the SRV query and the domain name in the target in the SRV record must both be valid according to the same site certificate. This ensures correct trust credentials with upstream clients.

APPENDIX E

SIP Call-Flow Scenarios

This appendix describes the types of Session Initiation Protocol (SIP) messages used by the Cisco SIP proxy server (Cisco SPS) and the flow of these messages during various call scenarios.

Contents

Note For more troubleshooting information, see Chapter 5, "Troubleshooting."

SIP Messages

Message Types

All SIP messages are either requests from a server or client or responses to a request (see Table E-1).

Table E-1 SIP Message Types

Type	Message	Action or Indication
Request	INVITE	Invites a user or service to participate in a call session
	ACK	Confirms that the client has received a final response to an INVITE request
	BYE	Terminates a call and can be sent by either the caller or the called party
	CANCEL	Cancels any pending searches but does not terminate a call that has already been accepted
	OPTIONS	Queries the capabilities of servers
	REGISTER	Registers the address listed in the To header field with a SIP server

Table E-1 SIP Message Types (continued)

Type	Message	Action or Indication
Response	SIP 1*xx*	Informational
	SIP 2*xx*	Successful
	SIP 3*xx*	Redirection
	SIP 4*xx*	Client failure
	SIP 5*xx*	Server failure
	SIP 6xx	Global failure

Messages are formatted according to RFC 822, *Standard for the Format of ARPA Internet Text Messages*. The general format for all messages is as follows:

- A start line
- One or more header fields
- An empty line
- An optional message body
- An ending carriage return-line feed (CRLF)

SIP URLs

The SIP URL in a message identifies the address of a user and takes a form similar to an e-mail address:

```
user@host
```

where *user* is the telephone number and *host* is either a domain name or a numeric network address.

For example, the Request-URI field in an INVITE request to a user appears as follows:

```
INVITE sip:555-0002@company.com; user=phone
```

The user=phone parameter indicates that the Request-URI address is a telephone number rather than a username.

Registration and Invitation Processes

SIP messages facilitate two types of process: registration and invitation.

Registration occurs when a client informs a proxy or redirect server of its location. The client sends a REGISTER request to the proxy or redirect server and includes the addresses at which it can be reached.

Invitation occurs when one SIP endpoint (user A) invites another SIP endpoint (user B) to join in a call. This process occurs as follows:

1. User A sends an INVITE message requesting that user B join a particular conference or establish a two-party conversation.
2. If user B wants to join the call, it sends an affirmative response (SIP 2xx). If not, it sends a failure response (SIP 4xx).
3. If user A still wants to establish the conference, it acknowledges the response with an ACK message. If not, it sends a BYE message.

Call-Flow Scenarios for Successful Calls

This section describes call flows for the following successful-call scenarios:

Note The messages shown are examples for reference only.

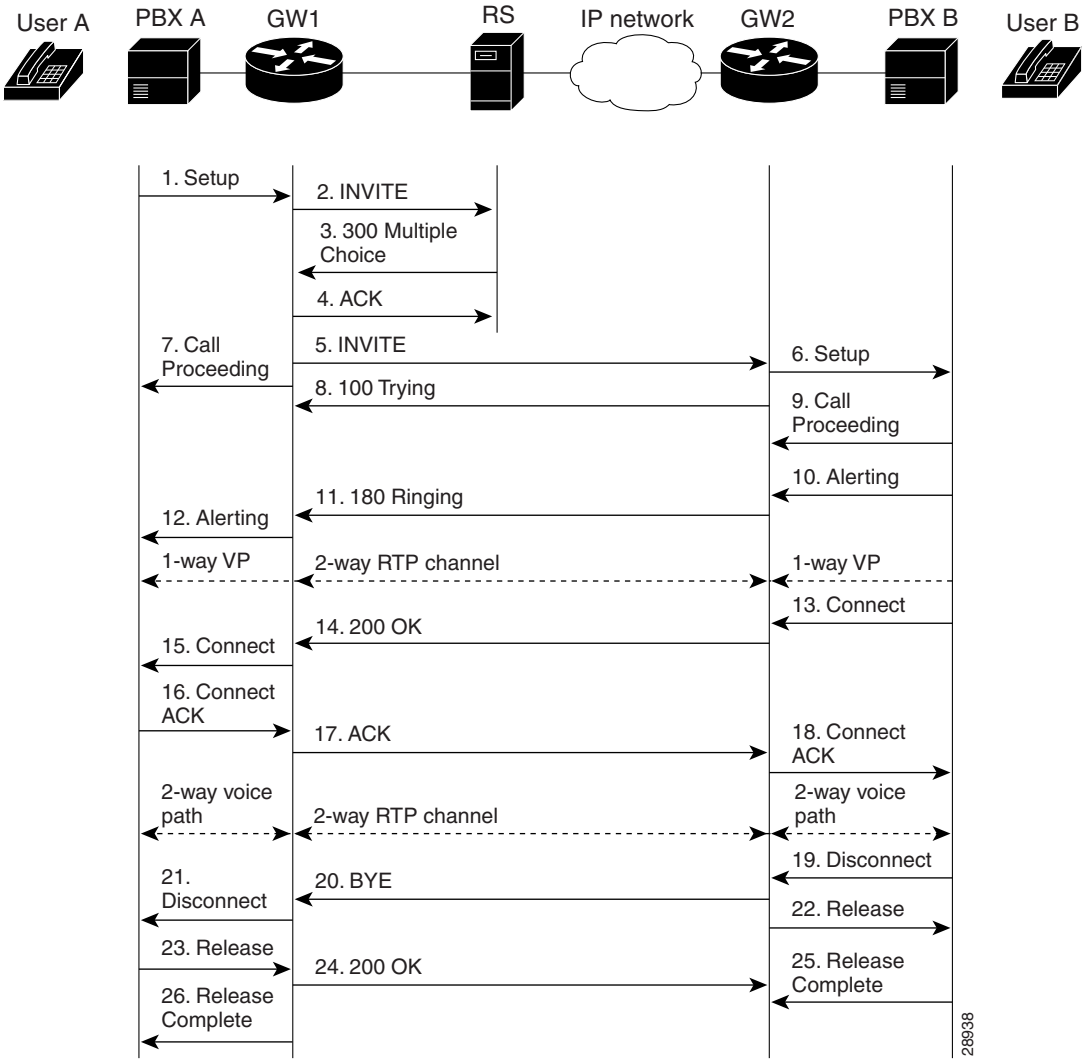
SIP Gateway to SIP Gateway—via SIP Redirect Server

Figure E-1 illustrates a successful gateway-to-gateway call setup and disconnect via a SIP redirect server. In this scenario, the two end users are identified as user A and user B. User A is located at PBX A. PBX A is connected to SIP gateway 1 via a T1/E1. SIP gateway 1 is using a SIP redirect server. User B is located at PBX B. PBX B is connected to SIP gateway 2 via a T1/E1. User B's phone number is 555-0002. SIP gateway 1 is connected to SIP gateway 2 over an IP network.

The call flow scenario is as follows:

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

Figure E-1 SIP Gateway to SIP Gateway—via SIP Redirect Server



	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP redirect server	SIP gateway 1 sends an INVITE request to the SIP redirect server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.

	Action	Description
Step 3	300 Multiple Choice—SIP redirect server to SIP gateway 1	The SIP redirect server sends a 300 Multiple Choice response to SIP gateway 1. The response indicates that the SIP redirect server accepted the INVITE request, contacted a location server with all or part of user B's SIP URL, and the location server provided a list of alternative locations where user B might be located. The SIP redirect server returns these possible addresses to SIP gateway 1 in the 300 Multiple Choice response.
Step 4	ACK—SIP gateway 1 to SIP redirect server	SIP gateway 1 acknowledges the 300 Multiple Choice response with an ACK.
Step 5	INVITE—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a new INVITE request to SIP gateway 2. The new INVITE request includes the first contact listed in the 300 Multiple Choice response as the new address for user B, a higher transaction number in the CSeq field, and the same Call-ID as the first INVITE request.
Step 6	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from SIP gateway 1 and initiates a call setup with user B via PBX B.
Step 7	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 8	100 Trying—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The response indicates that the INVITE request was received by SIP gateway 2, but that user B is not yet located.
Step 9	Call Proceeding—PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the call-setup request.
Step 10	Alerting—PBX B to SIP gateway 2	PBX B locates user B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.
Step 11	180 Ringing—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 180 Ringing response to SIP gateway 1. The response indicates that SIP gateway 2 has located, and is trying to alert user B.
Step 12	Alerting—SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to PBX A. User A hears ringback tone.
Note	At this point, a one-way voice path is established between SIP gateway 1 and PBXA and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 13	Connect—PBX B to SIP gateway 2	User B answers phone. PBX B sends a Connect message to SIP gateway 2. The message notifies SIP gateway 2 that the connection has been made.
Step 14	200 OK—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 200 OK response to SIP gateway 1. The response notifies SIP gateway 1 that the connection has been made. If user B supports the media capability advertised in the INVITE message sent by SIP gateway 1, it advertises the intersection of its own and user A's media capability in the 200 OK response. If user B does not support the media capability advertised by user A, it returns a 400 Bad Request response with a 304 Warning header field.
Step 15	Connect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The message notifies PBX A that the connection has been made.
Step 16	Connect ACK—PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
Step 17	ACK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that the 200 OK response has been received. The call is now in progress over a two-way voice path via RTP.

	Action	Description
Step 18	Connect ACK—SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message.
Note	At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 19	Disconnect—PBX B to SIP gateway 2	Once user B hangs up, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
Step 20	BYE—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a BYE request to SIP gateway 1. The request indicates that user B wants to release the call. Because it is user B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains user B's SIP URL.
Step 21	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 22	Release—SIP gateway 2 to PBX B	SIP gateway 2 sends a Release message to PBX B.
Step 23	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 24	200 OK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
Step 25	Release Complete—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
Step 26	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the session is terminated.

SIP Gateway to SIP Gateway—via SIP Proxy Server

Figure E-2 and Figure E-3 illustrate a successful gateway-to-gateway call setup and disconnect via a proxy server. In these scenarios, the two end users are user A and user B. User A is located at PBX A. PBX A is connected to SIP gateway 1 via a T1/E1. SIP gateway 1 is using a proxy server. SIP gateway 1 is connected to SIP gateway 2 over an IP network. User B is located at PBX B. PBX B is connected to SIP gateway 2 (a SIP gateway) via a T1/E1. User B's phone number is 555-0002.

In the scenario illustrated by Figure E-2, the record route feature is enabled on the proxy server. In the scenario illustrated by Figure E-3, record route is disabled on the proxy server.

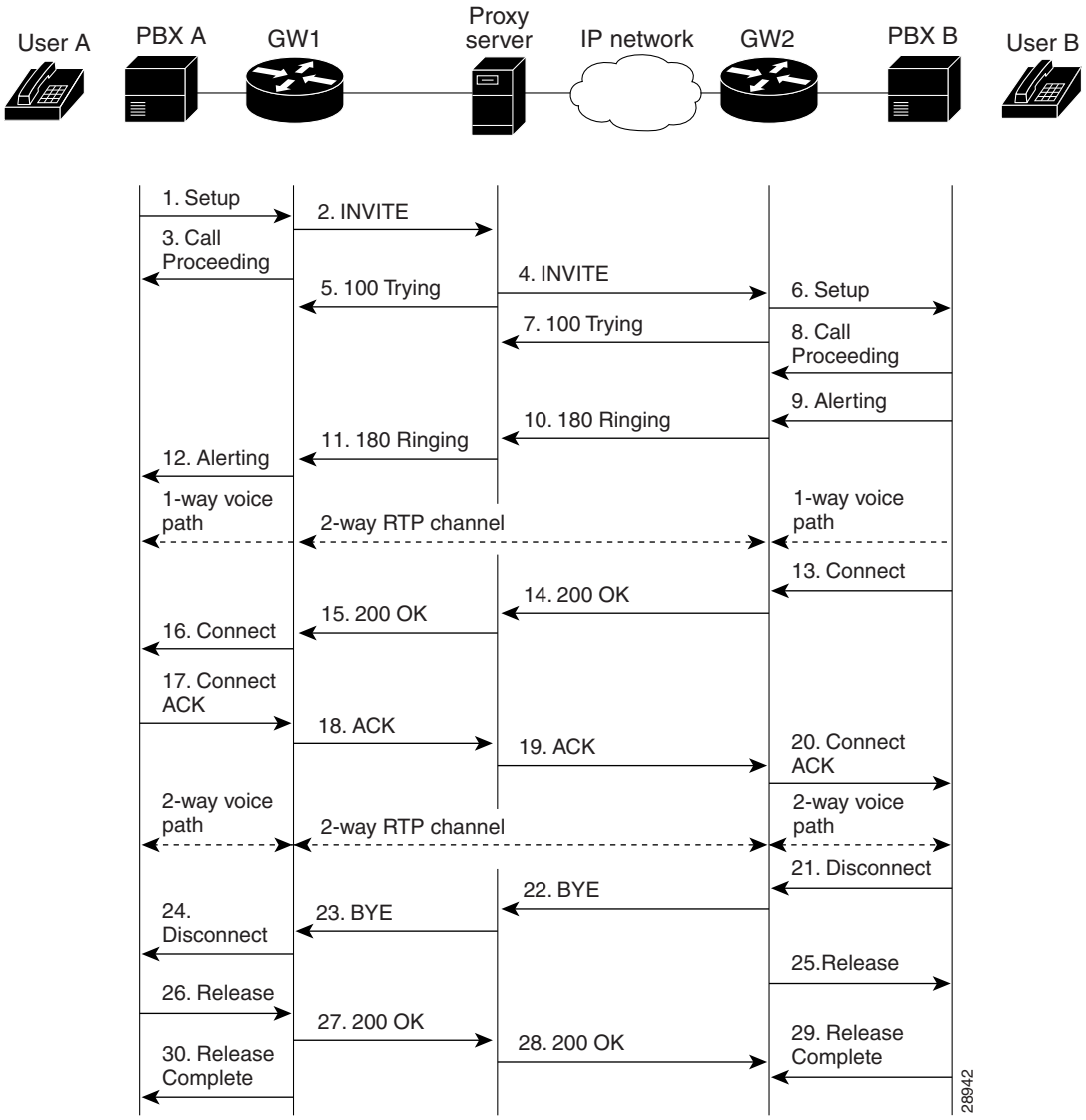
When record route is enabled, the proxy server adds the Record-Route header to the SIP messages to ensure that it is in the path of subsequent SIP requests for the same call leg. The Record-Route field contains a globally reachable Request-URI that identifies the proxy server. When record route is enabled,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

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

The call flow is as follows:

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

Figure E-2 SIP Gateway to SIP Gateway—via SIP Proxy Server: Record Route Enabled

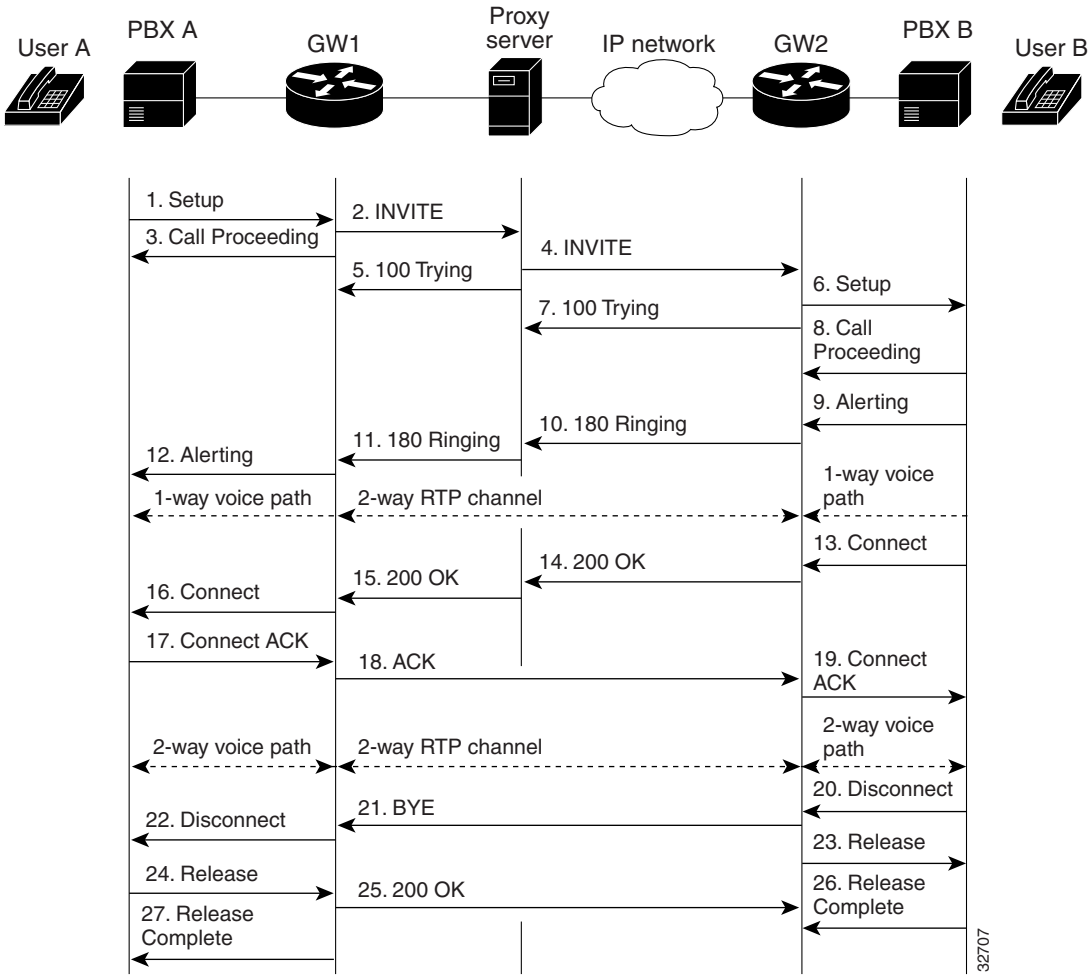


	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 4	INVITE—SIP proxy server to SIP gateway 2	SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and sends a new INVITE request to SIP gateway 2.
Step 5	100 Trying—SIP proxy server to SIP gateway 1	SIP proxy server sends a 100 Trying response to SIP gateway 1.
Step 6	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with user B via PBX B.
Step 7	100 Trying—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server. SIP proxy server might or might not forward the 100 Trying response to SIP gateway 1.
Step 8	Call Proceeding—PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the call-setup request.
Step 9	Alerting—PBX B to SIP gateway 2	PBX B locates user B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.
Step 10	180 Ringing—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 180 Ringing response to the SIP proxy server.
Step 11	180 Ringing—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 180 Ringing response to SIP gateway 1.
Step 12	Alerting—SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to user A via PBX A. The message indicates that SIP gateway 1 has received a 180 Ringing response. User A hears the ringback tone that indicates that user B is being alerted.
Note	At this point, a one-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 13	Connect—PBX B to SIP gateway 2	User B answers the phone. PBX B sends a Connect message to SIP gateway 2. The message notifies SIP gateway 2 that the connection has been made.

	Action	Description
Step 14	200 OK—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 200 OK response (including the Record-Route header received in the INVITE) to the SIP proxy server. The response notifies the SIP proxy server that the connection has been made. If user B supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and user A's media capability in the 200 OK response. If user B does not support the media capability advertised by user A, it returns a 400 Bad Request response with a 304 Warning header field. SIP proxy server must forward 200 OK responses upstream.
Step 15	200 OK—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 200 OK response that it received from SIP gateway 2 to SIP gateway 1. In the 200 OK response, the SIP proxy server forwards the Record-Route header to ensure that it is in the path of subsequent SIP requests for the same call leg. In the Record-Route field, the SIP proxy server adds its Request-URI.
Step 16	Connect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The message notifies PBX A that the connection has been made.
Step 17	Connect ACK—PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
Step 18	ACK—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an ACK to the SIP proxy server. The ACK confirms that SIP gateway 1 has received the 200 OK response from the SIP proxy server.
Step 19	ACK—SIP proxy server to SIP gateway 2	Depending on the values in the To, From, CSeq, and Call-ID field, the SIP proxy server might process the ACK locally or proxy it. If the fields in the ACK match those in previous requests processed by the SIP proxy server, the server proxies the ACK. If there is no match, the ACK is proxied as if it were an INVITE request. SIP proxy server forwards SIP gateway 1's ACK response to SIP gateway 2.
Step 20	Connect ACK—SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. The call session is now active. The two-way voice path is established directly between SIP gateway 1 and SIP gateway 2; not via the SIP proxy server.
Note	At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 21	Disconnect—PBX B to SIP gateway 2	After the call is completed, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
Step 22	BYE—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a BYE request to the SIP proxy server. The request indicates that user B wants to release the call. Because it is user B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains user B's SIP URL.
Step 23	BYE—SIP proxy server to SIP gateway 1	SIP proxy server forwards the BYE request to SIP gateway 1.
Step 24	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 25	Release—SIP gateway 2 to PBX B	After the call is completed, SIP gateway 2 sends a Release message to PBX B.

	Action	Description
Step 26	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 27	200 OK—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends a 200 OK response to the SIP proxy server. The response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
Step 28	200 OK—SIP proxy server to SIP gateway 2	SIP proxy server forwards the 200 OK response to SIP gateway 2.
Step 29	Release Complete—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
Step 30	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session is terminated.

Figure E-3 SIP Gateway to SIP Gateway—via a Proxy Server: Record Route Disabled



	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability that A is ready to receive is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 4	INVITE—SIP proxy server to SIP gateway 2	SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and sends a new INVITE request to SIP gateway 2.
Step 5	100 Trying—SIP proxy server to SIP gateway 1	SIP proxy server sends a 100 Trying response to SIP gateway 1.
Step 6	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with user B via PBX B.
Step 7	100 Trying—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server. SIP proxy server might or might not forward the 100 Trying response to SIP gateway 1.
Step 8	Call Proceeding—PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the call-setup request.
Step 9	Alerting—PBX B to SIP gateway 2	PBX B locates user B and sends an Alert message to SIP gateway 2. User B's phone begins to ring.
Step 10	180 Ringing—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 180 Ringing response to the SIP proxy server.
Step 11	180 Ringing—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 180 Ringing response to SIP gateway 1.
Step 12	Alerting—SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to user A via PBX A. The message indicates that SIP gateway 1 has received a 180 Ringing response. User A hears the ringback tone that indicates that user B is being alerted.
Note	At this point, a one-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 13	Connect—PBX B to SIP gateway 2	User B answers the phone. PBX B sends a Connect message to SIP gateway 2. The message notifies SIP gateway 2 that the connection has been made.

	Action	Description
Step 14	200 OK—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 200 OK response to the SIP proxy server. The response notifies the SIP proxy server that the connection has been made. If user B supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and user A's media capability in the 200 OK response. If user B does not support the media capability advertised by user A, it returns a 400 Bad Request response with a 304 Warning header field. SIP proxy server must forward 200 OK responses upstream.
Step 15	200 OK—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 200 OK response that it received from SIP gateway 2 to SIP gateway 1.
Step 16	Connect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Connect message to PBX A. The message notifies PBX A that the connection has been made.
Step 17	Connect ACK—PBX A to SIP gateway 1	PBX A acknowledges SIP gateway 1's Connect message.
Step 18	ACK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that SIP gateway 1 has received the 200 OK response from the SIP proxy server.
Step 19	Connect ACK—SIP gateway 2 to PBX B	SIP gateway 2 acknowledges PBX B's Connect message. The call session is now active. The two-way voice path is established directly between SIP gateway 1 and SIP gateway 2, and not via the SIP proxy server.
Note	At this point, a two-way voice path is established between SIP gateway 1 and PBX A and between SIP gateway 2 and PBX B. A two-way RTP channel is established between SIP gateway 1 and SIP gateway 2.	
Step 20	Disconnect—PBX B to SIP gateway 2	After the call is completed, PBX B sends a Disconnect message to SIP gateway 2. The Disconnect message starts the call session termination process.
Step 21	BYE—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a BYE request to SIP gateway 1. The request indicates that user B wants to release the call. Because it is user B that wants to terminate the call, the Request-URI field is now replaced with PBX A's SIP URL and the From field contains user B's SIP URL.
Step 22	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 23	Release—SIP gateway 2 to PBX B	After the call is completed, SIP gateway 2 sends a Release message to PBX B.
Step 24	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 25	200 OK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The response notifies SIP gateway 2 that SIP gateway 1 has received the BYE request.
Step 26	Release Complete—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.
Step 27	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session is terminated.

SIP IP Phone to SIP IP Phone—Call Forward Unconditionally

Figure E-4 and Figure E-5 illustrate a successful SIP IP phone-to-SIP IP phone call forward unconditionally via a SIP proxy. In these scenarios, the three end users and endpoints are identified as Alice at SIP IP phone A, Bob at SIP IP phone B, and Carol at SIP IP phone C. Bob's calls are configured to forward to Carol unconditionally. Figure E-4 illustrates the call as processed by a recursive proxy and Figure E-5 illustrates the call as processed by a nonrecursive proxy.

Figure E-4 SIP IP Phone to SIP IP Phone—Call Forward Unconditionally via Recursive Proxy

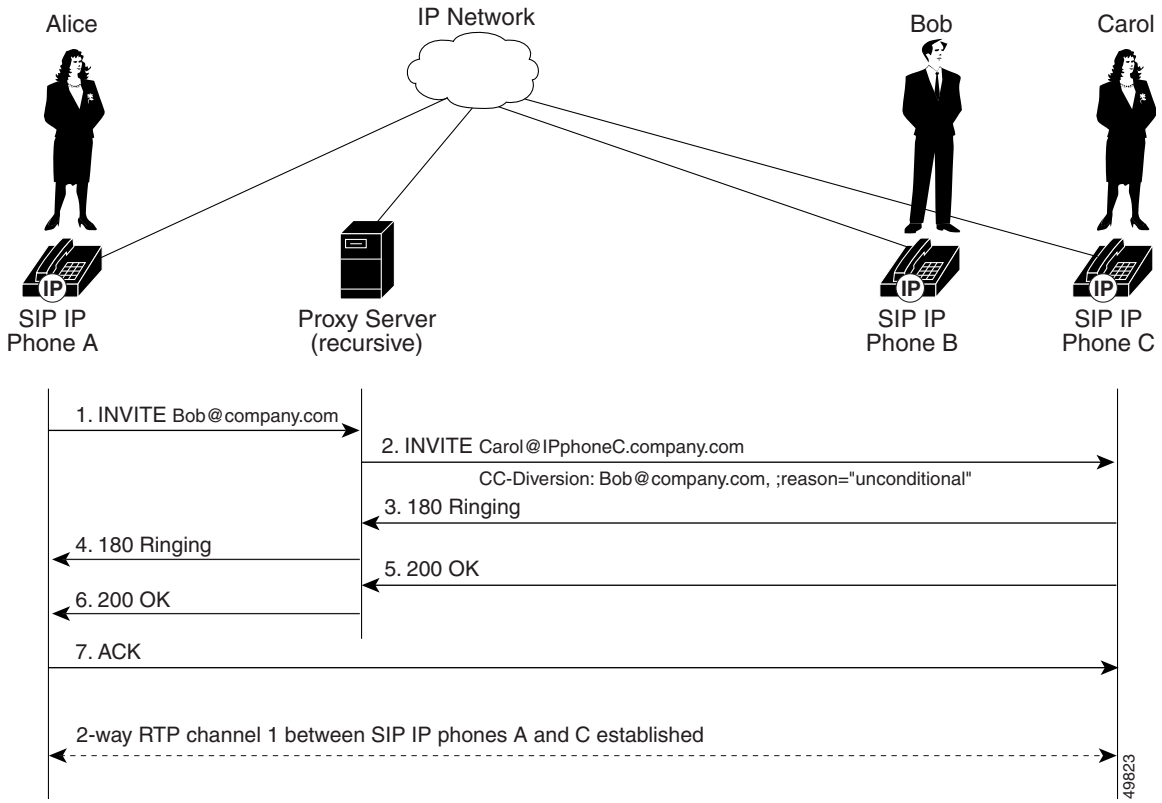
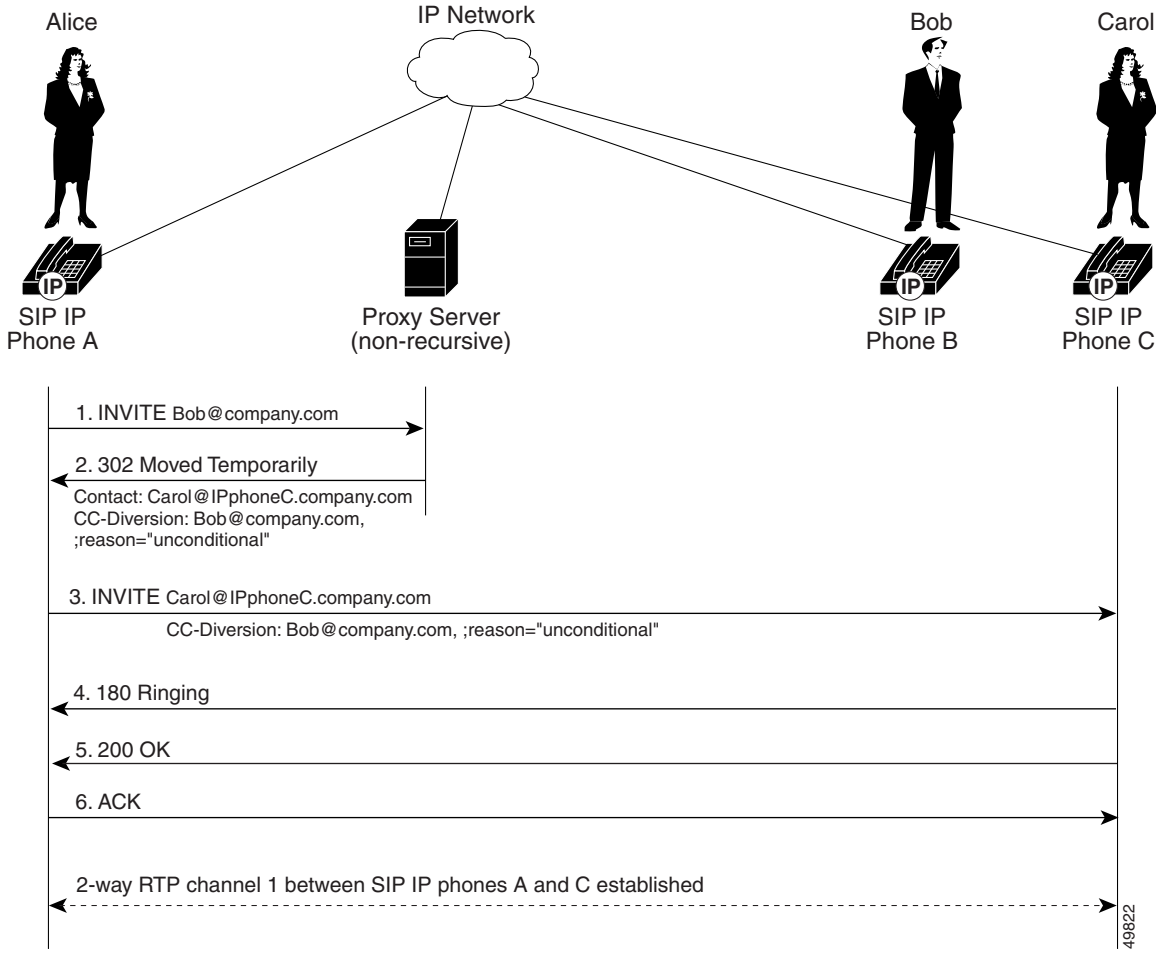


Figure E-5 SIP IP Phone to SIP IP Phone—Call Forward Unconditionally via Nonrecursive Proxy



	Action	Description
Step 1	INVITE—SIP IP phone A to SIP proxy server	Alice's phone A sends an INVITE request to the proxy server. The request is an invitation to Bob to participate in a call session. The following applies: • Bob's phone number is inserted in the Request-URI field in the form of a SIP URL. • Alice at phone A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of phone A is specified in the SDP. • The port on which phone A is prepared to receive RTP data is specified in the SDP.
Step 2	302 Moved Temporarily—SIP proxy server to SIP IP phone A	SIP proxy server determines that Bob's calls have been configured to forward unconditionally to Carol at phone C. It sends an 302 Moved Temporarily message to phone A. Carol at phone C is added as the Contact and a CC-Diversion header is added that contains the Request-URI from the initial INVITE and the reason for the diversion.

	Action	Description
Step 3	INVITE—SIP IP phone A to SIP IP phone C	Phone A sends an INVITE request to Carol at phone C. The request contains a CC-Diversion header that contains the Request-URI from the initial INVITE request and the reason for the diversion.
Step 4	180 Ringing—SIP IP phone C to SIP proxy server	Phone C sends a 180 Ringing response to phone A.
Step 5	200 OK—SIP IP phone C to SIP IP phone A	Phone C sends a 200 OK response to phone A. The response notifies phone A that Carol has answered the phone (for example, the handset went off-hook). If phone C supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and phone A's media capability in the 200 OK response. If phone C does not support the media capability advertised by phone A, it returns a 400 Bad Request response with a "Warning: 304 Codec negotiation failed" header field.
Step 6	ACK—SIP IP phone A to SIP IP phone C	Phone A sends an ACK to phone C. The ACK confirms that phone A has received the 200 OK response from phone C.
Note	At this point, a two-way RTP channel is established between SIP IP phone A and SIP IP phone C.	

SIP IP Phone to SIP IP Phone—Call Forward on Busy

Figure E-6 and Figure E-7 illustrate a successful SIP IP phone-to-SIP IP phone call forward on busy via a SIP proxy. In these scenarios, the three end users are identified as user A, user B, and user C. User B's calls are configured to forward to user C when user B's SIP IP phone sends a 486 Busy Here response. Figure E-6 illustrates the call as processed by a recursive proxy and Figure E-7 illustrates the call as processed by a nonrecursive proxy.

Figure E-6 SIP IP Phone to SIP IP Phone—Call Forward on Busy via Recursive Proxy

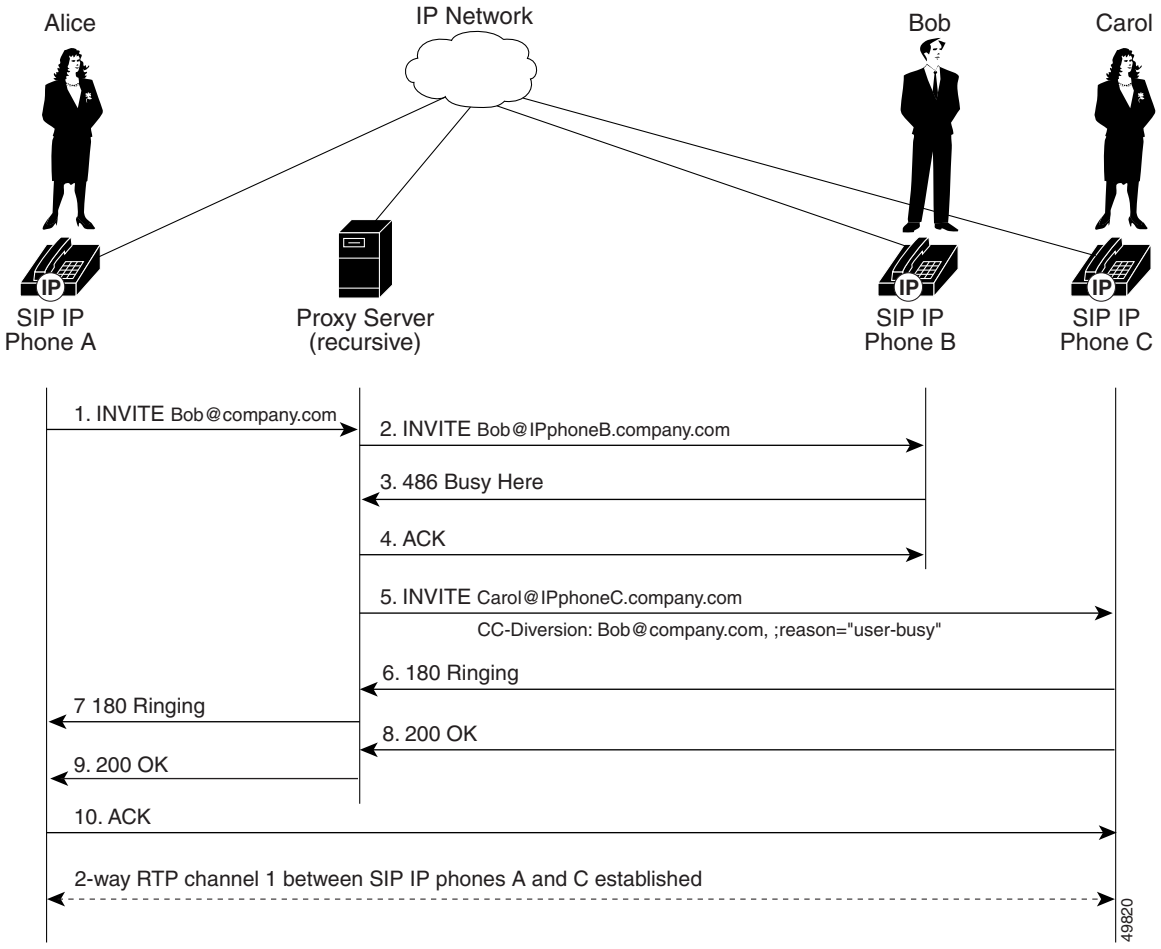
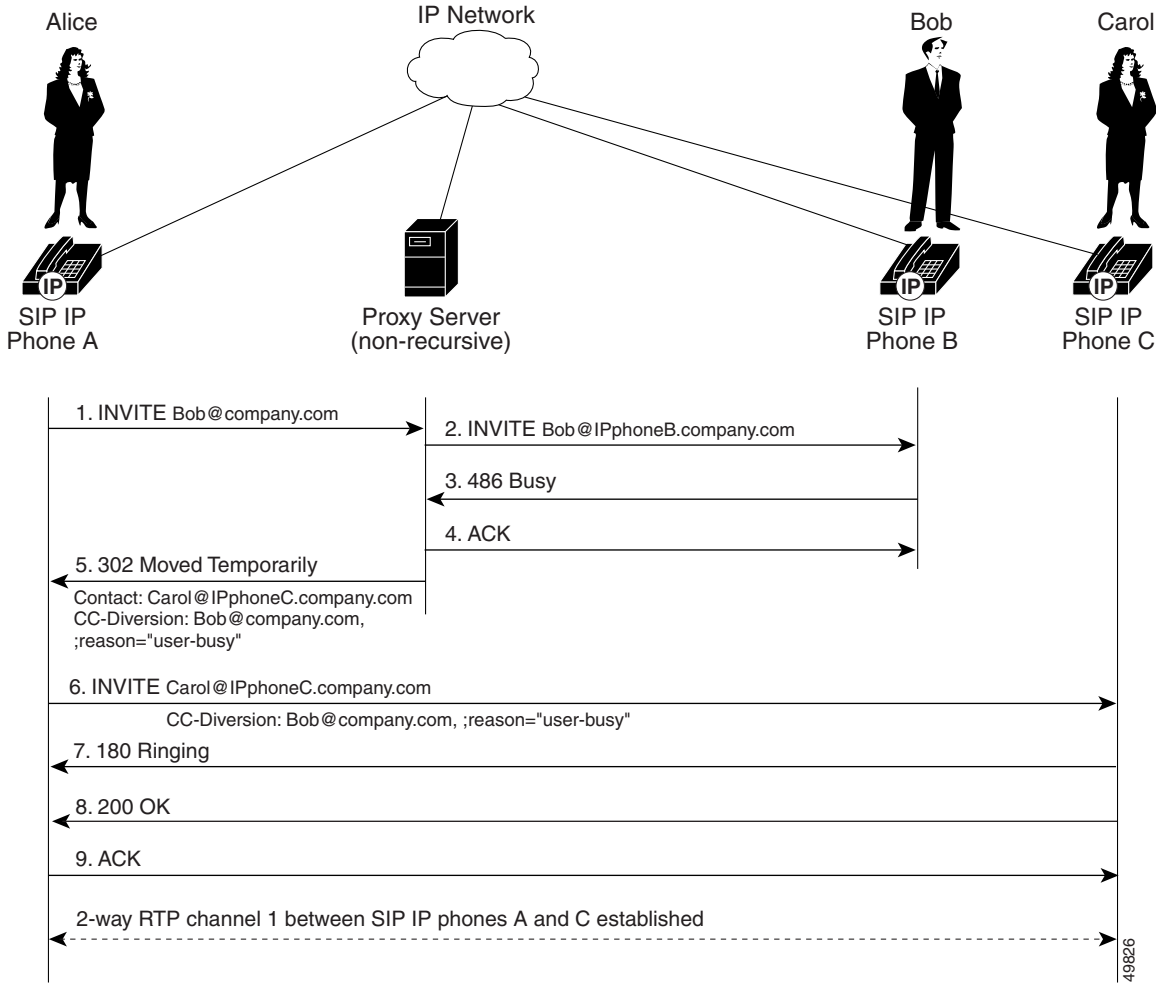


Figure E-7 SIP IP Phone to SIP IP Phone—Call Forward on Busy via Nonrecursive Proxy



	Action	Description
Step 1	INVITE—SIP IP phone A to SIP proxy server	Alice's phone A sends an INVITE request to the proxy server. The request is an invitation to Bob to participate in a call session. The following applies: • Bob's phone number is inserted in the Request-URI field in the form of a SIP URL. • Alice at phone A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of phone A is specified in the SDP. • The port on which phone A is prepared to receive RTP data is specified in the SDP.
Step 2	INVITE—SIP proxy server to SIP IP phone B	SIP proxy server forwards the INVITE request to Bob at phone B.

	Action	Description
Step 3	486 Busy Here—SIP IP phone B to the SIP proxy server	Phone B sends a 486 Busy response to the SIP proxy server. The response indicates that Bob at phone B was successfully contacted but was either unwilling or unable to take another call.
Step 4	302 Moved Temporarily—SIP proxy server to SIP IP phone A	SIP proxy server sends an 302 Moved Temporarily message to phone A. Carol at phone C is added as the Contact and a CC-Diversion header is added that contains the Request-URI from the initial INVITE and the reason for the diversion.
Step 5	INVITE—SIP proxy server to SIP IP phone C	SIP proxy server sends an INVITE request to Carol at phone C to which Bob's calls have been configured to forward on busy, changes the Request-URI to divert the request to Carol at phone C, and adds a CC-Diversion header containing the Request-URI from the initial INVITE request and the reason for the diversion.
Step 6	180 Ringing—SIP IP phone C to SIP IP phone A	Phone C sends a 180 Ringing response to phone A.
Step 7	200 OK—SIP IP phone C to SIP IP phone A	Phone C sends a 200 OK response to phone A. The response notifies phone A that Carol has answered the phone (for example, the handset went off-hook). If phone C supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and phone A's media capability in the response. If phone C does not support the media capability advertised by phone A, it returns a 400 Bad Request response with a "Warning: 304 Codec negotiation failed" header field.
Step 8	ACK—SIP IP phone A to SIP IP phone C	Phone A sends an ACK to phone C. The ACK confirms that phone A has received the 200 OK response from phone C.
Note	At this point, a two-way RTP channel is established between SIP IP phone A and SIP IP phone C.	

SIP IP Phone to SIP IP Phone—Call Forward No Answer

Figure E-8 and Figure E-9 illustrate a successful SIP IP phone-to-SIP IP phone call forward when no answer via a SIP proxy. In these scenarios, the three end users are identified as user A, user B, and user C. User B's calls are configured to forward to user C when an response timeout occurs. Figure E-8 illustrates the call as processed by a recursive proxy and Figure E-9 illustrates the call as processed by a nonrecursive proxy.

Figure E-8 SIP IP Phone to SIP IP Phone—Call Forward No Answer via Recursive Proxy

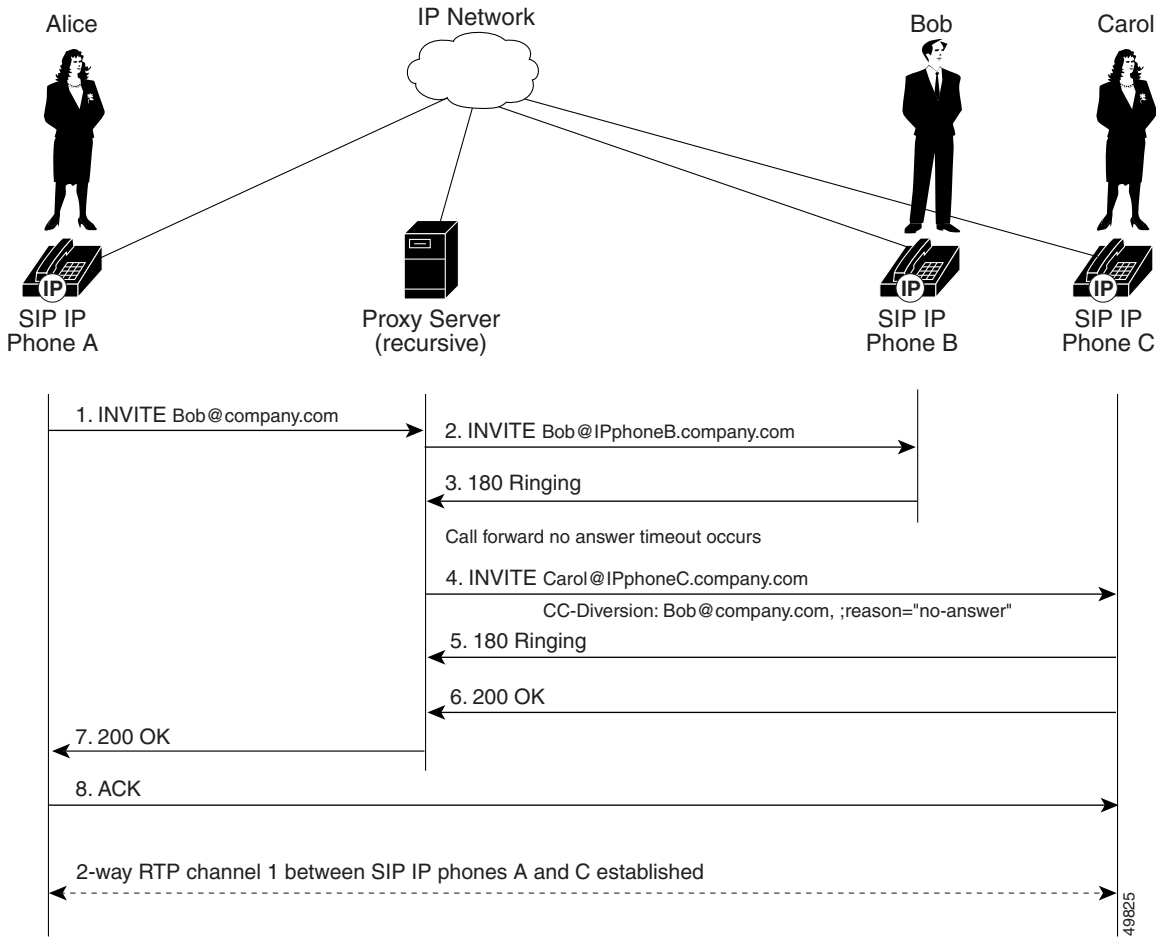
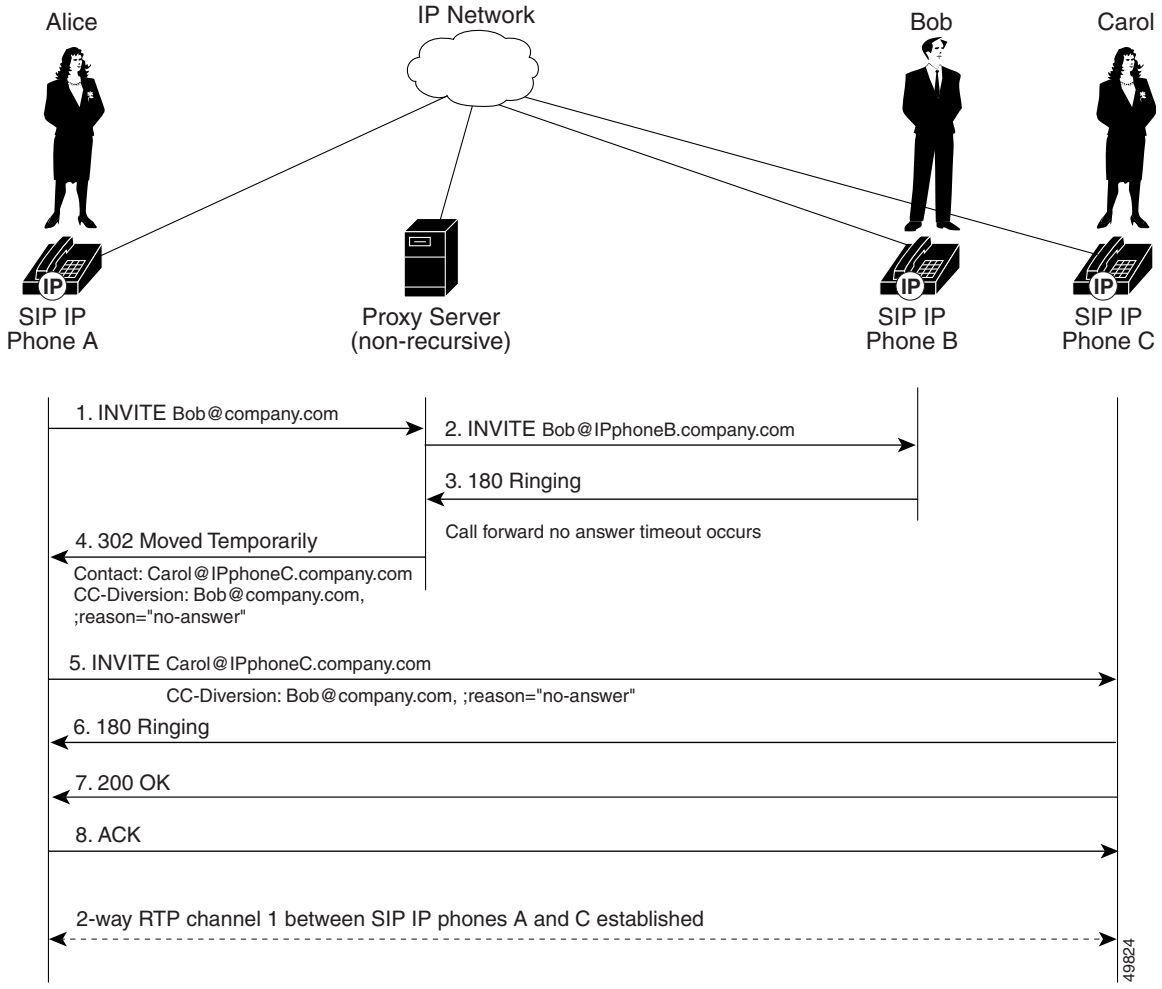


Figure E-9 SIP IP Phone to SIP IP Phone—Call Forward No Answer via Nonrecursive Proxy



	Action	Description
Step 1	INVITE—SIP IP phone A to SIP proxy server	Alice's phone A sends an INVITE request to the proxy server. The request is an invitation to Bob to participate in a call session. The following applies: • Bob's phone number is inserted in the Request-URI field in the form of a SIP URL. • Alice at phone A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of phone A is specified in the SDP. • The port on which phone A is prepared to receive RTP data is specified in the SDP.
Step 2	INVITE—SIP proxy server to SIP IP phone B	SIP proxy server forwards the INVITE request to Bob at phone B.

	Action	Description
Step 3	180 Ringing—SIP IP phone B to the SIP proxy server	Phone B sends a 180 Ringing response to the SIP proxy server.
Note	Timeout to INVITE request occurs.	
Step 4	302 Moved Temporarily—SIP proxy server to SIP IP phone A	SIP proxy server sends an 302 Moved Temporarily message to phone A. Carol at phone C is added as the Contact and a CC-Diversion header is added that contains the Request-URI from the initial INVITE and the reason for the diversion.
Step 5	INVITE—SIP IP phone A to SIP IP phone C	Phone A sends an INVITE request to Carol at phone C to which Bob's calls have been configured to forward when Bob is unavailable, changes the Request-URI to divert the request to Carol at phone C, and adds a CC-Diversion header containing the Request-URI from the initial INVITE request and the reason for the diversion.
Step 6	180 Ringing—SIP IP phone C to SIP IP phone A	Phone C sends a 180 Ringing response to phone A.
Step 7	200 OK—SIP IP phone C to SIP IP phone A	Phone C sends a 200 OK response to phone A. The response notifies phone A that Carol has answered the phone (for example, the handset went off-hook). If phone C supports the media capability advertised in the INVITE message sent by the SIP proxy server, it advertises the intersection of its own and phone A's media capability in the 200 OK response. If phone C does not support the media capability advertised by phone A, it returns a 400 Bad Request response with a "Warning: 304 Codec negotiation failed" header field.
Step 8	ACK—SIP IP phone A to SIP IP phone C	Phone A sends an ACK to phone C. The ACK confirms that phone A has received the 200 OK response from phone C.
Note	At this point, a two-way RTP channel is established between SIP IP phone A and SIP IP phone C.	

SIP IP Phone to SIP IP Phone—Call Forward Unavailable

Figure E-10 and Figure E-11 illustrate a successful SIP IP phone-to-SIP IP phone call forward when the callee is unavailable via a SIP proxy. In these scenarios, the three end users are identified as user A, user B, and user C. User B's calls are configured to forward to user C when user B is unavailable. Figure E-10 illustrates the call as processed by a recursive proxy and Figure E-11 illustrates the call as processed by a nonrecursive proxy.

Figure E-10 SIP IP Phone to SIP IP Phone—Call Forward Unavailable via Recursive Proxy

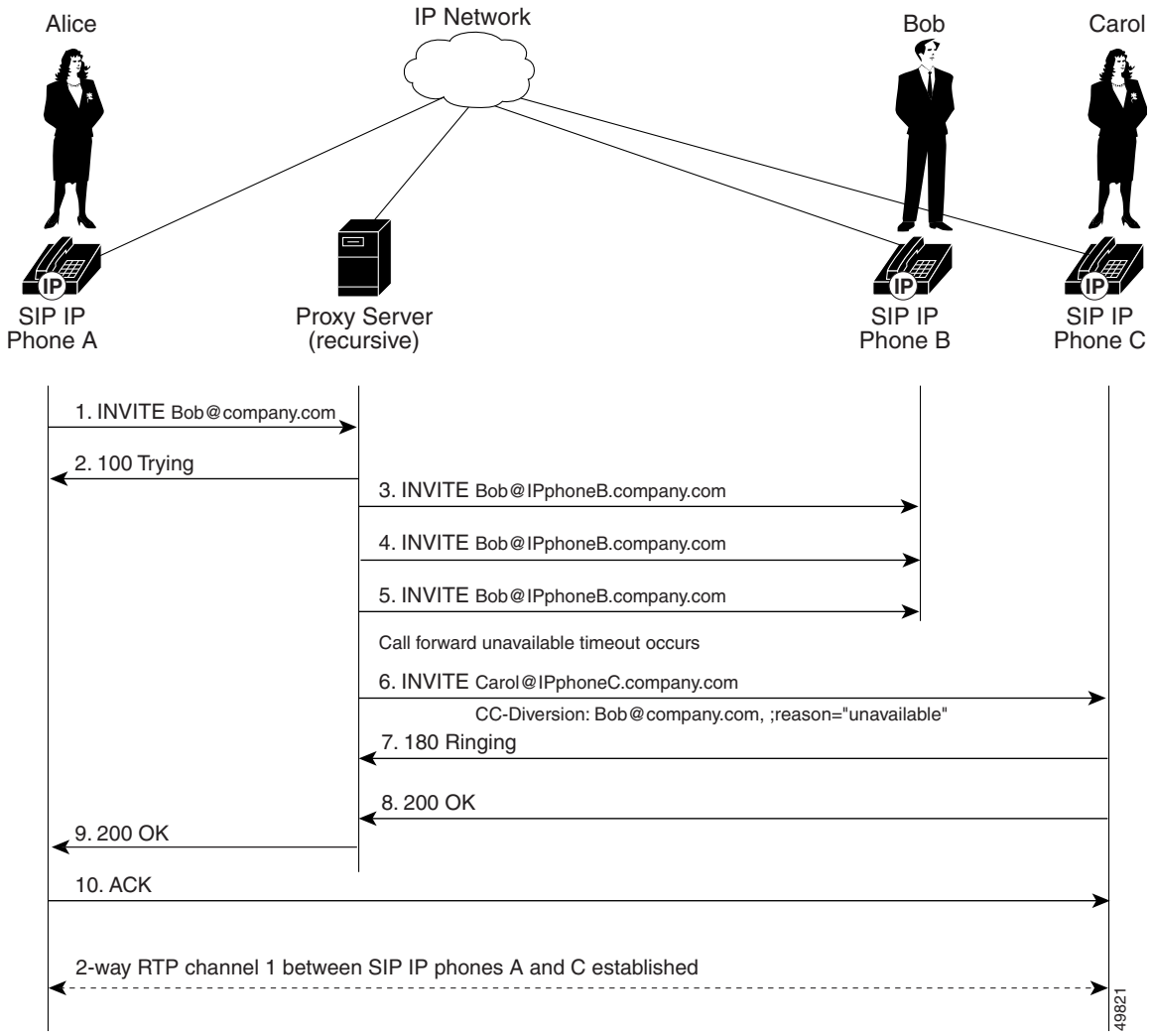
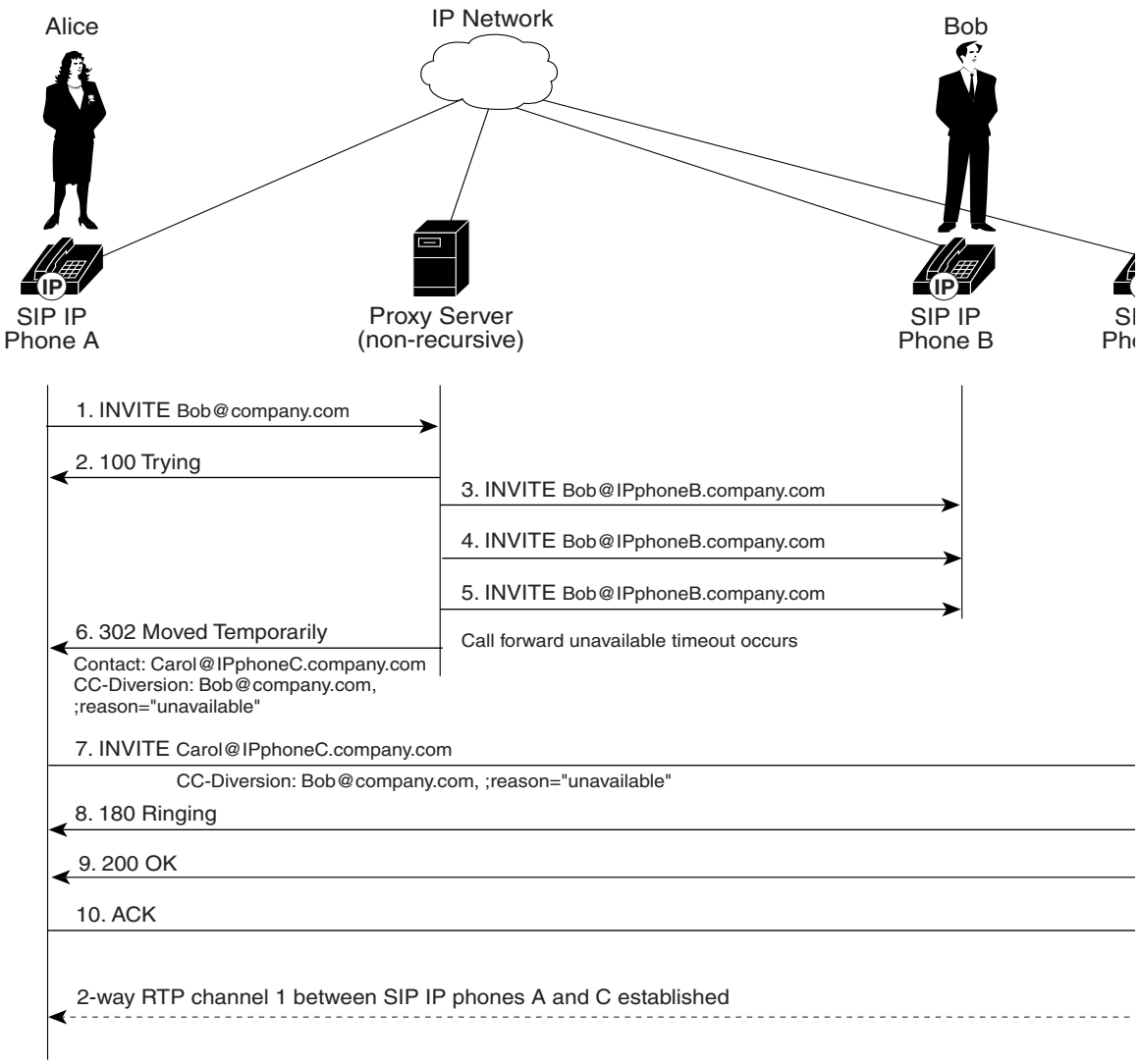


Figure E-11 SIP IP Phone to SIP IP Phone—Call Forward Unavailable via Nonrecursive Proxy



Call-Flow Scenarios for Failed Calls

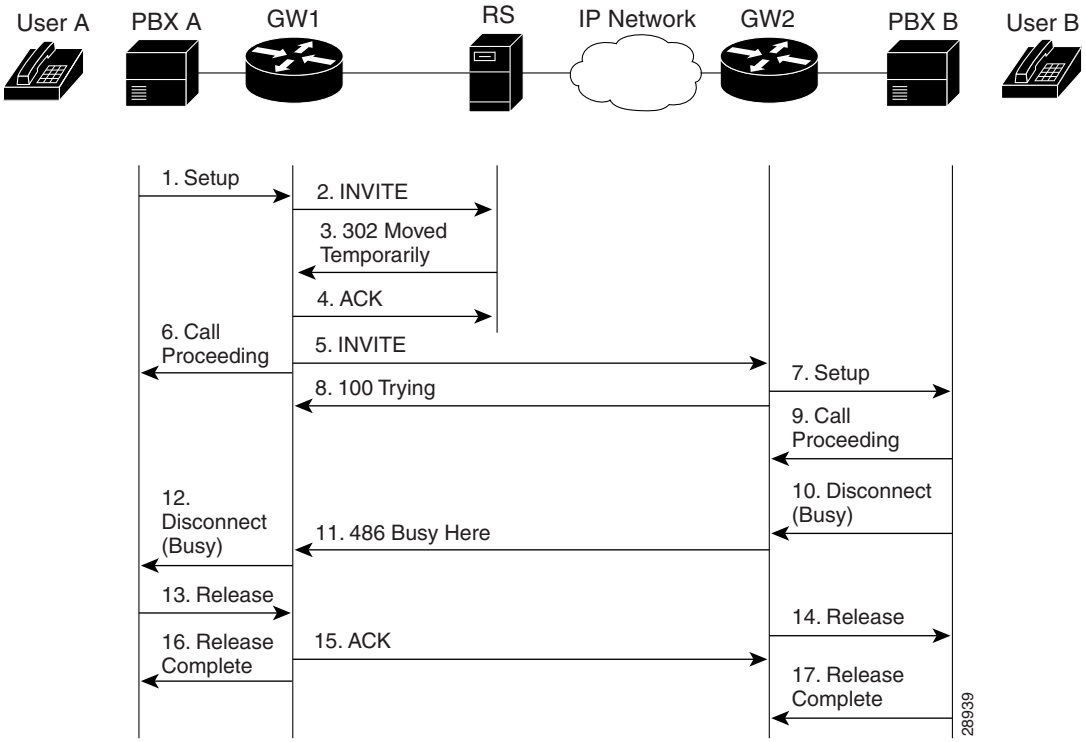
This section describes call flows for the following failed-call scenarios:

Note The messages are provided as examples for reference only.

SIP Gateway to SIP Gateway via SIP Redirect Server—Called User Is Busy

Figure E-12 illustrates an unsuccessful call in which user A initiates a call to user B but user B is on the phone and is unable or unwilling to accept another call.

Figure E-12 SIP Gateway to SIP Gateway via SIP Redirect Server—Called User Is Busy



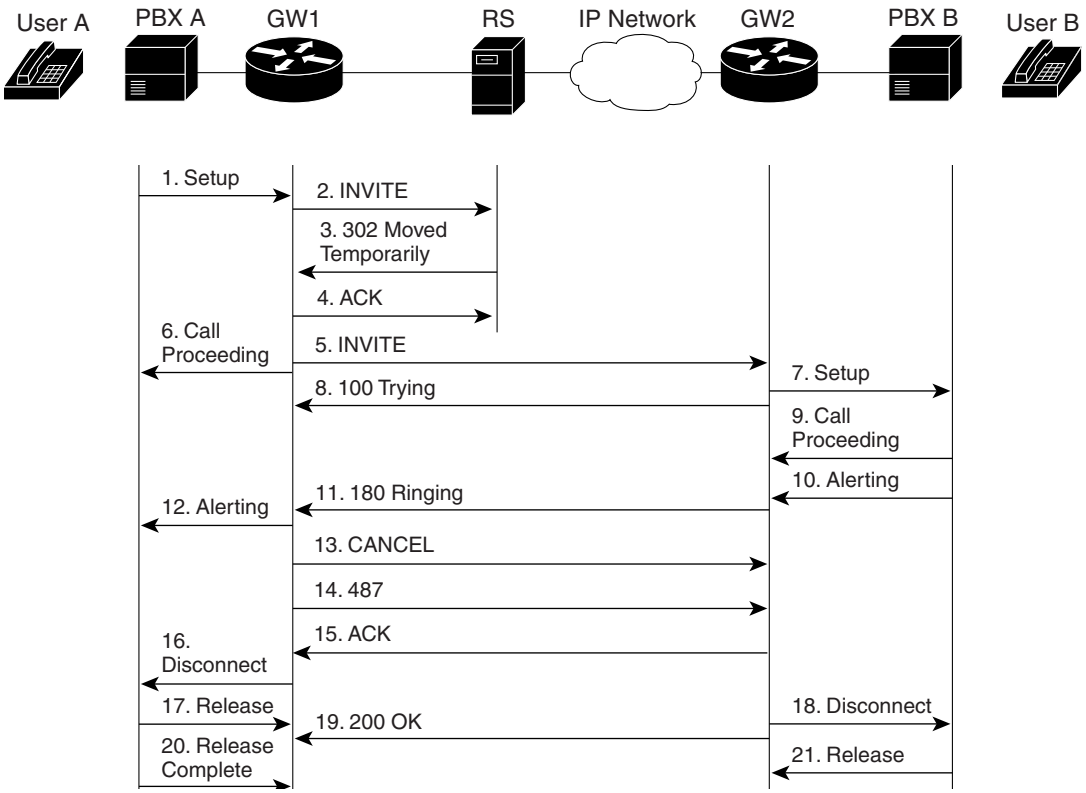
	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP redirect server	SIP gateway 1 sends an INVITE request to the SIP redirect server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	302 Moved Temporarily— SIP redirect server to SIP gateway 1	SIP redirect server sends a 302 Moved Temporarily message to SIP gateway 1. The message indicates that user B is not available and includes instructions to locate user B.
Step 4	ACK—SIP gateway 1 to SIP redirect server	SIP gateway 1 acknowledges the 302 Moved Temporarily response with an ACK.
Step 5	INVITE—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a new INVITE request to user B. The new INVITE request includes the first contact listed in the 300 Multiple Choice response as the new address for user B, a higher transaction number in the CSeq field, and the same Call-ID as the first INVITE request.
Step 6	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 7	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from SIP gateway 1 and initiates call setup with user B via PBX B.
Step 8	100 Trying—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The response indicates that the INVITE request has been received by SIP gateway 2 but that user B has not yet been located and that some unspecified action, such as a database consultation, is taking place.
Step 9	Call Proceeding—PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the call-setup request.
Step 10	Disconnect (Busy)—PBX B to SIP gateway 2	PBX B sends a Disconnect message to SIP gateway 2. The cause code indicates that user B is busy. The Disconnect message starts the call session termination process.
Step 11	486 Busy Here—SIP gateway 2 to SIP gateway 1	SIP gateway 2 maps the Release message cause code (Busy) to the 486 Busy response and sends the response to SIP gateway 1. The response indicates that user B's phone was successfully contacted but user B was either unwilling or unable to take another call.
Step 12	Disconnect (Busy) —SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A. User A hears a busy tone.
Step 13	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.

	Action	Description
Step 14	Release—SIP gateway 2 to PBX B	SIP gateway 1 sends a Release message to PBX B.
Step 15	ACK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that the 486 Busy Here response has been received.
Step 16	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session attempt is terminated.
Step 17	Release Complete—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2.

SIP Gateway to SIP Gateway via SIP Redirect Server—Called User Does Not Answer

Figure E-13 illustrates an unsuccessful call in which user A initiates a call to user B but user B does not answer.

Figure E-13 SIP Gateway to SIP Gateway via SIP Redirect Server—Called User Does Not Answer



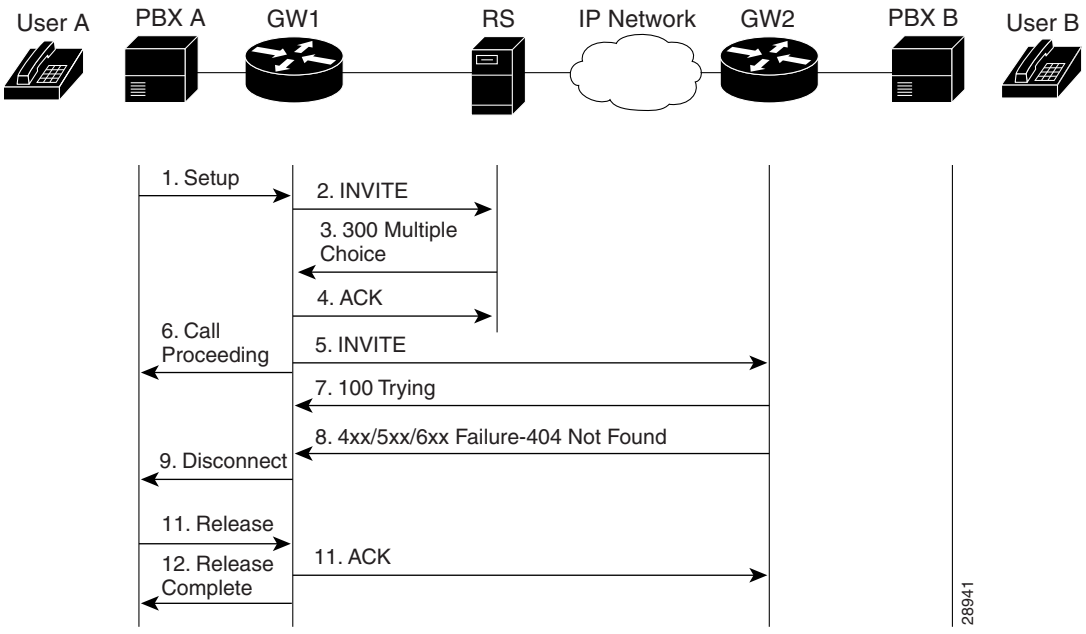
	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP redirect server	SIP gateway 1 sends an INVITE request to the SIP redirect server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	302 Moved Temporarily— SIP redirect server to SIP gateway 1	SIP redirect server sends a 302 Moved Temporarily message to SIP gateway 1. The message indicates that user B is not available and includes instructions to locate user B.
Step 4	ACK—SIP gateway 1 to SIP redirect server	SIP gateway 1 acknowledges the 302 Moved Temporarily response with an ACK.
Step 5	INVITE—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a new INVITE request to user B. The new INVITE request includes a new address for user B, a higher transaction number in the CSeq field, but the same Call-ID as the first INVITE request.
Step 6	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 7	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from SIP gateway 1 and initiates call setup with user B via PBX B.
Step 8	100 Trying—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The message indicates that the INVITE request has been received by SIP gateway 2 but that user B has not yet been located and that some unspecified action, such as a database consultation, is taking place.
Step 9	Call Proceeding—PBX B to SIP gateway 2	PBX B sends a Call Proceeding message to SIP gateway 2 to acknowledge the call-setup request.
Step 10	Alerting—PBX B to SIP gateway 2	PBX B sends an Alert message to SIP gateway 2. User B's phone begins to ring.
Step 11	180 Ringing—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 180 Ringing response to SIP gateway 1. The response indicates that SIP gateway 2 has located, and is trying to alert user B.
Step 12	Alerting—SIP gateway 1 to PBX A	SIP gateway 1 sends an Alert message to PBX A.
Step 13	CANCEL (Ring Timeout)—SIP gateway 1 to SIP gateway 2	Because SIP gateway 2 did not return an appropriate response within the time specified by the Expires header in the INVITE request, SIP gateway 1 sends a SIP CANCEL request to SIP gateway 2. A CANCEL request cancels a pending request with the same Call-ID, To, From, and CSeq header field values.
Step 14	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.

	Action	Description
Step 15	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 16	Disconnect—SIP gateway 2 to PBX B	SIP gateway 2 sends a Disconnect message to PBX B.
Step 17	200 OK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a 200 OK response to SIP gateway 2. The 200 OK response confirms that the CANCEL request has been received.
Step 18	Release Complete—PBX A to SIP gateway 1	PBX A sends a Release Complete message to SIP gateway 1 and the call session attempt is terminated.
Step 19	Release—PBX B to SIP gateway 2	PBX B sends a Release message to SIP gateway 2.
Step 20	Release Complete—SIP gateway 2 to PBX B	SIP gateway 2 sends a Release Complete message to PBX B.

SIP Gateway to SIP Gateway via SIP Redirect Server—Client, Server, or Global Error

Figure E-14 illustrates an unsuccessful call in which user A initiates a call to user B but SIP gateway 2 determines that user B does not exist at the domain specified in the INVITE request sent by SIP gateway 1. SIP gateway 2 refuses the connection.

Figure E-14 SIP Gateway to SIP Gateway via SIP Redirect Server—Client, Server, or Global Error



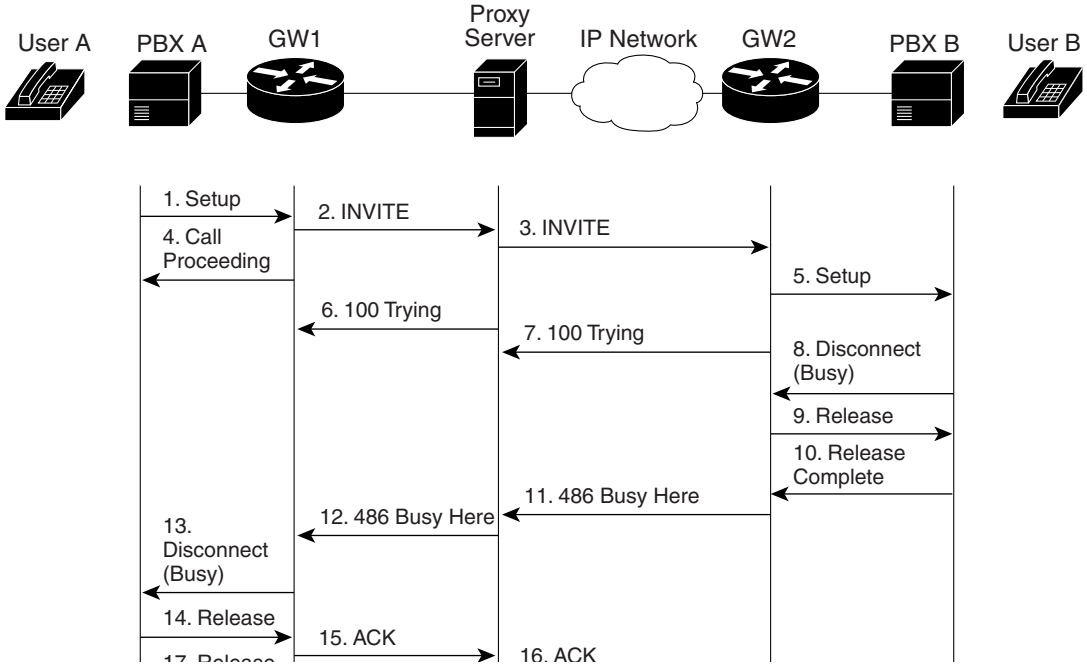
	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP redirect server	SIP gateway 1 sends an INVITE request to the SIP redirect server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	300 Multiple Choice—SIP redirect server to SIP gateway 1	The SIP redirect server sends a 300 Multiple Choice response to SIP gateway 1. The response indicates that the SIP redirect server accepted the INVITE request, contacted a location server with all or part of user B's SIP URL, and the location server provided a list of alternative locations where user B might be located. The SIP redirect server returns these possible addresses to user A in the 300 Multiple Choice response.
Step 4	ACK—SIP gateway 1 to SIP redirect server	SIP gateway 1 acknowledges the 300 Multiple Choice response with an ACK.
Step 5	INVITE—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a new INVITE request to user B. The new INVITE request includes a new address for user B, a higher transaction number in the CSeq field, but the same Call-ID as the first INVITE request.
Step 6	Call Proceeding—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 7	100 Trying—SIP gateway 2 to SIP gateway 1	SIP gateway 2 sends a 100 Trying response to the INVITE request sent by SIP gateway 1. The message indicates that the INVITE request has been received by SIP gateway 2 but that user B has not yet been located and that some unspecified action, such as a database consultation, is taking place.
Step 8	Class 4xx/5xx/6xx Failure—SIP gateway 2 to SIP gateway 1	SIP gateway 2 determines that user B does not exist at the domain specified in the INVITE request sent by SIP gateway 1. SIP gateway 2 refuses the connection and sends a 404 Not Found response to SIP gateway 1. The 404 Not Found response is a class 4*xx* failure response. The call actions differ, based on the class of failure response. If SIP gateway 2 sends a class 4*xx* failure response (a definite failure response that is a client error), the request is not retried without modification. If SIP gateway 2 sends a class 5*xx* failure response (an indefinite failure that is a server error), the request is not terminated but rather other possible locations are tried. If SIP gateway 2 sends a class 6*xx* failure response (a global error), the search for user B terminates because the response indicates that a server has definite information about user B, but not for the particular instance indicated in the Request-URI field. Therefore, all further searches for this user fail.

	Action	Description
Step 9	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 10	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 11	ACK—SIP gateway 1 to SIP gateway 2	SIP gateway 1 sends an ACK to SIP gateway 2. The ACK confirms that the 404 Not Found failure response has been received.
Step 12	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session attempt is terminated.

SIP Gateway to SIP Gateway via SIP Proxy Server—Called User Is Busy

Figure E-15 illustrates an unsuccessful call in which user A initiates a call to user B but user B is on the phone and is unwilling or unable to accept another call.

Figure E-15 SIP Gateway to SIP Gateway via SIP Proxy Server—Called User Is Busy



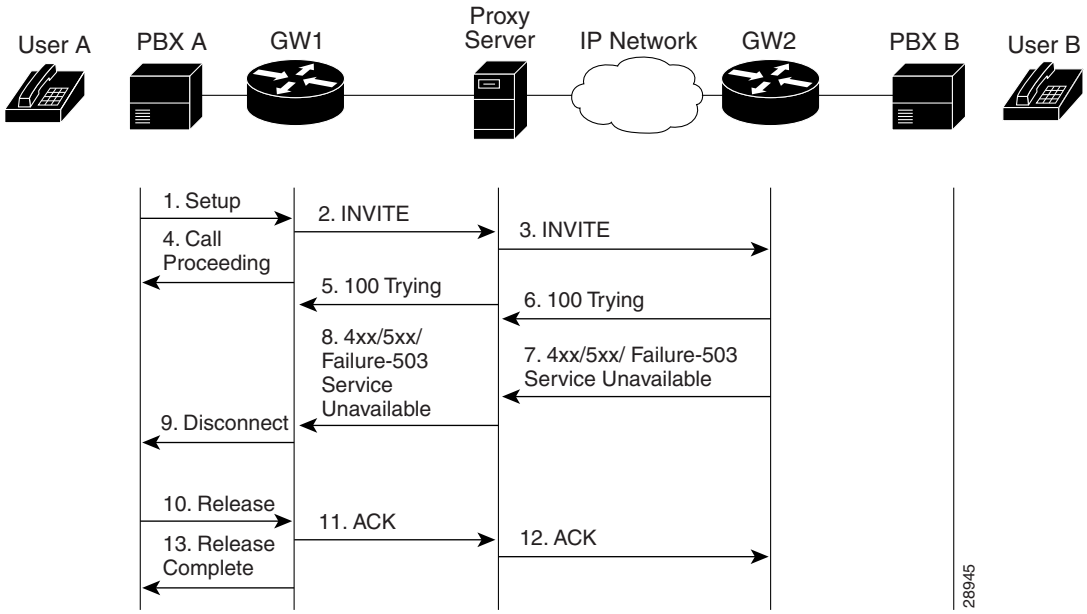
	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	INVITE—SIP proxy server to SIP gateway 2	SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and sends a new INVITE request to SIP gateway 2.
Step 4	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 5	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with user B via PBX B.
Step 6	100 Trying—SIP proxy server to SIP gateway 1	SIP proxy server sends a 100 Trying response to SIP gateway 1.
Step 7	100 Trying—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server.
Step 8	Release Complete (Busy)—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2. The cause code indicates that user B is busy. The Release Complete message starts the call session termination process.
Step 9	486 Busy Here—SIP gateway 2 to SIP proxy server	SIP gateway 2 maps the Release message cause code (Busy) to the 486 Busy response and sends the response to the SIP proxy server. The response indicates that user B's phone was successfully contacted but user B was either unwilling or unable to take another call.
Step 10	486 Busy Here—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 486 Busy response to SIP gateway 1.
Step 11	Disconnect (Busy)—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 12	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 13	ACK—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an SIP ACK to the SIP proxy server.

	Action	Description
Step 14	ACK—SIP proxy server to SIP gateway 2	SIP proxy server forwards the SIP ACK to SIP gateway 2. The ACK confirms that the 486 Busy Here response has been received.
Step 15	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session attempt is terminated.

SIP Gateway to SIP Gateway via SIP Proxy Server—Client or Server Error

Figure E-16 illustrates an unsuccessful call in which user A initiates a call to user B but there are no more channels available on SIP gateway 2. Therefore, SIP gateway 2 refuses the connection.

Figure E-16 SIP Gateway to SIP Gateway via SIP Proxy Server—Client or Server Error



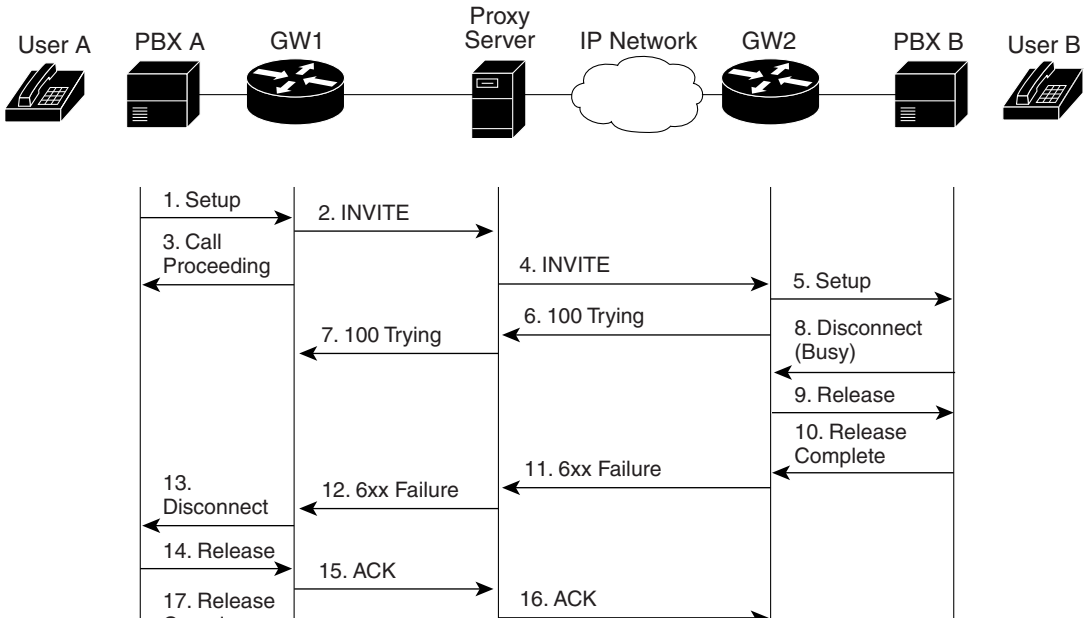
	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.

	Action	Description
Step 3	INVITE—SIP proxy server to SIP gateway 2	SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and sends a new INVITE request to SIP gateway 2.
Step 4	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 5	100 Trying—SIP proxy server to SIP gateway 1	SIP proxy server sends a 100 Trying response to SIP gateway 1.
Step 6	100 Trying—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server.
Step 7	Class 4xx/5xx/6xx Failure—SIP gateway 2 to SIP proxy server	SIP gateway 2 determines that it does not have any more channels available, refuses the connection, and sends a SIP 503 Service Unavailable response to the SIP proxy server.
Step 8	Class 4xx/5xx/6xx Failure—SIP proxy server to SIP gateway 1	SIP proxy server forwards the SIP 503 Service Unavailable response to SIP gateway 1.
Step 9	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 10	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 11	ACK—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an ACK to the SIP proxy server.
Step 12	ACK—SIP proxy server to SIP gateway 2	SIP proxy server forwards the SIP ACK to SIP gateway 2. The ACK confirms that the 503 Service Unavailable response has been received.
Step 13	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session attempt is terminated.

SIP Gateway to SIP Gateway via SIP Proxy Server—Global Error

Figure E-17 illustrates an unsuccessful call in which user A initiates a call to user B and receives a class 6*xx* response.

Figure E-17 SIP Gateway to SIP Gateway via SIP Proxy Server—Global Error

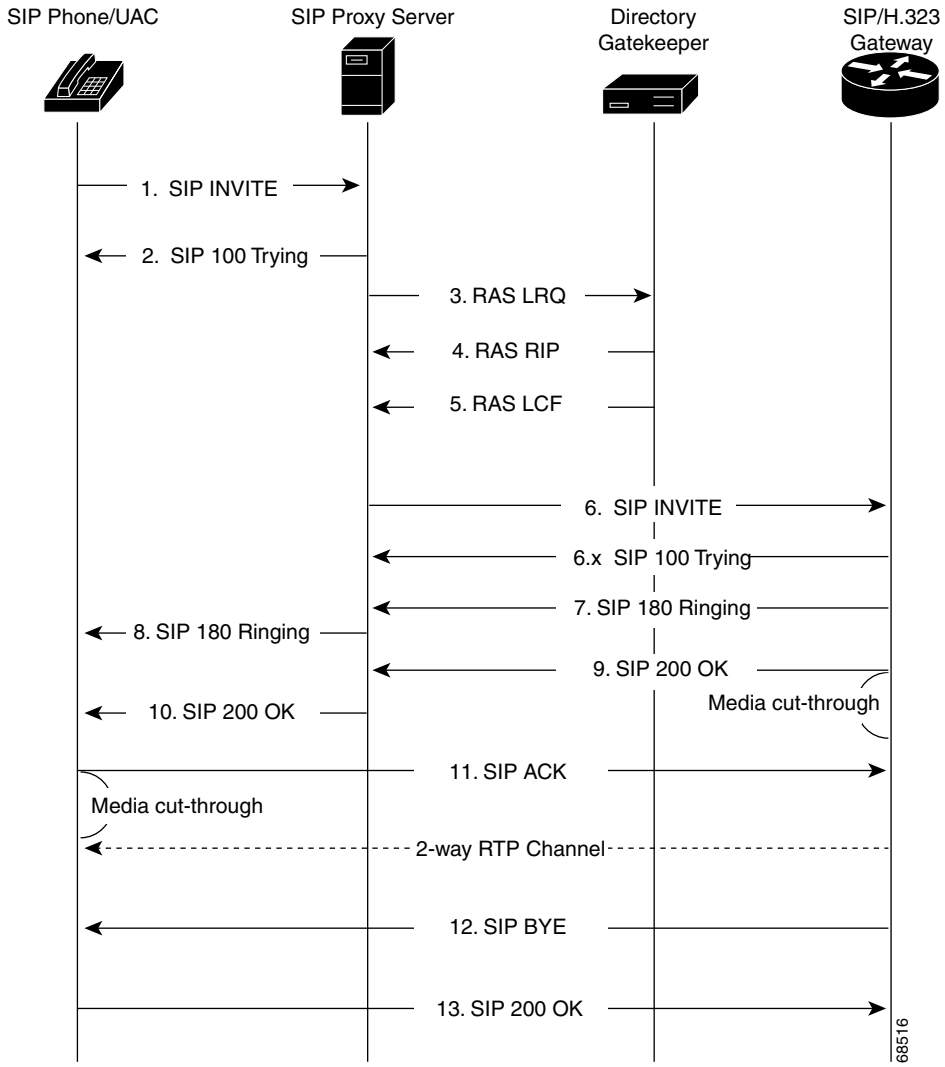


	Action	Description
Step 1	Setup—PBX A to SIP gateway 1	Call setup is initiated between PBX A and SIP gateway 1. Setup includes the standard transactions that take place as user A attempts to call user B.
Step 2	INVITE—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an INVITE request to the SIP proxy server. The request is an invitation to user B to participate in a call session. The following applies: • The phone number of user B is inserted in the Request-URI field in the form of a SIP URL. • PBX A is identified as the call-session initiator in the From field. • A unique numeric identifier is assigned to the call and inserted in the Call-ID field. • The transaction number within a single call leg is identified in the CSeq field. • The media capability of user A is specified. • The port on which SIP gateway 1 is prepared to receive RTP data is specified.
Step 3	Call Proceeding—SIP gateway 1 to PBX A	SIP gateway 1 sends a Call Proceeding message to PBX A to acknowledge the call-setup request.
Step 4	INVITE—SIP proxy server to SIP gateway 2	SIP proxy server checks whether its own address is contained in the Via field (to prevent loops), directly copies the To, From, Call-ID, and Contact fields from the request it received from SIP gateway 1, changes the Request-URI to indicate the server to which it intends to send the INVITE request, and sends a new INVITE request to SIP gateway 2.

	Action	Description
Step 5	Setup—SIP gateway 2 to PBX B	SIP gateway 2 receives the INVITE request from the SIP proxy server and initiates call setup with user B via PBX B.
Step 6	100 Trying—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a 100 Trying response to the SIP proxy server. SIP proxy server might or might not forward the 100 Trying response to SIP gateway 1.
Step 7	100 Trying—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 100 Trying response to SIP gateway 1.
Step 8	Release Complete—PBX B to SIP gateway 2	PBX B sends a Release Complete message to SIP gateway 2. The Release Complete message starts the call session termination process.
Step 9	6*xx* Failure—SIP gateway 2 to SIP proxy server	SIP gateway 2 sends a class 6*xx* failure response (a global error) to the SIP proxy server. The response indicates that a server has definite information about user B, but not for the particular instance indicated in the Request-URI field. All further searches for this user fail, therefore the search is terminated. SIP proxy server must forward all class 6*xx* failure responses to the client.
Step 10	6xx Failure—SIP proxy server to SIP gateway 1	SIP proxy server forwards the 6*xx* failure to SIP gateway 1.
Step 11	Disconnect—SIP gateway 1 to PBX A	SIP gateway 1 sends a Disconnect message to PBX A.
Step 12	Release—PBX A to SIP gateway 1	PBX A sends a Release message to SIP gateway 1.
Step 13	ACK—SIP gateway 1 to SIP proxy server	SIP gateway 1 sends an ACK to the SIP proxy server.
Step 14	ACK—SIP proxy server to SIP gateway 2	SIP proxy server sends an ACK to SIP gateway 2. The ACK confirms that the 6xx failure response has been received.
Step 15	Release Complete—SIP gateway 1 to PBX A	SIP gateway 1 sends a Release Complete message to PBX A and the call session attempt is terminated.

SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Disabled

Figure E-18 SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Disabled



	Action	Description

Step 1	INVITE—SIP phone to SIP proxy server	SIP UAC sends an INVITE request to the SIP proxy server.
		
		Example
		`INVITE sip:20002@proxy.cisco.com;user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`
		
		The phone number of called party is inserted in the Request-URI field in the form of a SIP URL. A unique numeric identifier is assigned to the call and is inserted in the Call-ID field. The transaction number within a single call leg is identified in the Cseq field. The media capability the calling party is ready to receive is specified.

	INVITE—SIP phone to SIP proxy server	SIP UAC sends an INVITE request to the SIP proxy server.
		
		Example
		`INVITE sip:20002@proxy.cisco.com;user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`

	Action	Description
Step 2	100—Trying SIP Proxy sends to UAC	SIP proxy server sends 100-Trying response message to the upstream UAC upon receiving the INVITE in step ++SIP/2.0 100 **Example** `TryingVia: SIP/2.0/UDP 161.44.3.207:49489Call-ID: 23-99990146-0-5894369F@161.44.3.207 From: "555-0001" <sip:+19195550001@bounty.cisco.com>To: <sip:20002@company.com;user=phone;phone-context=000000>CSeq: 1 INVITEContent-Length: 0`

Step 3	RAS LRQ—SIP Proxy sends a RAS LRQ message to a DGK	SIP proxy server expands the 20002 number into a 19193920002 number but finds no static route to route the request. It then invokes the new routing module and creates an LRQ RAS message from the incoming INVITE SIP message. The LRQ message is sent to one of the DGK configured in the sipd.conf file.
		
		SIP proxy server prepends a technology prefix 001# to the expanded number and uses it to fill the “destinationInfo” field of the LRQ RAS message.
		
		Example
		`value RasMessage ::= locationRequest :`
		
		`{ requestSeqNum 2519 destinationInfo { e164 : "001#19193920002" } nonStandardData { nonStandardIdentifier h221NonStandard : { t35CountryCode 181 t35Extension 0 manufacturerCode 18 } data '8284901100ECAA98A025220008000000000E1963...'H } replyAddress ipAddress : { ip 'AC12C247'H port 1719 } sourceInfo { h323-ID : {"genuity-sip1"} } canMapAlias TRUE }`

	Action	Description

Step 4	RAS RIP—H.323 DGK returns a RIP to the SIP proxy server	Upon receiving the RAS LRQ message from the SIP proxy server, the H.323 DGK can return a RIP with delay timer value. SIP server should adjust timer accordingly.
		
		Example
		`value RasMessage ::= requestInProgress :`
		`{`
		`requestSeqNum 2519`
		`delay 9000`
		`}`

Step 5	RAS LCF—H.323 DGK returns a LCF to the SIP proxy server	`{`
		`requestSeqNum 2519`
		`callSignalAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 1720`
		`}`
		`rasAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 56812`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '00024009006300330036003200300002D0032002D...'H`
		`}`
		`destinationType`
		`{`
		`gateway`
		`{`
		`protocol`
		`{`
		`voice :`
		`{`
		`supportedPrefixes`
		`{`
		`}`
		`}`
		`}`
		`}`

Action	Description

	mc FALSE
	undefinedNode FALSE
	}
	}
	value LCFnonStandardInfo ::=
	{
	termAlias
	{
	h323-ID : {"c3620-2-gw"},
	e164 : "001#19193920002"
	}
	gkID {"c3620-1-gk"}
	gateways
	{
	{
	gwType voip : NULL
	gwAlias
	{
	
	h323-ID : {"c3620-2-gw"},
	e164 : "001#19193920002"
	}
	sigAddress
	{
	ip 'AC12C250'H
	port 1720
	}
	resources
	{
	maxDSPs 0
	inUseDSPs 0
	maxBChannels 0
	inUseBChannels 0
	activeCalls 0
	bandwidth 0
	inuseBandwidth 0
	}
	}
	}
	}

	Action	Description

Step 6	SIP INVITE—SIP proxy server forwards the INVITE to the gateway	SIP proxy server receives the RAS LCF message, decode it and obtain the gateway transport address (172.18.194.80) value from the callSignalAddress ipAddress field of the LCF message. It then adds the SIP port number (5060) and forwards the INVITE to the gateway. Since the 001# tech-prefix flag is turned on in the sipd.conf file, the 001# string is not stripped from the request-URI.
		
		Example
		`INVITE sip:001#19193920002@172.18.194.80:5060;`
		`user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP  161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP PhoneCSeq:1`
		`INVITEMax-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Callt=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

Step 7	SIP 180 Ringing—Gateway sends 180 Ringing back to the SIP proxy server	The SIP/H.323 gateway receives the forwarded SIP INVITE message from the SIP proxy server and sends it downstream. Assume the call signal reaches the end-point and a SIP 180 Ringing is sent from the gateway up to the SIP proxy server.
		
		Example
		`SIP/2.0 180 Ringing`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 8	SIP 180 Ringing—SIP proxy server forwards to the UAC	SIP proxy server receives the 180 Ringing from the gateway, it found the record in TCB and forwards the 180 Ringing upstream to the UAC.
		
		Example
		`SIP/2.0 180 Ringing`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

Step 9	SIP 200 OK—Gateway sends 200 OK to upstream SIP proxy server	The called party picks up the phone. The gateway sends a 200 OK to the SIP proxy server.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Contact: <sip:001#19195550002@172.18.194.80>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`
		`v=0`
		`o=CiscoSystemsSIP- gateway 537556 235334 IN IP4 172.18.194.80`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 gateway.cisco.com`
		`m=audio 5004 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

Step 10	SIP 200 OK—SIP proxy server forward the 200 OK to the calling UAC	SIP proxy server receives the 200 OK from the gateway. It forwards it upstream to the calling UAC.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Contact: <sip:001#19195550002@172.18.194.80>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`
		
		`v=0`
		`o=CiscoSystemsSIP- gateway 537556 235334 IN IP4 172.18.194.80`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 gateway.cisco.com`
		`m=audio 5004 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

	Action	Description

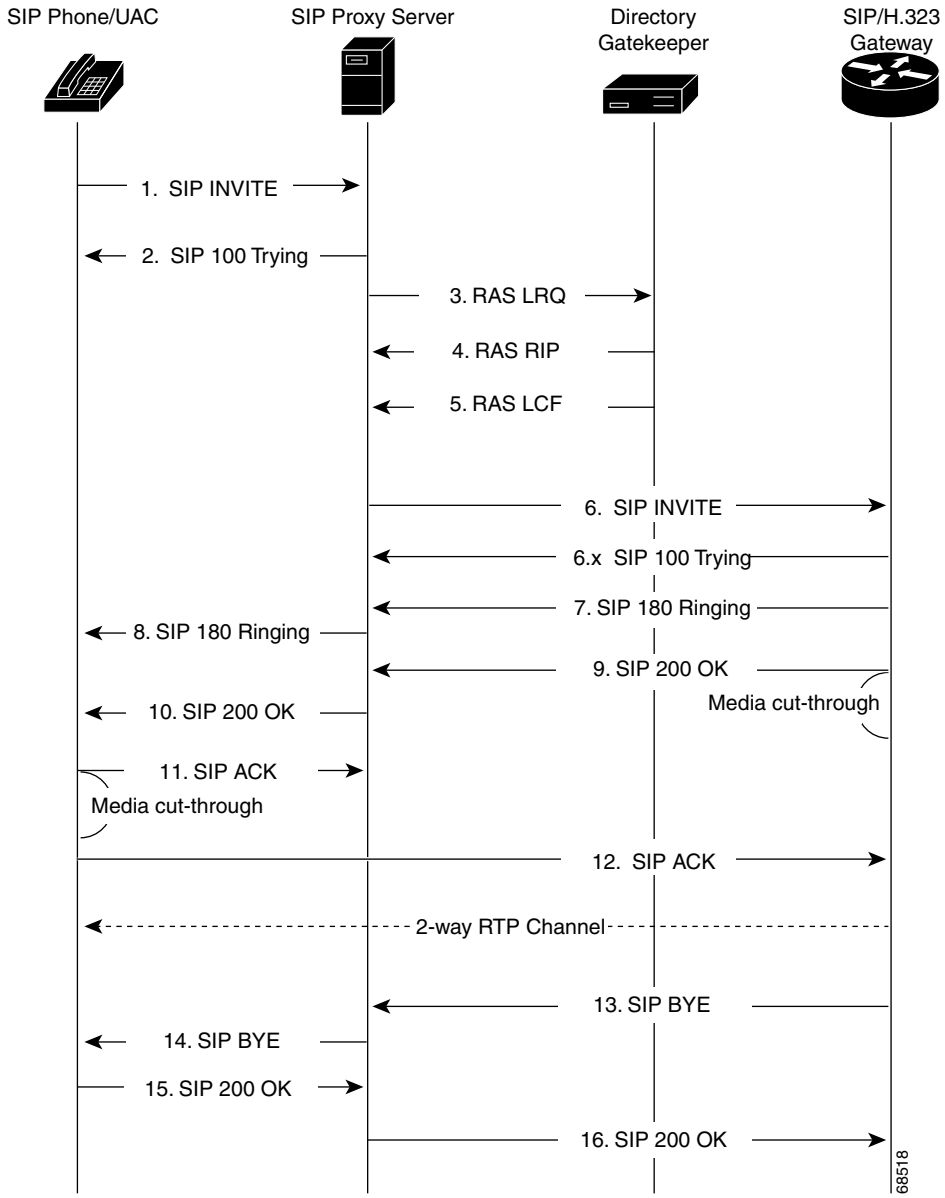
Step 11	SIP ACK—Calling UAC sends ACK directly to the gateway	Upon receiving the 200 OK message, the UAC opens the media port and responds with ACK directly to the gateway.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

Step 12	SIP BYE—Gateway sends BYE to the calling UAC	The callee hangs up the phone. The gateway sends a BYE to the calling UAC.
		
		Example
		`SIP/2.0 BYE`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 BYE`

Step 13	SIP 200 OK—Calling UAC returns a 200 OK to the gateway	The calling UAC receives the BYE from the gateway, it returns a 200 OK.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 BYE`

SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Enabled

Figure E-19 SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Enabled



	Action	Description

Step 1	INVITE—SIP phone to SIP proxy server	SIP UAC sends an INVITE request to the SIP proxy server.
		
		Example
		`INVITE sip:20002@proxy.cisco.com;user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`
		
		The following applies:
		• The phone number of the called party is inserted in the Request-URI field in the form of a SIP URL.
		• A unique numeric identifier is assigned to the call and is inserted in the Call-ID field.The transaction number within a single call leg is identified in the Cseq field.
		The media capability the calling party is ready to receive is specified.

Step 2	100—Trying SIP Proxy sends to UAC	SIP proxy server sends 100-Trying response message to the upstream UAC upon receiving the INVITE in step 1.
		
		Example
		`SIP/2.0 100 Trying`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "255-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 3	RAS LRQ—SIP Proxy sends a RAS LRQ message to a DGK	SIP proxy server expands the 20002 number into a 9193920002 number but finds no static route to route the request. It then invokes the new routing module and creates an LRQ RAS message from the incoming INVITE SIP message. The LRQ message is sent to one of the DGK configured in the sipd.conf file.
		
		SIP proxy server prepends a technology prefix 001# to the expanded number and uses it to fill the destinationInfo field of the LRQ RAS message. The (decoded) RAS LRQ looks like the following example:
		
		Example
		`value RasMessage ::= locationRequest :`
		`{`
		`  requestSeqNum 2519`
		`  destinationInfo`
		`  {`
		`  e164 : "001#19193920002"`
		`  }`
		`  nonStandardData`
		`  {`
		`  nonStandardIdentifier h221NonStandard :`
		`  {`
		`  t35CountryCode 181`
		`  t35Extension 0`
		`  manufacturerCode 18`
		`  }`
		`data '8284901100ECAA98A025220008000000000E1963...'H`
		`  }`
		`  replyAddress ipAddress :`
		`  {`
		`  ip 'AC12C247'H`
		`  port 1719`
		`  }`
		`  sourceInfo`
		`  {`
		`  h323-ID : {"genuity-sip1"}`
		`  }`
		`  canMapAlias TRUE`
		`  }`

Step 4	RAS RIP—H.323 DGK returns a RIP to the SIP proxy server	Upon receiving the RAS LRQ message from the SIP proxy server, H.323 DGK can return a RIP with delay timer value. SIP server should adjust timer accordingly.
		
		Example
		`value RasMessage ::= requestInProgress :`
		`{`
		`  requestSeqNum 2519`
		`  delay 9000`
		`}`

	Action	Description

Step 5	RAS LCF—H.323 DGK returns a LCF to the SIP proxy server	H.323 DGK forwards the request to the H.323 network and finds a SIP/H.323 gateway that can handle this particular call. It then returns a RAS LCF message to the SIP proxy server.
		
		Example
		`value RasMessage ::= locationConfirm :`
		`{`
		`requestSeqNum 2519`
		`callSignalAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 1720`
		`}`
		`rasAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 56812`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '0002400900630033003600320030002D0032002D...'H`
		`}`
		`destinationType`
		`{`
		`gateway`
		`{`
		`protocol`
		`{`
		`voice :`
		`{`
		`supportedPrefixes`
		`{`
		`}`
		`}`
		`}`
		`}`
		`mc FALSE`
		`undefinedNode FALSE`
		`}`
		`}`

Action	Description

	value LCFnonStandardInfo ::=
	{
	termAlias
	{
	h323-ID : {"c3620-2-gw"},
	e164 : "001#19193920002"
	}
	gkID {"c3620-1-gk"}
	gateways
	{
	{
	gwType voip : NULL
	gwAlias
	{
	h323-ID : {"c3620-2-gw"},
	e164 : "001#19193920002"
	}
	sigAddress
	{
	ip 'AC12C250'H
	port 1720
	}
	resources
	{
	maxDSPs 0
	inUseDSPs 0
	maxBChannels 0
	inUseBChannels 0
	activeCalls 0
	bandwidth 0
	inuseBandwidth 0
	}
	}
	}
	}

Step 6 SIP INVITE—SIP proxy server forwards the INVITE to the gateway	**SIP proxy server forwards the INVITE to the gateway.** Example To: <sip:20002@company.com;user=phone;phone-context=000000> Date: Thu, 18 Mar 2000 04:48:28 UTC Call-ID: 23-99990146-0-5894369F@161.44.3.207 Cisco-Guid: 428806444-2576941380-0-1486104925 User-Agent: Cisco IP Phone CSeq:1 INVITE Max-Forwards: 6 Timestamp: 732430108 Contact: <sip:+19193920001@bounty.cisco.com:49489;user=phone> Expires: 5 Content-Type: application/sdp v=0 o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101 s=SIP Call t=0 0 c=IN IP4 172.18.193.101 m=audio 20354 RTP/AVP 0 3 a=rtpmap:0 PCMU/8000 a=rtpmap:3 GSM/8000

	Action	Description

Step 7	SIP 180 Ringing—Gateway sends 180 Ringing back to the SIP proxy server	SIP/H.323 gateway receives the forwarded SIP INVITE message from the SIP proxy server and sends it downstream. Assume the call signal reaches the end-point and a SIP 180 Ringing is sent from the gateway up to the SIP proxy server.
		
		Example
		`SIP/2.0 180 Ringing`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Record-Route: < sip:001#9195550002@proxy.cisco.com;`
		`maddr=proxy.cisco.com>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

Step 8	SIP 180 Ringing—SIP proxy server forwards to the UAC	SIP proxy server receives the 180 Ringing from the gateway, it found the record in TCB and forwards the 180 Ringing upstream to the UAC.
		
		Example
		`SIP/2.0 180 Ringing`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Record-Route: < sip:001#9193920002@proxy.cisco.com;`
		`maddr=proxy.cisco.com>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

Step 9	SIP 200 OK—Gateway sends 200 OK to upstream SIP proxy server	The called party picks up the phone. The gateway sends a 200 OK to the SIP proxy server.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Record-Route: < sip:001#9193920002@proxy.cisco.com;`
		`maddr=proxy.cisco.com>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Contact: <sip:001#19193920002@172.18.194.80>`
		`Content-Length: 0`
		`v=0`
		`o=CiscoSystemsSIP- Gateway 537556  235334 IN IP4 172.18.194.80`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 gateway.cisco.com`
		`m=audio 5004 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

	Action	Description

Step 10	SIP 200 OK—SIP proxy server forward the 200 OK to the calling UAC	SIP proxy server receives the 200 OK from the gateway. It forwards it upstream to the calling UAC.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Record-Route: < sip:001#19193920002@proxy.cisco.com;`
		`maddr=proxy.cisco.com>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Contact: <sip:001#19193920002@172.18.194.80>`
		`Content-Length: 0`
		
		`v=0`
		` o=CiscoSystemsSIP- Gateway 537556  235334 IN IP4 172.18.194.80`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 gateway.cisco.com`
		`m=audio 5004 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

Step 11	SIP ACK—Calling UAC sends ACK to the SIP proxy	The caller UAC opens the media port and responds with an ACK to the SIP proxy.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Route: <sip:001#19193920002@172.18.194.80>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

Step 12	SIP ACK—SIP proxy forwards an ACK to the gateway	SIP proxy server forwards the ACK to the downstream gateway.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 172.18.194.80:48987`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:20002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

	Action	Description

Step 13	SIP BYE—Gateway sends BYE to the SIP proxy	The callee hang up the phone. The gateway sends a BYE to the SIP proxy.
		Example
		`SIP/2.0 BYE sip: +19195550001@bounty.cisco.com`
		`Via: SIP/2.0/UDP 172.18.194.80:5060`
		`Route: < sip: +19195550001@ bounty.cisco.com >`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+19193920002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

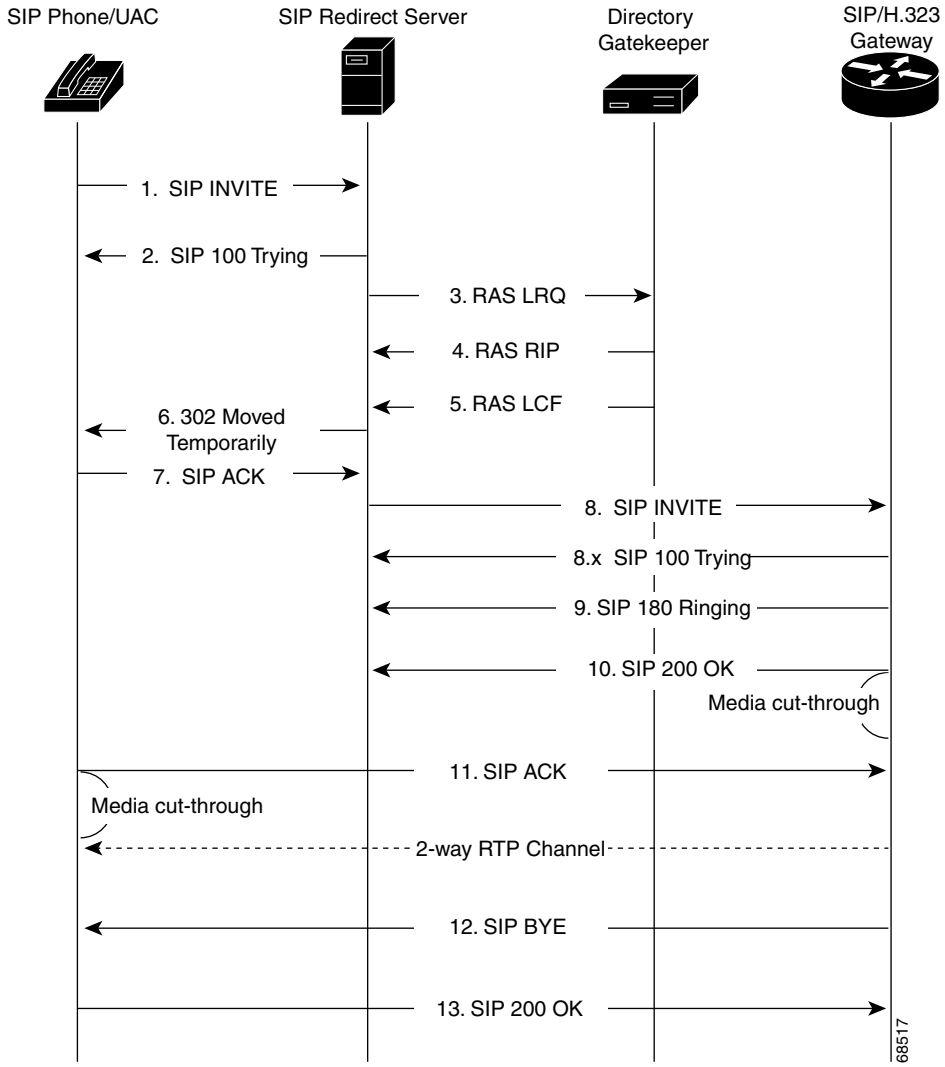
Step 14	SIP BYE—SIP proxy forwards BYE to the calling party	SIP proxy server receives the BYE from the gateway and forwards it upstream to the calling user agent.
		Example
		`SIP/2.0 BYE sip: +19195550001@ bounty.cisco.com:5060`
		`Via: SIP/2.0/UDP 172.18.194.80:5060`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Record-Route: <sip: +19195550001@proxy.cisco.com>`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+19193920002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

Step 15	SIP 200 OK—Calling UAC returns a 200 OK to the SIP proxy	The calling UAC receives the BYE from the gateway, it returns a 200 OK to the SIP proxy.
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+19193920002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

Step 16	SIP 200 OK—SIP proxy forwards the 200 OK to the gateway	SIP proxy receives the 200 OK from the calling UAC and forwards it to the gateway.
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP proxy.cisco.com:5060`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+19193920002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

SIP Phone to SIP/H.323 Gateway via SIP Redirect Server

Figure E-20 SIP Phone to SIP/H.323 Gateway via SIP Redirect Server



	Action	Description

Step 1	INVITE—SIP phone to SIP redirect server	SIP UAC sends an INVITE request to the SIP redirect server.
		
		Example
		`INVITE sip:50002@redirect.cisco.com;user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`
		
		The following applies:
		• The phone number of called party is inserted in the Request-URI field in the form of a SIP URL.
		• A unique numeric identifier is assigned to the call and is inserted in the Call-ID field.
		• The transaction number within a single call leg is identified in the Cseq field.
		• The media capability the calling party is ready to receive is specified.

Step 2	100—Trying SIP redirect server returns 100 Trying to UAC	SIP redirect server sends 100-Trying response message to the upstream UAC upon receiving the INVITE in step 1.
		
		Example
		`SIP/2.0 100 Trying`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 3	RAS LRQ—SIP redirect server sends a RAS LRQ message to a DGK	The SIP redirect server expands the 50002 number into a 9193650002 number but finds no static route. It then invokes the new routing module and creates an LRQ RAS message from the incoming INVITE SIP message. The LRQ message is sent to one of the DGK configured in the sipd.conf file.
		
		The SIP redirect server prepends a technology prefix 002# to the expanded number and uses it to fill the destinationInfo field of the LRQ RAS message.
		
		Example
		`value RasMessage ::= locationRequest :`
		`{`
		`requestSeqNum 2519`
		`destinationInfo`
		`{`
		`e164 : "002#19193650002"`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '8284901100ECAA98A025220008000000000E1963...'H`
		`}`
		`replyAddress ipAddress :`
		`{`
		`ip 'AC12C247'H`
		`port 1719`
		`}`
		`sourceInfo`
		`{`
		`h323-ID : {"genuity-sip1"}`
		`}`
		`canMapAlias TRUE`
		`}`

Step 4	RAS RIP—H.323 DGK returns a RIP to the SIP redirect server	Upon receiving the RAS LRQ message from the SIP redirect server, the H.323 DGK can return a RIP with delay timer value. SIP server should adjust timer accordingly.
		
		Example
		`value RasMessage ::= requestInProgress :`
		`{`
		`requestSeqNum 2519`
		`delay 9000`
		`}`

	Action	Description

Step 5	RAS LCF—H.323 DGK returns a LCF to the SIP redirect server	The H.323 DGK forwards the request to the H.323 network and finds a SIP/H.323 gateway that can handle this particular call. It then returns a RAS LCF message to the SIP redirect server.
		
		Example
		`value RasMessage ::= locationConfirm :`
		`{`
		`requestSeqNum 2519`
		`callSignalAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 1720`
		`}`
		`rasAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 56812`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '0002400900630033003600320030002D0032002D...'H`
		`}`
		`destinationType`
		`{`
		`gateway`
		`{`
		`protocol`
		`{`
		`voice :`
		`{`
		`supportedPrefixes`
		`{`
		`}`
		`}`
		`}`
		`}`
		`mc FALSE`
		`undefinedNode FALSE`
		`}`
		`}`

	Action	Description

		value LCFnonStandardInfo ::=
		{
		termAlias
		{
		h323-ID : {"c3620-2-gw"},
		e164 : "4056701000"
		}
		gkID {"c3620-1-gk"}
		gateways
		{
		{
		gwType voip : NULL
		gwAlias
		{
		h323-ID : {"c3620-2-gw"},
		e164 : "002#19193650002"
		}
		sigAddress
		{
		ip 'AC12C250'H
		port 1720
		}
		
		resources
		{
		maxDSPs 0
		inUseDSPs 0
		maxBChannels 0
		inUseBChannels 0
		activeCalls 0
		bandwidth 0
		inuseBandwidth 0
		}
		}
		}
		}

Step 6	SIP 302 Moved Temporarily—SIP redirect server sends a 302 Moved Temporarily to the UAC	The SIP redirect server receives the RAS LCF message, decodes it, and obtains the gateway transport address (172.18.194.80) from the callSignalAddress ipAddress field of the LCF message. It then adds the SIP port number (5060) and returns the 302 Moved Temporarily message back to the UAC. Since the 002# tech-prefix flag is turned off in the sipd.conf file, the 002# string is stripped from the contact header. **Example** SIP/2.0 302 MovedTemporarily Via: SIP/2.0/UDP 161.44.3.207:49489 Call-ID: 23-99990146-0-5894369F@161.44.3.207 From: "555-0001" <sip:+19195550001@bounty.cisco.com> To: <sip:50002@company.com;user=phone;phone-context=000000> CSeq: 1 INVITE Contact: <sip:19193650002@172.18.194.80:5060> Content-Length: 0

	Action	Description

Step 7	SIP ACK—UAC returns a SIP ACK to the redirect server	Upon receiving of the 302 response message, the UAC returns a SIP ACK to the redirect server.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

Step 8	SIP INVITE—UAC sends directly to the gateway	The UAC sends a new INVITE directly to the gateway.
		
		Example
		`INVITE sip:19193650002@172.18.194.80:5060;`
		`user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq: 2 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

Step 9	SIP 180 Ringing—Gateway sends 180 Ringing back to the UAC	The SIP/H.323 gateway receives the SIP INVITE message from the UAC and sends it downstream. Assume the call signal reaches the end-point and a SIP 180 Ringing is sent from the gateway to the UAC.
		
		Example
		`SIP/2.0 180 Ringing`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 2 INVITE`
		`Content-Length: 0`

	Action	Description

Step 10	SIP 200 OK—Gateway sends 200 OK to the calling UAC	The called party picks up the phone and the gateway sends 200 OK to the calling UAC.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 2 INVITE`
		`Content-Length: 0`
		
		`v=0`
		` o=CiscoSystemsSIP- Gateway 537556 235334 IN IP4 172.18.194.80`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 gateway.cisco.com`
		`m=audio 5004 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

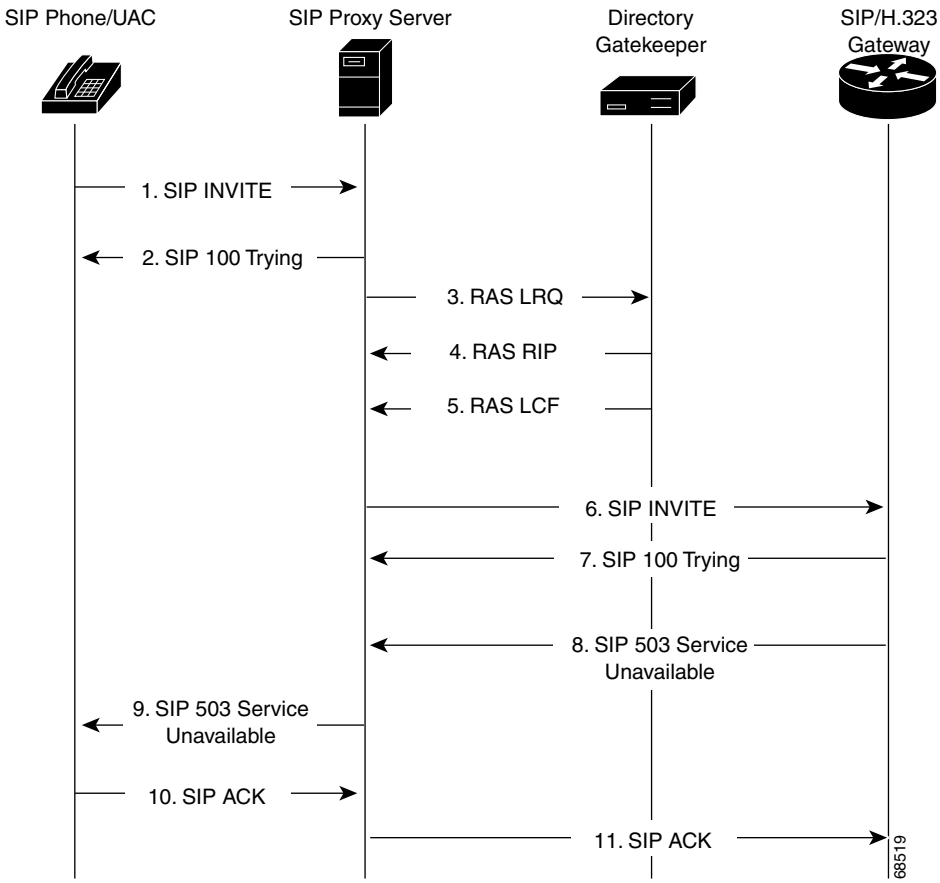
Step 11	SIP ACK—Calling UAC sends ACK to the gateway	Upon receiving the 200 OK message, the UAC opens the media port and responds with ACK to the gateway.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 2 ACK`

Step 12	SIP BYE—Gateway sends BYE to the calling UAC	User hangs up the phone. The gateway sends a BYE to the calling UAC.
		
		Example
		`SIP/2.0 BYE sip:+19195550001@bounty.cisco.com`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+1913650002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

Step 13	SIP 200 OK—Calling UAC returns a 200 OK to the gateway	Calling UAC receives the BYE from the gateway, it returns a 200 OK.
		
		Example
		`SIP/2.0 200 OK`
		`Via: SIP/2.0/UDP 172.18.194.80:43576`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: <sip:+19193650002@company.com;user=phone>`
		`To: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`CSeq: 1 BYE`

SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Disabled (Call Failed with a 503 Service Unavailable Response)

Figure E-21 SIP Phone to SIP/H.323 Gateway via SIP Proxy Server—Record-Route Disabled



	Action	Description

Step 1	INVITE-SIP phone to SIP proxy server	SIP UAC sends an INVITE request to the SIP proxy server.
		
		Example
		`INVITE sip:50002@proxy.cisco.com;user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`
		
		The following applies:
		• The phone number of called party is inserted in the Request-URI field in the form of a SIP URL.
		• A unique numeric identifier is assigned to the call and is inserted in the Call-ID field.
		• The transaction number within a single call leg is identified in the Cseq field.
		The media capability the calling party is ready to receive is specified.

Step 2	100-Trying SIP Proxy sends to UAC	SIP proxy server sends 100-Trying response message to the upstream UAC upon receiving the INVITE in step 1.
		
		Example
		`SIP/2.0 100 Trying`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 3	RAS LRQ—SIP Proxy sends a RAS LRQ message to a DGK	SIP proxy server expands the 50002 number into a 9193650002 number but finds no static route to route the request. It then invokes the new routing module, creates an LRQ RAS message from the incoming INVITE SIP message, and sends the LRQ message to one of the DGK configured in the sipd.conf file.
		
		SIP proxy server adds a technology prefix 002# to the expanded number and uses it to fill the destinationInfo field of the LRQ RAS message.
		
		Example
		`value RasMessage ::= locationRequest :`
		`{`
		`requestSeqNum 2519`
		`destinationInfo`
		`{`
		`e164 : "002#19193650002"`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '8284901100ECAA98A025220008000000000E1963...'H`
		`}`
		`replyAddress ipAddress :`
		`{`
		`ip 'AC12C247'H`
		`port 1719`
		`}`
		`sourceInfo`
		`{`
		`h323-ID : {"genuity-sip1"}`
		`}`
		`canMapAlias TRUE`
		`}`

Step 4	RAS RIP—H.323 DGK returns a RIP to the SIP proxy server	Upon receiving the RAS LRQ message from the SIP proxy server, the H.323 DGK can return a RIP with delay timer value. SIP server should adjust timer accordingly.
		
		Example
		`value RasMessage ::= requestInProgress :`
		`{`
		`requestSeqNum 2519`
		`delay 9000`
		`}`

	Action	Description

Step 5	RAS LCF—H.323 DGK returns a LCF to the SIP proxy server	The H.323 DGK forwards the request to the H.323 network and finds a SIP/H.323 gateway that can handle this particular call. It then returns a RAS LCF message to the SIP proxy server.
		
		Example
		`value RasMessage ::= locationConfirm :`
		`{`
		`requestSeqNum 2519`
		`callSignalAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 1720`
		`}`
		`rasAddress ipAddress :`
		`{`
		`ip 'AC12C250'H`
		`port 56812`
		`}`
		`nonStandardData`
		`{`
		`nonStandardIdentifier h221NonStandard :`
		`{`
		`t35CountryCode 181`
		`t35Extension 0`
		`manufacturerCode 18`
		`}`
		`data '0002400900630033003600320030002D0032002D...'H`
		`}`
		`destinationType`
		`{`
		`gateway`
		`{`
		`protocol`
		`{`
		`voice :`
		`{`
		`supportedPrefixes`
		`{`
		`}`
		`}`
		`}`
		`mc FALSE`
		`undefinedNode FALSE`
		`}`
		`}`

Action	Description

	value LCFnonStandardInfo ::=
	{
	termAlias
	{
	h323-ID : {"c3620-2-gw"},
	e164 : "002#19193650002"
	}
	gkID {"c3620-1-gk"}
	gateways
	{
	{
	gwType voip : NULL
	gwAlias
	{
	h323-ID : {"c3620-2-gw"},
	e164 : "002#19193650002"
	}
	sigAddress
	{
	ip 'AC12C250'H
	port 1720
	}
	resources
	{
	maxDSPs 0
	inUseDSPs 0
	maxBChannels 0
	inUseBChannels 0
	activeCalls 0
	bandwidth 0
	inuseBandwidth 0
	}
	}
	}
	}

	Action	Description

Step 6	SIP INVITE—SIP proxy server forwards the INVITE to the gateway	SIP proxy server receives the RAS LCF message, decodes it, and obtains the gateway transport address (172.18.194.80) from the callSignalAddress ipAddress field of the LCF message. It then adds the SIP port number (5060) and forwards the INVITE to the gateway. Since the 002# tech-prefix flag is turned off in the sipd.conf file, the 002# string is stripped from the request-URI.
		
		Example
		`INVITE sip: 19193650002@172.18.194.80:5060;`
		`user=phone;phone-context=000000 SIP/2.0`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`Date: Thu, 18 Mar 2000 04:48:28 UTC`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`Cisco-Guid: 428806444-2576941380-0-1486104925`
		`User-Agent: Cisco IP Phone`
		`CSeq:1 INVITE`
		`Max-Forwards: 6`
		`Timestamp: 732430108`
		`Contact: <sip:+19195550001@bounty.cisco.com:49489;user=phone>`
		`Expires: 5`
		`Content-Type: application/sdp`
		
		`v=0`
		`o=CiscoSystemsSIP- UserAgent 8870 5284 IN IP4 172.18.193.101`
		`s=SIP Call`
		`t=0 0`
		`c=IN IP4 172.18.193.101`
		`m=audio 20354 RTP/AVP 0 3`
		`a=rtpmap:0 PCMU/8000`
		`a=rtpmap:3 GSM/8000`

Step 7	SIP 100 Trying—Gateway sends 100 Trying back to the SIP proxy server	The SIP/H.323 gateway receives the forwarded SIP INVITE message from the SIP proxy server and sends 100-Trying back to the SIP proxy server.
		
		Example
		`SIP/2.0 100 Trying`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 8	SIP 100-Trying—SIP proxy server forwards to the UAC	SIP proxy server receives the 100-Trying from the gateway. It finds the record in TCB and forwards the 100-Trying upstream to the UAC.
		
		Example
		`SIP/2.0 100-Trying`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

Step 9	SIP 503 Service Unavailable—Gateway sends 503 Service Unavailable to upstream SIP proxy server	The gateway overloaded and sends a 503 Service Unavailable to the upstream SIP proxy server.
		
		Example
		`SIP/2.0 503 Service Unavailable`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

Step 10	SIP 503 Service Unavailable—SIP proxy server forwards the 503 Service Unavailable to the calling UAC	SIP proxy server receives the 503 Service Unavailable from the gateway and forwards it upstream to the calling UAC.
		
		Example
		`SIP/2.0 503 Service Unavailable`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 INVITE`
		`Content-Length: 0`

	Action	Description

Step 11	SIP ACK—Calling UAC sends ACK to the SIP proxy server	Upon receiving the 503 Service Unavailable message, the UAC responds with ACK to the SIP proxy server.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

Step 12	SIP ACK—SIP proxy server sends ACK to the downstream gateway	Upon receiving the ACK from UAC, the SIP proxy server forwards the ACK to the downstream gateway.
		
		Example
		`SIP/2.0 ACK`
		`Via: SIP/2.0/UDP proxy.cisco.com:48754; branch=1`
		`Via: SIP/2.0/UDP 161.44.3.207:49489`
		`Call-ID: 23-99990146-0-5894369F@161.44.3.207`
		`From: "555-0001" <sip:+19195550001@bounty.cisco.com>`
		`To: <sip:50002@company.com;user=phone;phone-context=000000>`
		`CSeq: 1 ACK`

Call-Flow Scenarios with CLIR Support

There are some typical or characteristic use cases where Calling Line Identity Restriction (CLIR) needs to be supported. In a setup that has at least three ATA phones (A1, A2, and A3, untrusted upstream and downstream), one SPS, one PSTN gateway (PGW) (trusted upstream and downstream), and two PSTN phones (P1 and P2, accessed via the PGW), we can have the following call setups, assuming that the initial call is anonymous and the second (forwarding) person, if any, is also set to anonymous when redirecting with 302, or has CLIR enabled when call forwarding is invoked:

- A1 call A2
- A1 call P1
- P1 call A1
- A1 call A2 forward to A3
- A1 call A2 forward to P1
- P1 call A1 forward to A2
- P1 call A1 forward to P2

The first three cases are covered by the last four cases, which add forwarding on top of them. The call flows shown here correspond to the last four cases, and assume that privacy-related directives are set to On, as follows:

- Privacy—On
- PrivacyWithPAI—On
- PrivacyWithRPID—On
- PrivacyWithDiversion—On

Note that setting these directives all to Off causes SPS to behave in the same way as before CLIR is supported.

For each of the four cases, there are two types of forwarding—302 redirected or Call Forwarding invoked in SPS—and therefore we show two call flows for each case.

A1 Call A2 Forward to A3

Figure E-22 Setup—A1 Call A2 Forward to A3

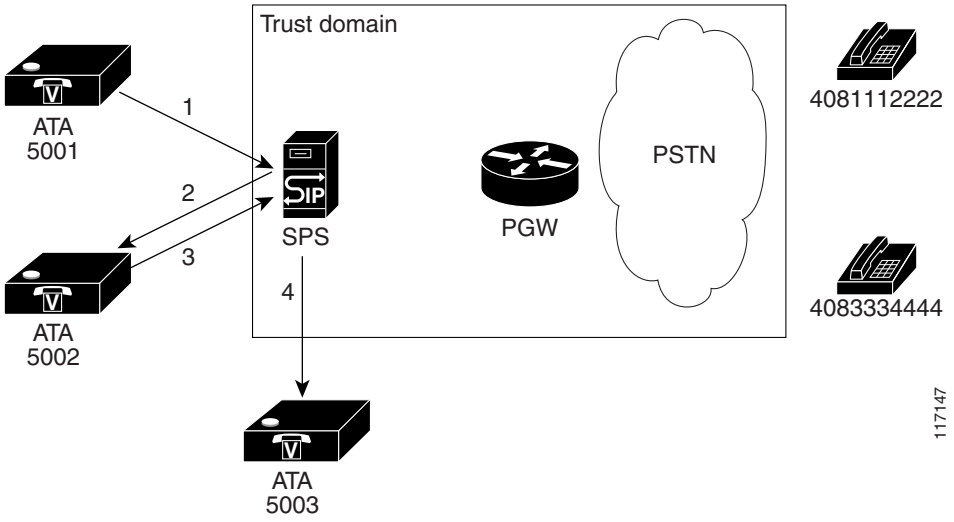


Figure E-23 Call Flow—A1 Call A2 Redirect to A3

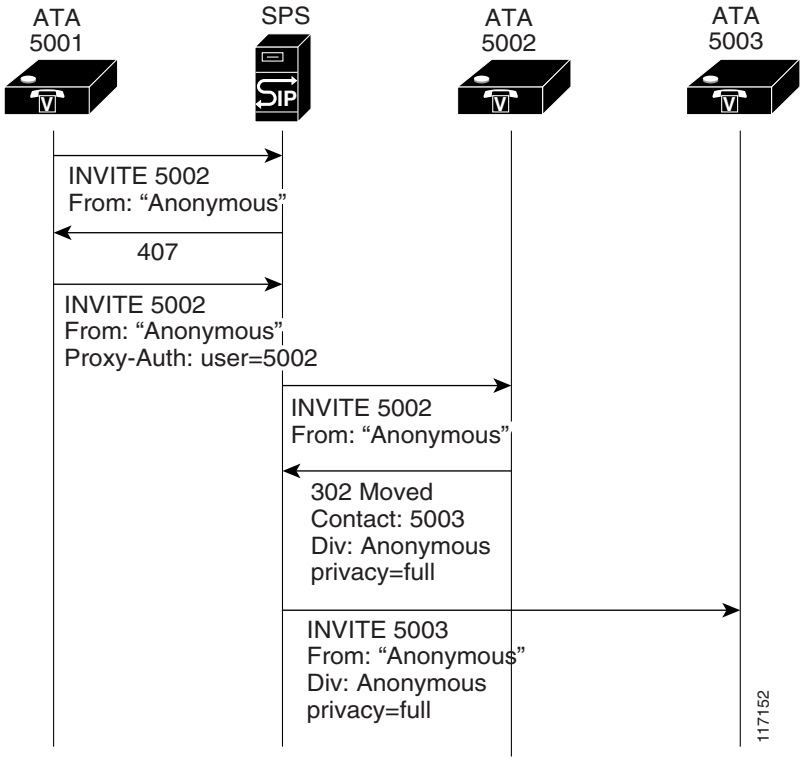
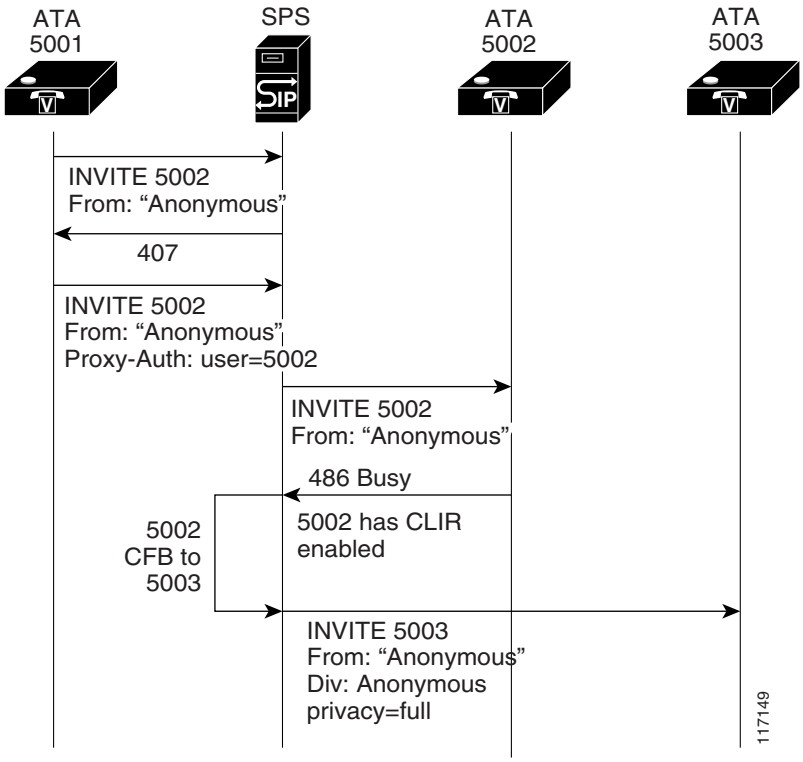


Figure E-24 Call Flow—A1 Call A2 Call Forward Busy to A3



A1 Call A2 Forward to P1

Figure E-25 Setup—A1 Call A2 Forward to P1

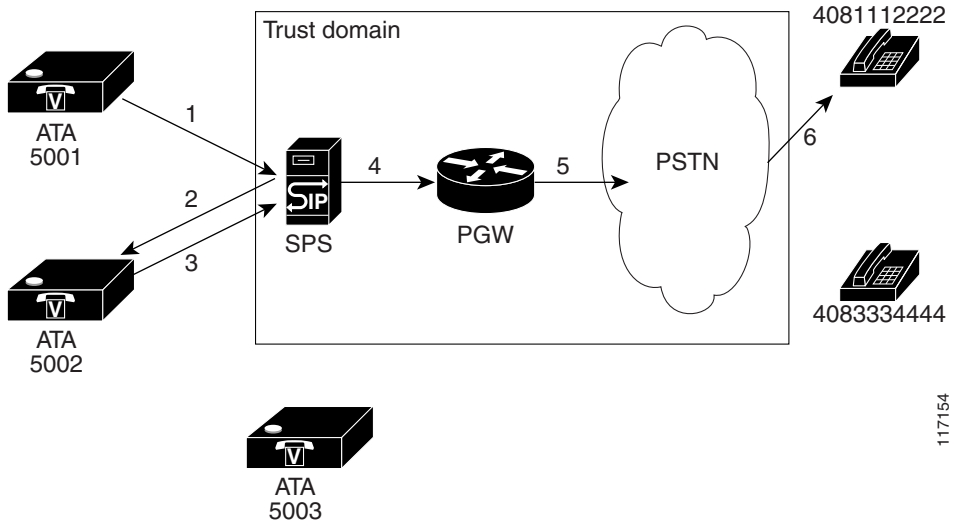


Figure E-26 Call Flow—A1 Call A2 Redirect to P1

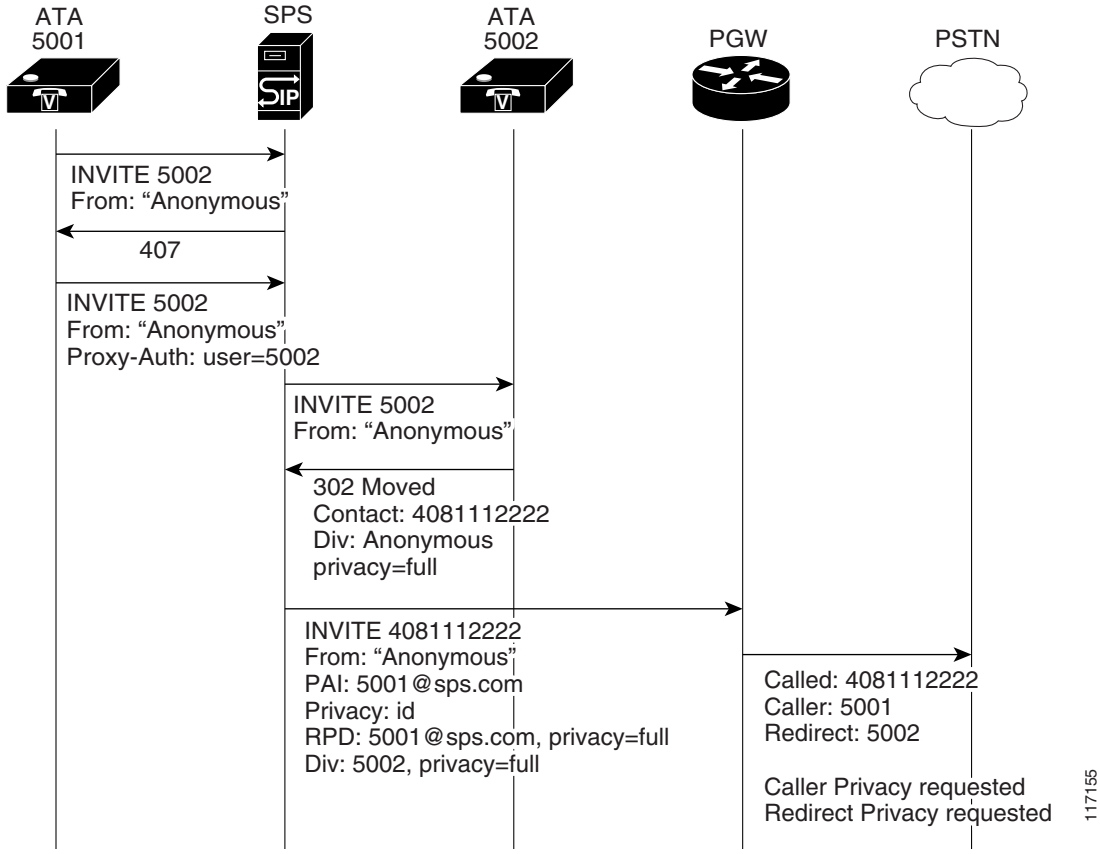
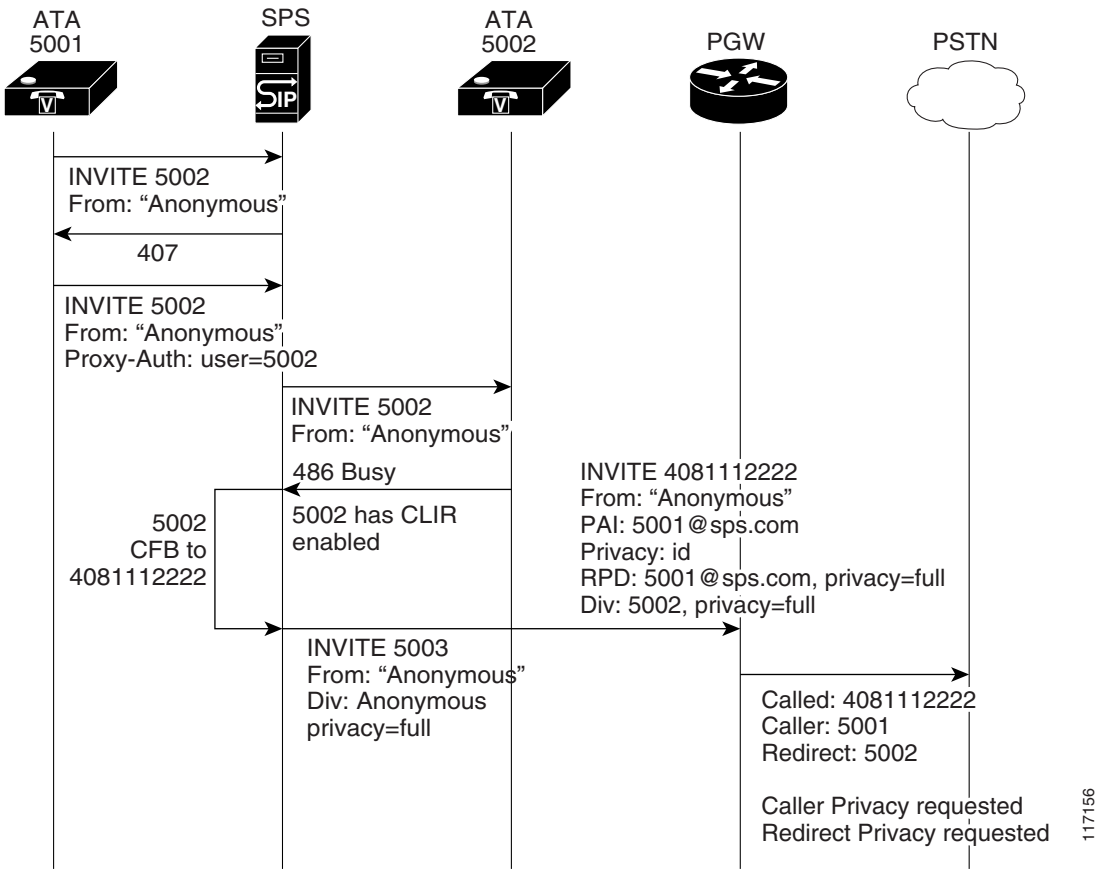


Figure E-27 Call Flow—A1 Call A2 Call Forward Busy to P1



P1 Call A1 Forward to A2

Figure E-28 Setup—P1 Call A1 Forward to A2

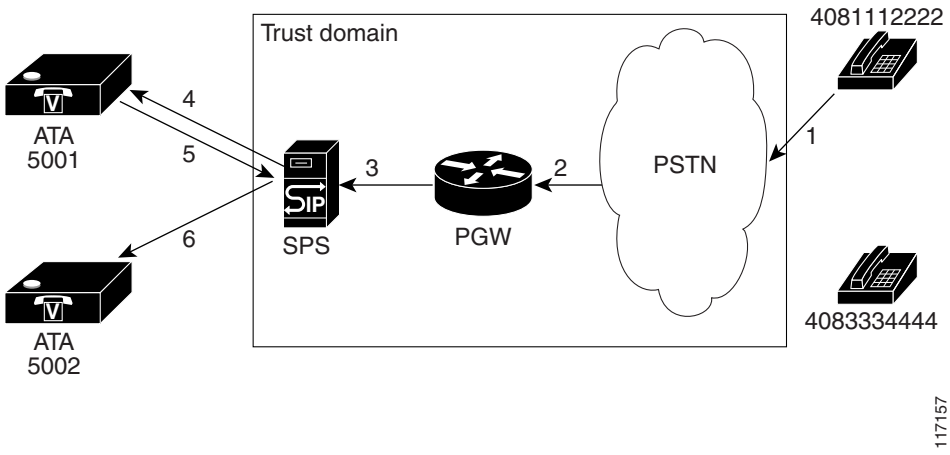


Figure E-29 Call Flow—P1 Call A1 Redirect to A2

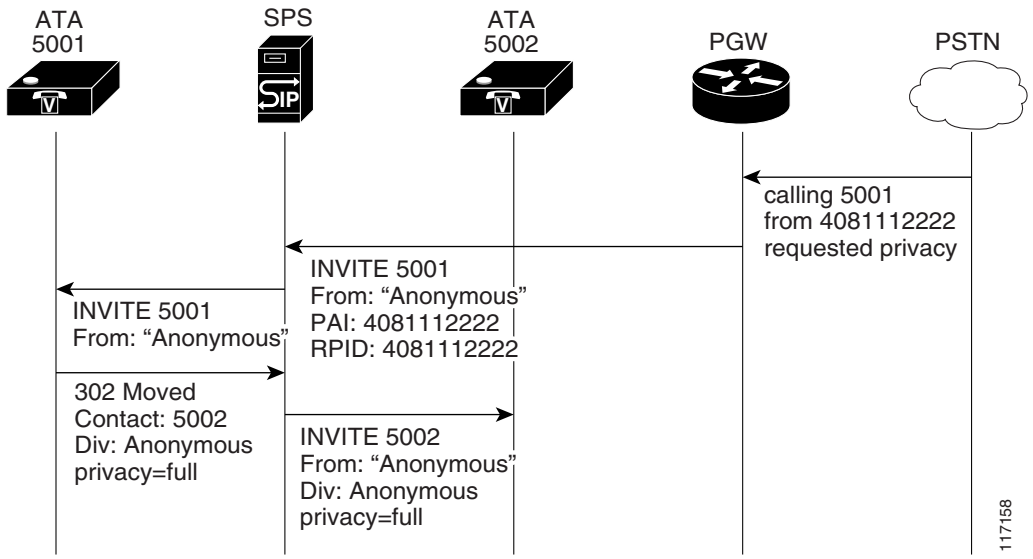
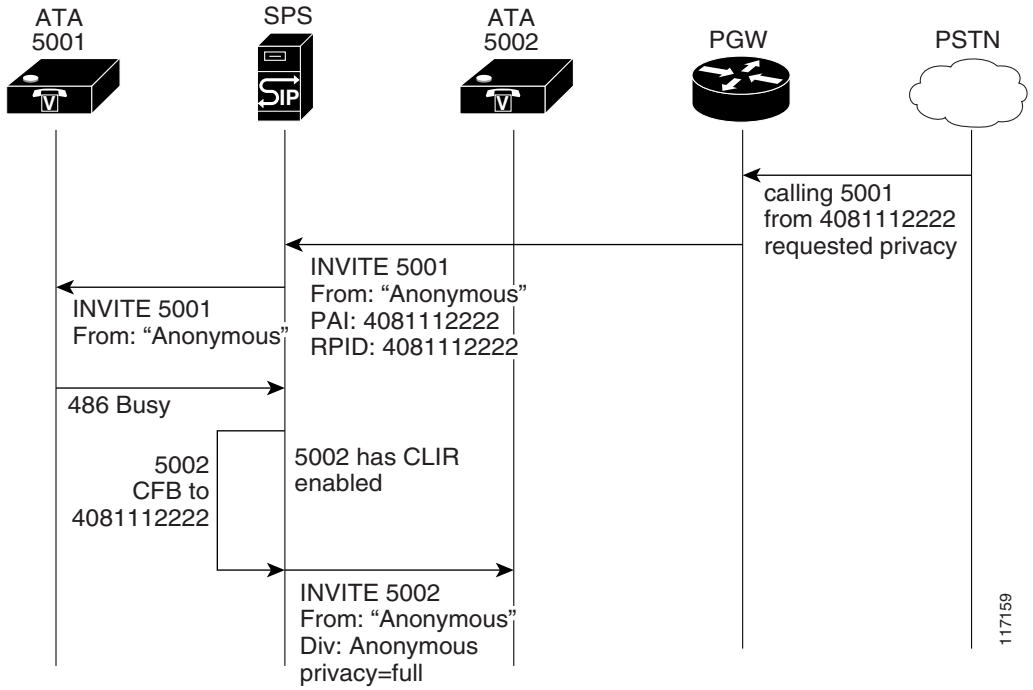


Figure E-30 Call Flow—P1 Call A1 Call Forward Busy to A2



P1 Call A1 Forward to P2

Figure E-31 Setup—P1 Call A1 Forward to P2

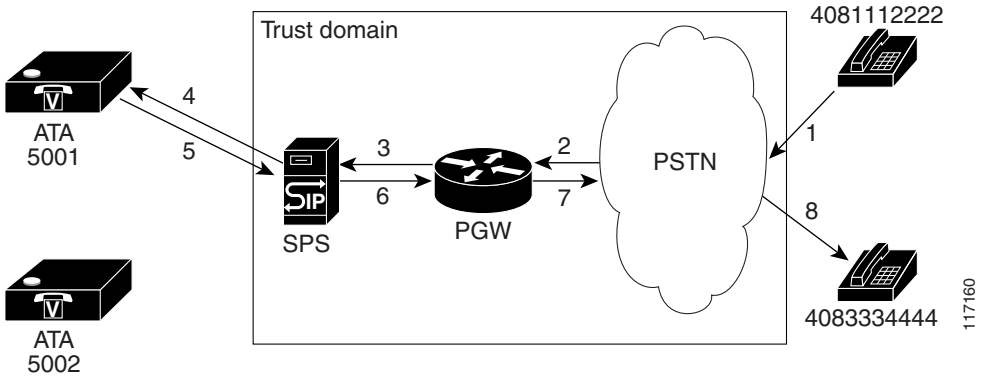


Figure E-32 Call Flow—P1 Call A1 Redirect to P2

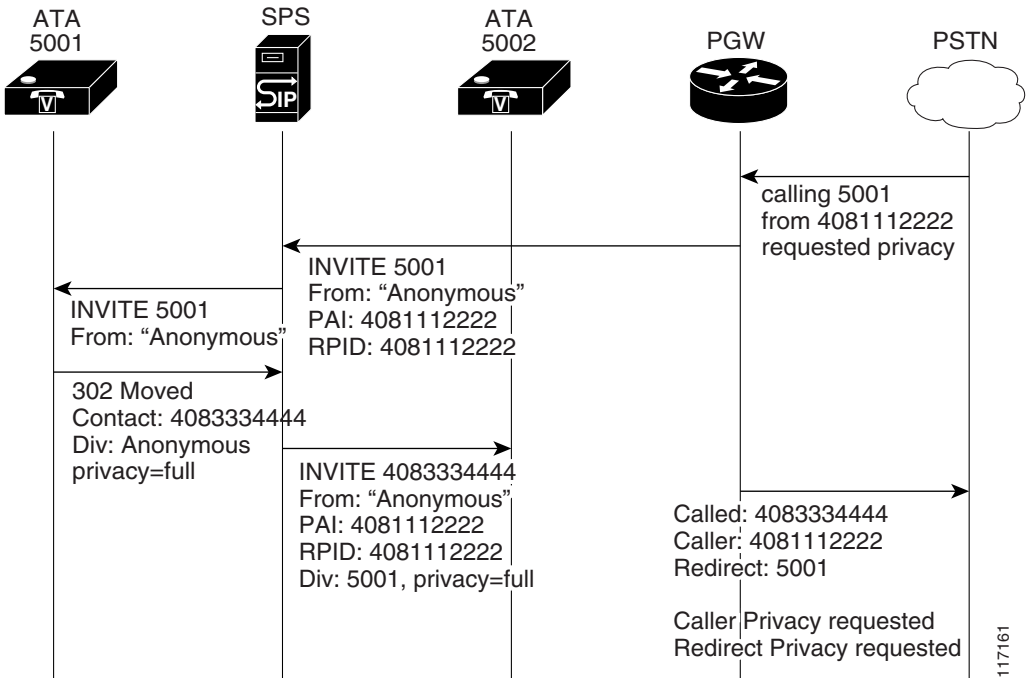
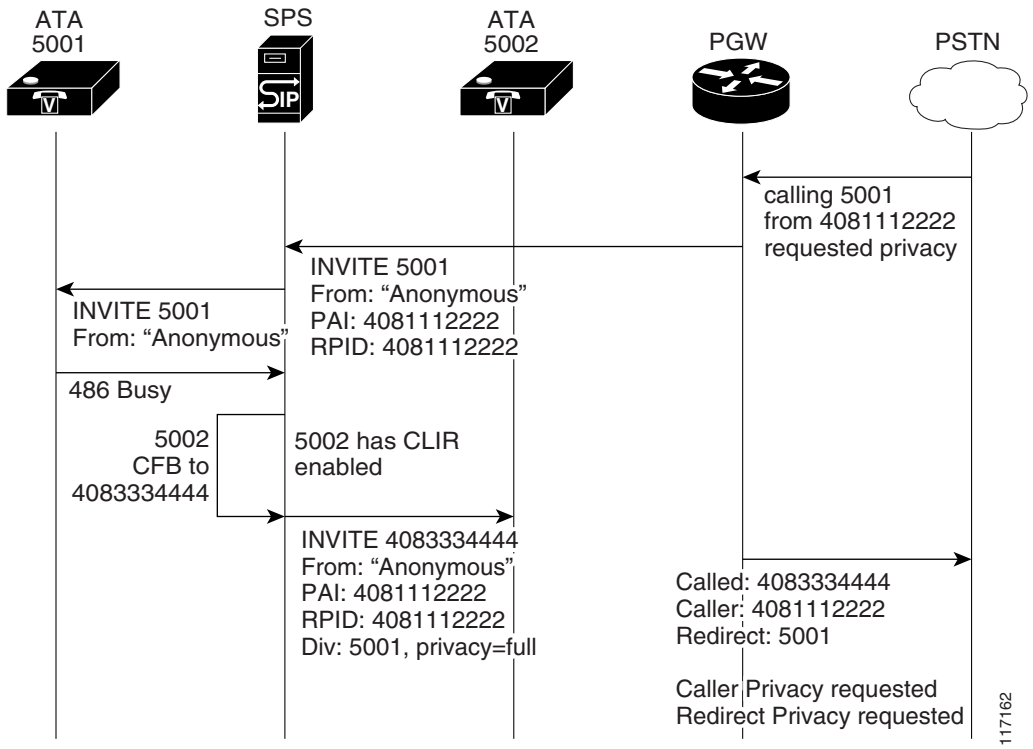


Figure E-33 Call Flow—P1 Call A1 Call Forward Busy to P2



GLOSSARY

NOTE: Refer to the *Cisco Dictionary of Internetworking Terms and Acronyms* **for terms not included in this glossary.**

A

AAA Authentication, authorization, and accounting. The network security services that provide the primary framework through which you set up access control on your router or access server.

address resolution A method for resolving differences produced by the use of computer addressing schemes. Address resolution usually specifies a method for mapping network layer (Layer 3) addresses to data link layer (Layer 2) addresses.

agent An object or application that can be a server, a client, or both.

ASCII American Standard Code for Information Interchange. An 8-bit code for character representation (7 bits plus parity).

awk A pattern-scanning and processing language.

B

bash Bourne-again shell. Interactive UNIX shell based on the traditional Bourne shell, but with increased functionality.

C

call Voice or data connection between two endpoints.

CBC Cipher block chaining. Encryption algorithm that combines an encrypted block with the previous block so that identical patterns in different messages are encrypted differently, depending upon the difference in the previous data.

CDR Call detail record. A record written to a database for use in postprocessing activities. This includes information such as where the call originated, start time, to whom the call was made, and when the call ended.

CHAP Challenge-Handshake Authentication Protocol.

cipher Cryptographic algorithm for encryption and decryption.

CMIP Common Management Information Protocol. OSI network management protocol created and standardized by ISO for the monitoring and control of heterogeneous networks.

codec Coder-decoder. Device that transforms analog voice into digital bit stream and vice-versa.

cron Clock daemon that starts a process that executes commands at a certain date and time.

crypto Encrytped information.

D

DES Data Encryption Standard. Standard cryptographic algorithm developed by the U.S. National Bureau of Standards.

DHCP Dynamic Host Control Protocol. A protocol used to dynamically allocate and assign IP addresses. DHCP allows you to move network devices from one subnet to another without administrative attention.

dial peer An addressable call endpoint. Voice over IP allows two types of dial peer: POTS and VoIP.

dial plan Description of the dialing arrangements for customer use on a network.

directive Configuration command that controls one or more aspects of system behavior. Directives reside in the system's configuration file.

DNIS Dialed number identification service (the called number). Feature of trunk lines where the called number is identified; this called-number information is used to route the call to the appropriate service. DNIS is a service used with toll-free dedicated services whereby calls placed to specific toll-free numbers are routed to the appropriate area within a company to be answered.

DNS Domain Naming System. A system used in the Internet for translating names of network nodes into addresses.

DSL Digital subscriber line. Public network technology that delivers high bandwidth over conventional copper wiring at limited distances. DSL is provisioned by means of modem pairs, with one modem located at a central office and the other at the customer site. Most DSL technologies do not use the whole bandwidth of the twisted pair, leaving room for a voice channel.

DTMF Dual-tone multifrequency. Tones generated when a button is pressed on a telephone, primarily used in the U.S. and Canada.

E

E.164 number space Global plan for telephone numbers wherein every device connected to the telephone network is assigned a unique numerical address.

E1 Wide-area digital transmission scheme used predominantly in Europe that carries data at a rate of 2.048 Mbps. E1 lines can be leased for private use from common carriers.

endpoint SIP or H.323 terminal or gateway. An endpoint can call and be called. It generates and terminates the information stream.

ENUM Informally, electronic number. DNS-based method for mapping phone numbers to IP addresses.

L

LCF message Location-confirm message. Message that contains the transport address of the destination endpoint that the gatekeeper sends in response to an LRQ message.

LDAP Lightweight Directory Access Protocol. A protocol that provides access for management and browser applications that provide read/write interactive access to the X.500 Directory. LDAP enables anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the Internet or on a corporate intranet. LDAP is a "lightweight" (smaller amount of code) version of DAP (Directory Access Protocol), which is part of X.500, a standard for directory services in a network.

LEC Local exchange carrier. A telephone company that provides customer access to the world-wide public switched network through one of its central offices.

lm License manager. Cisco SIP proxy server software that is automatically installed when the provisioning server (pserver) is installed. It handles the storage of license keys.

LNP Local number portability. Before Signaling System 7 (SS7), 800 numbers were not portable. If a company moved, they had to get a new number. The Telecom Act of 1996 mandated that personal phone numbers should also be portable. Telcos are required to support the porting of telephone numbers within a geographic area, increasing the demands on the SS7 network.

location server Device that processes requests (typically from a redirect or proxy server) to provide information about the possible location of a target end user.

LRJ message Location-reject message. Message that a gatekeeper sends to reject an LRQ message.

LRQ message Location-request message. Message that an endpoint sends to request that a gatekeeper provide address translation.

M

MGC Media gateway controller. A device that provides control of media and signaling gateways.

MGCP Media Gateway Control Protocol. Protocol that helps bridge the gap between circuit-switched and IP networks. It combines Internet Protocol Device Control (IPDC) and Simple Gateway Control Protocol (SGCP), and allows software programs to exert external control and management of data communications devices or media gateways at the edges of multiservice packet networks.

MIB Management Information Base. Database of network management information that is used and maintained by means of a network management protocol such as Simple Network Management Protocol (SNMP). The value of a MIB object can be changed or retrieved, usually through a GUI -based network-management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MySQL Database used to store and access provisioning system and subscriber feature data.

N

NAM Network application manager. A NAM contains a small configuration that allows it to directly route a subset of calls and dispatch the other requests.

name mapping The process of associating a name with a network location.

NAPTR record Naming-authority pointer record. Specifies a regular-expression-based rewrite rule that converts an existing string into a new domain label or uniform resource identifier (URI). This conversion enables the use of DNS to look up services for a variety of resource names that are not in domain-name syntax.

NAT Network Address Translation. Internet standard for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space.

next-hop routing Type of routing that relies on destination (next-hop) associations that tell a router that a particular destination can be reached optimally by sending the packet to a particular router representing the next hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop.

NTP Network Time Protocol. Protocol built on top of TCP that ensures accurate local time-keeping with reference to radio and atomic clocks located on the Internet. NTP is capable of synchronizing distributed clocks within milliseconds over long time periods.

P

PBX Private branch exchange. Digital or analog telephone switchboard located on the subscriber premises and used to connect private and public telephone networks.

PDU Protocol data unit. Another term for packet.

PEM Privacy-enhanced mail. Internet e-mail that provides confidentiality, authentication, and message integrity by means of various encryption methods. Not widely deployed in the Internet.

PID Protocol identifier. Field in a Call Request Packet message sent to an ISP host.

POTS Plain old telephone service. Basic telephone service supplying standard single-line telephones, telephone lines, and access to the public switched telephone network (PSTN).

proxy server Server that initiates requests on behalf of and receives requests from a client.

pserver Provisioning server. The main server used by the Cisco SPS GUI-based provisioning system.

PSTN Public switched telephone network. General term referring to the variety of telephone networks and services in place worldwide. Sometimes called POTS.

Q

QoP Quality of protection: authentication only, authentication and integrity, or both.

R

RADIUS Remote Authentication Dial-In User Service. An authentication and accounting system used by many internet service providers.

RAS Registration, Admission, and Status Protocol. Protocol that is used between endpoints and the gatekeeper to perform management functions. RAS signaling performs registration, admissions, bandwidth changes, status, and disengage procedures between the VoIP gateway and the gatekeeper.

redirect server Server that receives SIP requests from a client, strips out the address in the request, checks its address tables for any other addresses that might be mapped to the one in the request, and then returns the results of the address mapping to the client.

registrar server Server that accepts REGISTER requests from user-agent clients (UACs) for registration of their current location. Registrar servers are often colocated with proxy or redirect servers.

RFC Request for comments. Document series used as the primary means for communicating information about the Internet. Some RFCs are designated as Internet standards. Most RFCs document protocol specifications, such as Telnet and FTP, but some are humorous or historical. RFCs are available online from numerous sources.

RPC Remote-procedure call. Technological foundation of client/server computing. RPCs are procedure calls that are built or specified by clients and are executed on servers, with the results returned over the network to the clients.

RPM A Linux command-line-driven package-management system capable of installing, uninstalling, verifying, querying, and updating computer software packages.

RPMS Resource pool manager server. Server that enables telephone companies and Internet service providers to count, control, manage, and provide accounting data for shared resources for wholesale Virtual Private Dial-Up Network (VPDN) and non-VPDN dial network services across one or more network access server (NAS) stacks.

RR message Ready-to-receive message.

RSA Rivest, Shamir, and Adelman, inventors of the RSA public-key cryptographic system for encryption and authentication.

RSVP Resource Reservation Protocol. Protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive. RSVP depends on IPv6. Also known as Resource Reservation Setup Protocol.

RTCP RTP Control Protocol. Protocol that monitors the quality of service (QoS) of an IPv6 RTP connection and conveys information about the on-going session.

RTP Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

RTSP Real Time Streaming Protocol. Protocol that enables the controlled delivery of real-time data, such as audio and video. Sources of data can include both live data feeds such as live audio and video and stored content such as prerecorded events. RTSP is designed to work with established protocols such as RTP and HTTP.

S

SAP Session Announcement Protocol. A protocol used to assist in the advertisement of multicast multimedia conferences and other multicast sessions, and to communicate the relevant session setup information to prospective participants.

SDP Session Description Protocol. A protocol used to describe the characteristics of multimedia sessions for the purpose of session announcement, session invitation, and other forms of multimedia session initiation.

sed Stream editor. A software program that reads text files and makes editing changes according to a script of editing commands.

SES Severely errored second. A second during which the bit error ratio is greater than a specified limit and transmission performance is significantly degraded.

SHA-1 Secure Hash Algorithm 1. An algorithm that takes a message of fewer than 264 bits in length and produces a 160-bit message digest. The large message digest provides security against brute-force collision and inversion attacks.

signaling Process of sending a transmission signal for purposes of communication.

SIP Session Initialization Protocol. A protocol that offers many of the same architectural features as H.323, but relies on IP-specific technologies such as DNS. It also incorporates the concept of fixed port numbers for all devices and allows for the use of proxy servers.

sipd SIP proxy server. A server that handles all call processing and SIP messages.

SNMP Simple Network Management Protocol. A network management protocol used in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

spa SIP provisioning agent. Agent that resides on a Cisco SPS farm member and handles requests that the provisioning server gets from the GUI-based provisioning system. It receives requests from the provisioning server, accesses and updates (as needed) the SIP directives (sipd.conf) file, and provides feedback, by way of the provisioning server, to the GUI.

SRV record Server record. Record that allows administrators to use several servers for a single domain, to move services from host to host with little difficulty, and to designate some hosts as primary servers for a service and others as backups. Clients ask for a specific service or protocol for a specific domain and receive the names of any available servers.

SSL Secure Socket Layer. Encryption technology for the web used to provide secure transactions, such as the transmission of credit card numbers for e-commerce.

T

T1 Digital WAN carrier facility. T1 carries DS-1 formatted data at 1.544 Mbps through the telephone-switching network. T1 is the North American equivalent of an E1 line.

TCB Transaction control block. A data structure in which Cisco SPS stores from which it accesses the state information associated with SIP transactions.

TCL Toolkit Command Language. A scripting language used for gateway products both internally and externally to Cisco IOS software code.

TCP Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

TFTP Trivial File Transfer Protocol. Simplified version of FTP that allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password).

TLS Transport Layer Security Protocol. An IETF protocol that offers an alternative to SSL encryption technology.

INDEX

A

B

C

D

M

N

O

P

R

S

T

U

V

W