Chapter 29

Voice over IP (VoIP)

Introduction

This chapter explains Voice over IP (VoIP), VoIP protocols, and the benefits and applications of VoIP. This feature is available on routers with PIC bays.

VoIP Overview

Voice over IP provides the ability to make phone calls over IP-based networks. The VoIP PIC can communicate with the following devices:

■ Another terminal on the IP network such as the VoIP PIC.

■ Any LAN SIP endpoint on the IP network, for instance a telephone, or an IP phone directly connected to the IP network.

■ Any LAN H.323 endpoint on the IP network, for instance a telephone, or an IP phone directly connected to the IP network.

■ A PSTN phone or fax.

Figure 29-1 on page 29-4 shows two possible VoIP call scenarios; an IP to IP call, and an IP to PSTN call.

Figure 29-1: VoIP PIC Scenarios
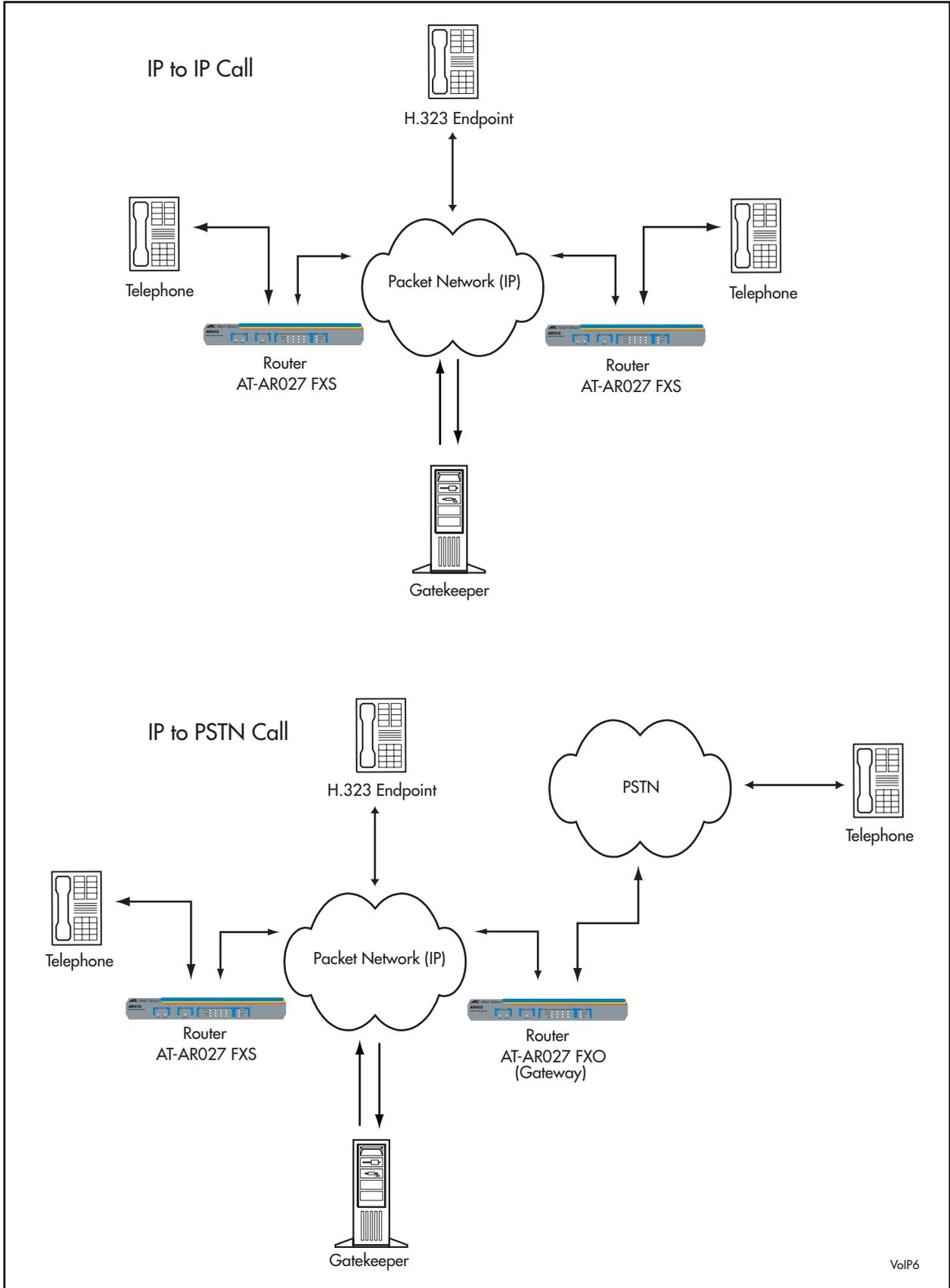
VoIP Benefits and Business Applications

Examples of VoIP benefits are:

■ Cost savings on long distance calls, due to the flat-rate pricing on the Internet. There should not be any additional constraints on the end user, for example, users should not have to use a microphone on a PC.

■ Integration of voice and data networks.

■ Reduction of resource costs. The ability to share equipment and operations across users of data and voice networks may improve network efficiency as excess bandwidth on one network can be used on the other.

■ Removing the need for common infrastructure tools, e.g. physical ports for voice mail services.

■ Open standards that mean businesses and service providers can have equipment from multiple vendors on site.

Examples of typical VoIP business applications are:

■ PSTN gateways.
Connecting the Internet to the PSTN can be provided by a gateway integrated into a PBX, or a separate device, such as a PC-based telephone. The telephone would have access to the public network by calling a gateway at a point close to the destination, which would minimise long distance call charges.

■ Inter-office trunking over the intranet.
Replacement of tie trunks between company-owned PBXs using an Internet link would help to consolidate network facilities.

■ Remote access from a branch or home office.
A small, or home, office could have access to corporate voice, data, and fax services using the company's Intranet.

■ Voice calls from a mobile PC via the Internet.
Calls to the office can be made using a PC that is connected to the Internet. For example, using the Internet to call the office from a hotel instead of using the hotel telephone would reduce long distance call charges.

■ Internet call centre access.
This would allow users enquiring about products being offered on the Internet to access customer service assistants online. It could also interconnect multiple call centres.

■ Internet-aware telephones.
Ordinary telephones can be enhanced to act as Internet access devices as well as providing normal telephony services. For example, accessing Directory Services, asking for a phone number and receiving a voice or text reply.

VoIP FXS Interface Components

A Foreign Exchange Station (FXS) interface connects directly to a standard analog telephone, fax machine or similar device and supplies ring, voltage and dial tone. In the next paragraphs, the main functions and features of the FXS analogue interface are described.

Ring Generation

The ring waveform is the one generated on the FXS port when a call is received and the phone is on-hook. The ring waveform is specific to the country and can be customised by changing the following parameters:

- OnRing time in milliseconds (0-5000)
- OffRing time in milliseconds (0-5000)
- Frequency in Hertz (16-70)

Tone Generation

Tone is the audible sound used to signal to the phone user a specific state. Table 29-1 on page 29-6 lists the tone names and their corresponding meanings.

Table 29-1: Tone Generation

Tone Name	Description
Ring	A number has been dialled and the called party phone is ringing.
Dial	The phone is off-hook and the device is ready to collect digits to make a call.
Busy	The called party is busy.
Disconnect	The device is not able to complete the placed call.

Each tone must be customised for the specific country. Parameters that define tones are:

- On time in milliseconds (0-5000)
- Off time in milliseconds (0-5000)
- Frequency in Hertz (20-1000)

Port Gain

For each FXS port a gain/attenuation can be specified for each direction (receive and transmit). The minimum increment/decrement is 3 dB and the value must be included in the -24 to +24 dB range.

Port Impedance

The FXS port impedance must match the phone impedance to guarantee maximum quality and avoid annoying echo.

Voice Activation and Silence Detection

The Digital Signal Processor (DSP) detects silence and avoids sending packets to the network when the phone user is not talking. This minimises network traffic, but a comfort noise must be generated on the remote end so that the remote party understands the call is ongoing.

Digit Collection

The dialled digits are collected until a configurable timeout occurs or the hash (#) key is pressed.

VoIP Protocols

VoIP uses the following call-control protocol stacks:

- H.323

- Session Initiation Protocol (SIP)

H.323

H.323 protocol specifies the components, protocols, and processes that provide multimedia communication services, real-time audio, video, and data communications over *packet-based networks* including the Internet. H.323 is part of a family of ITU-T recommendations called H.32x that provides multimedia communication services over a variety of networks. Packet-based networks include IP-based (including the Internet) or Internet packet exchange (IPX) based local-area networks (LANs), enterprise networks (ENs), metropolitan-area networks (MANs), and wide area networks (WANs).

H.323 can be applied in a variety of mechanisms, such as audio only (IP telephony), audio and video (video telephony), audio and data, and audio, video and data. H.323 can also be applied to multipoint-multimedia communications. H.323 provides a number of services and, therefore, can be applied in a wide variety of areas including consumer, business, and entertainment applications.

H.323 Components

The H.323 standard specifies the following components:

- Terminals

- Gateways

- Gatekeepers

- Multipoint Control Units

When these components are networked together they provide point-to-point and point-to-multipoint multimedia-communication services. Figure 29-2 on page 29-8 illustrates the H.323 components.

Figure 29-2: H.323 components
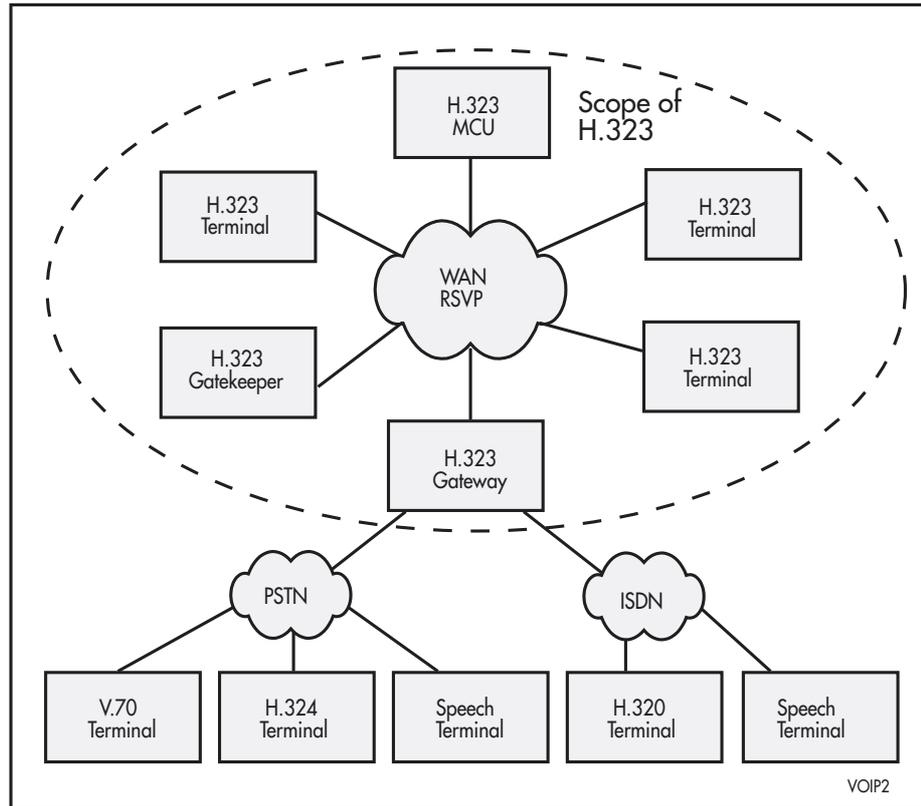
Terminals

An H.323 terminal can be a personal computer (PC) or a standalone device, running an H.323 stack and multimedia communications applications. Terminals support audio communications and can optionally support video or data communications.

H.323 terminals must support the following:

- H.245 for exchanging terminal capabilities and creation of media channels

- H.225 for call signalling and call setup

- RAS for registration and other admission control with a gatekeeper

- RTP/RTCP for sequencing audio and video packets

Gateways

A *gateway* connects two dissimilar networks. An H.323 gateway provides connectivity between an H.323 network and a non-H.323 network. A gateway can connect and provide communication between an H.323 terminal and a *Switched Circuit Network* (SCN). An SCN includes all switched telephony networks, e.g. *public switched telephone network* (PSTN). This connectivity of dissimilar networks is achieved by translating protocols for call setup and release, converting media formats between different networks, and transferring information between networks connected by the gateway. A gateway is not required for communication between two terminals on an H.323 network.

On the H.323 side, a gateway runs H.245 control signalling for exchanging capabilities, H.225 call signalling for call setup and release, and H.225 registration, admissions, and status (RAS) for registration with the *gatekeeper*.

On the SCN side, a gateway runs SCN-specific protocols (e.g. ISDN and SS7 protocols). Terminals communicate with gateways using the H.245 control-signalling protocol and H.225 call-signalling protocol. The gateway translates these protocols in a transparent fashion to the respective counterparts on the non-H.323 network and vice versa. The gateway also performs call setup and clearing on both the H.323-network side and the non-H.323 network side.

A gateway can also perform translation between audio, video, and data formats. Audio and video translation may not be required if both terminal types find a common communications mode. For example, in the case of a gateway to H.320 terminals on the ISDN, both terminal types require G.711 audio and H.261 video, so a common mode always exists. The gateway has the characteristics of both an H.323 terminal on the H.323 network and the other terminal on the non-H.323 network it connects. Gatekeepers are aware of the endpoints that are gateways because this is indicated when the terminals and gateways register with the gatekeeper. A gateway may be able to support several simultaneous calls between the H.323 and non-H.323 networks. A gateway is a logical component of H.323 and can be implemented as part of a gatekeeper or an MCU.

Gatekeepers

The *gatekeeper* is the brain of the H.323 network. It is the focal point for all calls within the H.323 network. Gatekeepers do not have to be present, but if a gatekeeper is present it must perform address translation, admission control, bandwidth control, and zone management. If a gatekeeper is not present, static address translation entries should be configured on the router. Optional functions the gatekeeper can provide include call control signalling, call authorisation, bandwidth management, and call management.

Call monitoring by the gatekeeper provides better control of the calls in the network. Routing calls through gatekeepers provides better performance in the network, as the gatekeeper can make routing decisions based on a variety of factors, for example, load balancing among gateways.

Gatekeeper services are defined by RAS. H.323 networks that do not have gatekeepers may not have these capabilities, but H.323 networks that contain IP telephony gateways should also contain a gatekeeper to translate incoming E.164 telephone addresses into transport addresses. A gatekeeper is a logical component of H.323 but can be implemented as part of a gateway or MCU.

Gatekeeper Discovery

The gatekeeper discovery process is used by endpoints to determine with which gatekeeper to register. It can be a manual or automatic process. Manual discovery configures endpoints with the gatekeeper's IP address, so the endpoints register immediately, but only with the defined gatekeeper. Auto discovery enables an endpoint that may not know its gatekeeper to find who their gatekeeper is by sending a Gatekeeper Request (GRQ) multicast message.

Multipoint Control Units

Multipoint Control Units (MCUs) provide support for conferences of three or more H.323 terminals. All terminals participating in the conference establish a connection with the MCU. The MCU manages conference resources, negotiates between terminals for the purpose of determining the audio or video coder/decoder (CODEC) to use, and may handle the media stream. The multipoint control function can be part of a terminal, gateway, gatekeeper or MCU.

Protocols Specified by H.323

The protocols specified by H.323 are:

- **Audio CODEC**
- **Video CODEC**
- **H.225 Registration, Admission, and Status**
- **H.225 call signalling**
- **H.245 control signalling**
- **Real-Time Transport Protocol**
- **Real-Time Transport Control Protocol**

H.323 terminals must support the G.711 audio CODEC. Optional components in an H.323 terminal are video CODECs, T.120 data-conferencing protocols, and MCU capabilities. H.323 is independent of the packet network and the transport protocols over which it runs.

Audio CODEC

An audio CODEC encodes the audio signal from a microphone for transmission on the transmitting H.323 terminal and decodes the received audio code that is sent to the speaker on the receiving H.323 terminal. Because audio is the minimum service provided by the H.323 standard, all H.323 terminals must have at least one audio CODEC support, as specified in the ITU G.711 recommendation (audio coding at 64 kbps). Additional audio CODEC recommendations such as G.722 (64, 56, and 48 kbps), G.723.1 (5.3 and 6.3 kbps), G.728 (16 kbps), and G.729 (8 kbps) may also be supported.

Video CODEC

A video CODEC encodes video from a camera for transmission on the transmitting H.323 terminal and decodes the received video code that is sent to the video display on the receiving H.323 terminal. Because H.323 specifies support of video as optional, the support of video CODECs is optional as well. However, any H.323 terminal providing video communications must support video encoding and decoding as specified in the ITU H.261 recommendation.

H.225 Registration, Admission, and Status

Registration, admission, and status (RAS) is the protocol used between endpoints (terminals and gateways) and gatekeepers to perform registration, admission control, bandwidth changes, status, and disengage procedures between endpoints and gatekeepers. A *RAS channel* exchanges RAS messages. This signalling channel is opened between an endpoint and a gatekeeper prior to the establishment of any other channels.

H.225 call signalling

H.225 call signalling establishes a connection between two H.323 endpoints. This is achieved by exchanging H.225 protocol messages on the call-signalling channel. The call-signalling channel is opened between two H.323 endpoints or between an endpoint and the gatekeeper.

H.245 control signalling

H.245 control signalling exchanges end-to-end control messages governing the operation of the H.323 endpoint. These control messages carry information related to the following:

- capability exchange

- opening and closing of logical channels used to carry media streams

- flow-control messages

- general commands and indications

shows the relationship between H.323 components.

Figure 29-3: Relationships between H.323 components
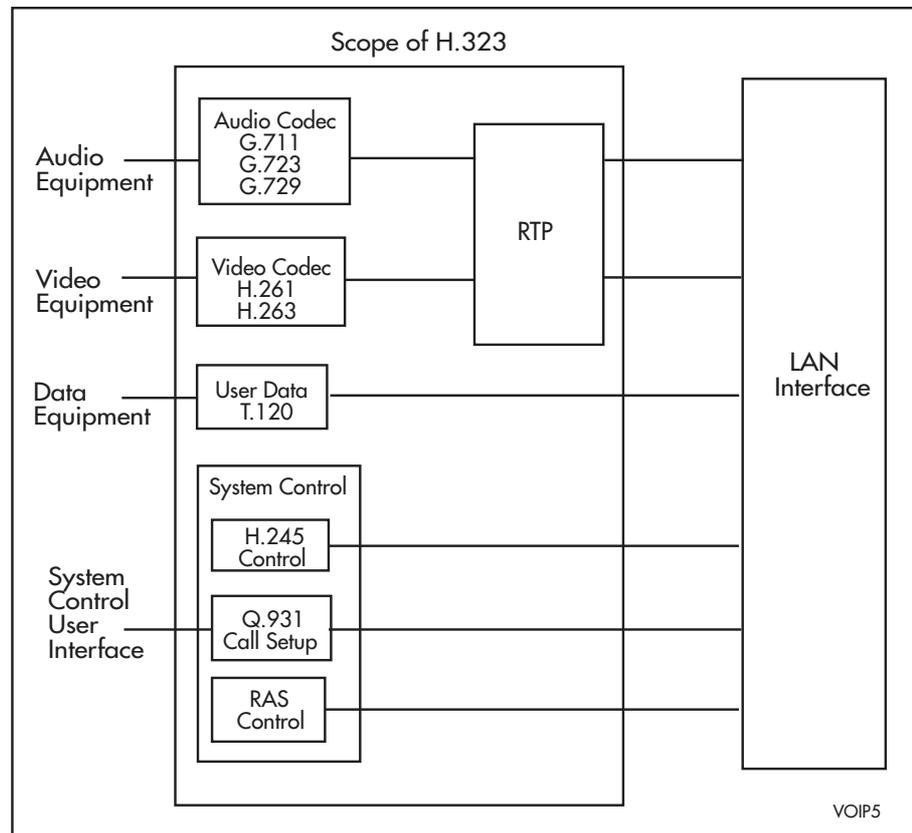
Real-Time Transport Protocol

Real-time Transport Protocol (RTP) provides end-to-end delivery services of delay-sensitive traffic, such as real-time audio and video across packet-based networks. Whereas H.323 transports data over IP-based networks, RTP is typically used to transport data via the user datagram protocol (UDP). RTP, together with UDP, provides transport-protocol functionality. RTP provides sequence numbering information, to determine whether the packets are arriving in the correct order, and time stamping information to determine delivery delays (jitter). RTP can also be used with other transport protocols.

Real-Time Transport Control Protocol

Real-time Transport Control Protocol (RTCP) is the counterpart of RTP that provides control services and real-time conferencing of any size group within the Internet. The primary function of RTCP is to provide feedback on the quality of the data distribution and support for synchronisation of different media streams. Other RTCP functions include ensuring on-time delivery of packets, resource reservation, and reliability.

Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is an application layer protocol that establishes, maintains, and terminates multimedia sessions. These sessions include Internet multimedia conferences, Internet (or any IP Network) telephone calls, and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or via a combination of these. SIP supports session descriptions that allow participants to agree on a set of compatible media types, and supports user mobility by proxying and redirecting requests to the user's current location. SIP is not tied to any particular conference control protocol.

SIP assists in providing advanced telephony services across the Internet. Internet telephony is evolving from its use as a cheap (but low quality) way to make international phone calls to a serious business telephony capability. SIP is one of a group of protocols required to ensure that this evolution can occur.

SIP is part of the IETF standards process and is modelled upon other Internet protocols such as SMTP (Simple Mail Transfer Protocol) and HTTP (Hypertext Transfer Protocol). SIP establishes, changes and *tears down* (ends) calls between one or more users in an IP-based network. In order to provide telephony services, a number of different standards and protocols must come together - specifically to ensure transport (RTP), signalling inter-working with today's telephony network, to be able to guarantee voice quality (RSVP, YESSIR), to be able to provide directories (LDAP), to authenticate users (RADIUS, DIAMETER), and to scale to meet the anticipated growth curves.

Tip: A SIP Application Layer Gateway is available to allow SIP traffic to pass through the firewall. See .

SIP Components

The components within SIP are:

- User Agent
- Network Server

The User Agent is the end system component for the call and the Network Server is the network device that handles the signalling associated with multiple calls.

The User Agent itself has a client element – the User Agent Client (UAC); and a server element – the User Agent Server (UAS). The client element initiates the calls and the server element answers the calls. This allows peer-to-peer calls to be made using a client-server protocol.

The Network Server also provides more than one type of server in the network:

- stateful proxy server
- stateless proxy server
- redirect server

The main function of the SIP servers is to provide name resolution and user location since the caller is unlikely to know the IP address or host name of the called party. SIP addresses users by an email-like address. Each user is identified through a hierarchical URL that is built around elements such as a user's phone number or host name.

An example of a SIP URL is SIP:408562222@171.171.171.1

Because of the similarity, SIP URLs are easy to associate with a user's email address. Using this information, the caller's user agent can identify with a specific server to "resolve the address information". It is likely that this will involve many servers in the network.

A SIP proxy server receives requests, determines where to send these, and passes them onto the next server (using next hop routing principals). There can be many server hops in the network. The difference between a stateful and stateless proxy server is that a stateful proxy server remembers the incoming requests it receives, along with the responses it sends back and the outgoing requests it sends on. A stateless proxy server forgets all information once it has sent on a request. This allows a stateful proxy server to split, or "fork", an incoming call request so that several extensions can be rung at once. The first extension to answer takes the call. This feature is handy if a user is working between two locations (a lab and an office, for example), or where someone is ringing both a boss and their secretary. Stateless Proxy servers are most likely to be the fast, backbone of the SIP infrastructure. Stateful proxy servers are then most likely to be the local devices close to the User Agents, controlling domains of users and becoming the prime platform for the application services.

A redirect server receives requests, but rather than passing these onto the next server it sends a response to the caller indicating the address for the called user. This provides the address for the caller to contact the called party at the next server directly.

SIP is typically used over UDP or TCP.

SIP Functions

SIP provides the following functions:

- **Name translation and user location**
- **Feature negotiation**
- **Call participant management**
- **Call feature changes**
- **Network Address Translation**

Name translation and user location

Name translation and user location ensure that a call reaches the called party wherever they are located, carries out any mapping of descriptive information to location information, and ensures that details of the nature of the call (session) are supported.

Feature negotiation

Feature negotiation allows the group involved in a call (this may be a multi-party call) to agree on the features supported recognising that not all the parties can support the same level of features, (e.g. video may or may not be supported). As any form of MIME type is supported by SIP, there is plenty of scope for negotiation.

Call participant management

Call participant management ensures that during a call, a participant can bring other users onto the call or cancel connections to other users. In addition, users can be transferred or placed on hold.

Call feature changes

Call feature changes ensure that a user can change call characteristics during the course of the call. For example, a call may have been set up as "voice-only", but in the course of the call the users may need to enable a video function. A third party joining a call may require different features to be enabled in order to participate in the call.

Network Address Translation

Network Address Translation (NAT) allows a single device to act as an agent between the Internet (the "public" network) and a local ("private") network. See the Chapter 46, Firewall for more information on NAT. NAT handles the following combination of circumstances:

- the PIC (or the router where it resides) has a private IP address, but is in behind a device that is performing NAT

- the SIP proxy that the PIC has to register with is on the other side of the NAT device

When the PIC registers with the SIP proxy, it sends a packet where it embeds its phone number, IP address, and UDP port number. If the PIC has a private address, it is put into the registration packet. The proxy server registers the PIC's phone number as being at the private address. This private address is not accessible to hosts outside the PIC's own LAN so the registration entry on the SIP proxy server is not very useful. Instead, the registration message must contain the global IP address that is used by the NAT device, and a global port number that the NAT device recognizes so that packets can be routed to the PIC.

Use the **set sip gateway** command on page 29-41 to modify the NAT feature.

SIP provides protocol mechanisms so that end systems and proxy servers can provide the following services:

■ User capability

■ User availability

■ Call set-up

- Call handling

- Call forwarding, including:

  - The equivalent of 700-, 800- and 900- type calls

  - Call-forwarding no answer

  - Call-forwarding busy

  - Call-forwarding unconditional

  - Other address-translation services

- Callee and calling "number delivery", where numbers can be any (preferably unique) naming scheme.

- Personal mobility, i.e. the ability to reach a called party under a single, location-independent address even when the user changes terminals.

- Terminal-type negotiation and selection. A caller can be given a choice on how to reach the party, e.g. via Internet telephony, mobile phone, an answering service, etc.

- Terminal capability negotiation.

- Caller and callee authentication.

- Blind and supervised call transfer. Blind call transfer occurs when the proxy server provides a call transfer feature without any involvement from the endpoint. All signalling messages required are generated by the proxy and are transparent to the Endpoint.

- Invitations to multicast conferences.

SIP Operations

When a user wants to call another user, the caller initiates the call with an invite request. The request contains enough information for the called party to join the session. If the client knows the location of the other party, it can send the request directly to their IP address. If not, the client can send it to a locally configured SIP network server. If this is a proxy server, it tries to resolve the called user's location and send the request to them.

There are many ways it can resolve a location, such as by searching the DNS or accessing databases. Alternatively, the server may be a redirect server that may return the called user location to the calling client for it to try directly. During the course of locating a user, one SIP network server can proxy or redirect the call to additional servers until it arrives at one that definitely knows the IP address where the called user can be found. Once found, the request is sent to the user, and from there several options arise. In the simplest case, the user's telephony client receives the request, that is, the user's phone rings.

If the user takes the call, the client responds to the invitation with the *designated capabilities* of the client software and a connection is established. If the user declines the call, the session can be redirected to a voice mail server or to another user. Designated capabilities refers to the functions that the user wants to invoke. The client software might support video-conferencing, for example, but the user may only want to use audio-conferencing. Regardless, the user can always add functions such as video-conferencing, whiteboarding, or a third user by issuing another invite request to other users on the link.

The following figure illustrates how SIP operates.

Figure 29-4: SIP operation
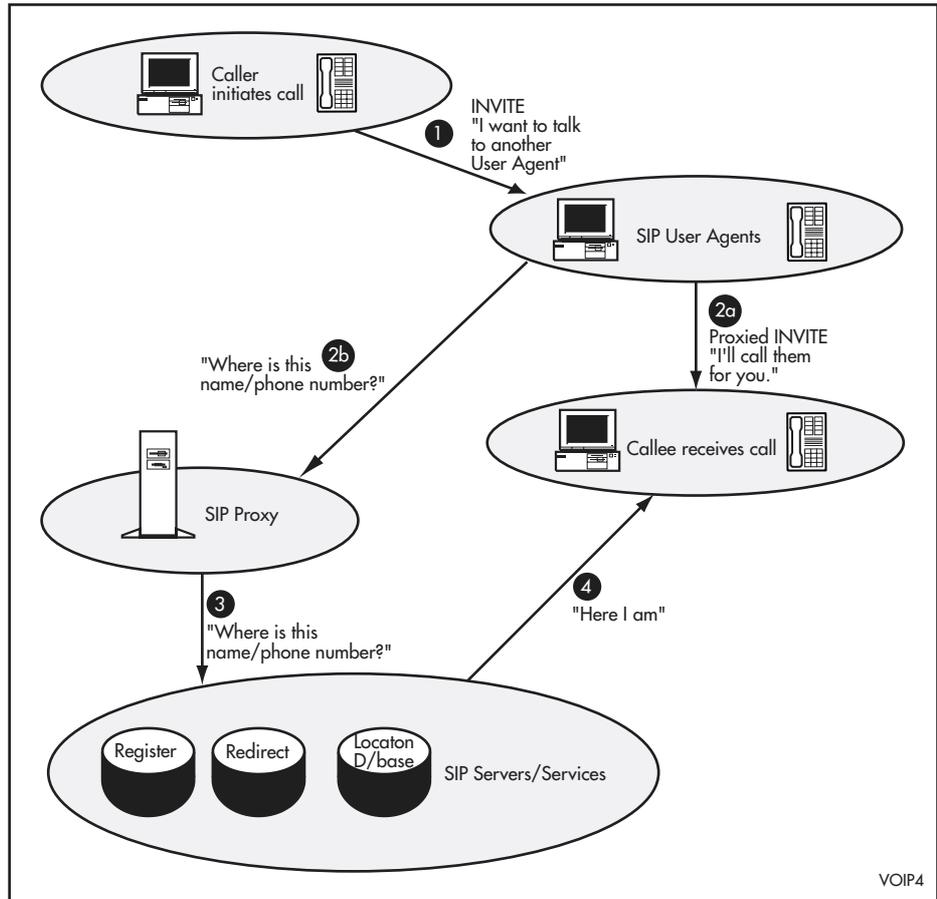
SIP has the unique ability in that it can return different media types. For
example, when a user contacts a company, and the SIP server receives the
client's connection request, it can return to the customer's phone client via a
web Interactive Voice Response (IVR) page (also known as an Interactive Web
Response (IWR) page), with the extensions of the available departments or
users provided on the list. Clicking the appropriate link sends an invitation to
that user to set up a call.

SIP Messages

The types of SIP messages are requests initiated by the client and responses
returned from the server.

A SIP request message consists of the following elements:

- request Line
- header
- message body

A SIP response message consists of the following elements:

- status line
- header
- message body

The request line and header field define the nature of the call in terms of services, addresses, and protocol features. The message body is independent of the SIP protocol and can contain anything.

SIP defines the following *methods* (SIP uses the term "method" to describe the specification areas):

Table 29-2: SIP methods

Method	Description
Invite	Invites a user to join a call
Bye	Terminates the call between two users on a call
Options	Requests information on the capabilities of a server
Ack	Confirms that a client has received a final response to an invite
Register	Provides the map for address resolution, letting a server know the location of other users
Cancel	Ends a pending request but does not end the call
Info	For mid-session signalling

VoIP Engines

Putting voice in packets and handling the VoIP protocols is a specialised and intensive task. To relieve the router CPU of this task, VoIP interfaces are implemented using a semi-autonomous VoIP engine. Each engine supports one or more VoIP interfaces, depending on the hardware configuration. Engines are named fxs$n$, where n is the engine number, and their associated VoIP interfaces are named fxs$n$.0, fxs$n$.1 and so on.

Some VoIP configuration commands relate to an engine and its associated VoIP interfaces as a whole, and others to individual VoIP interfaces. The **show voip instance** command on page 29-65 shows the names of all VoIP interfaces in a router. Engine and interface commands can have abbreviated names (for example, fxs1.0) from the command line. However, configuration scripts should use fully qualified names (for example, bay1.fxs0.0) to avoid configuration problems if a removable engine is taken out.

The VoIP engine executes boot code that is distinct from the router version files. Each time a router is restarted, the boot code must be downloaded by the engine from an external TFTP server. However, an external server is not necessary when the firmware file is stored in flash memory.

Before the engine can download the application code, the boot code must first be downloaded from flash memory. The **set voip bootcode** command on page 29-45 configures the name of the binary file containing the boot code and the location of the application code. The location of the application code can be a TFTP server IP address or in flash. The name of the application code file must be configured using the **set voip file** command on page 29-45.

The engine needs an IP address so it can communicate with the TFTP server. By default this is 192.168.255.n, where n is the number of the engine. The router automatically translates this address to the router's IP address when communicating with the TFTP server. However, a problem arises if the engine's private IP address clashes with one of the router's IP addresses. In this case, the

engine's private IP address can be changed with the **set voip** command on page 29-42. The **set voip public interface** command on page 29-53 indicates the IP address to use when setting up a call or registering with the H.323 gatekeeper or SIP server.

After the router and engine have been configured with the previous commands, use the **enable voip** command on page 29-33 to initiate the firmware download. This happens in two stages: the TFTP client code is first downloaded from the router's flash memory, followed by the protocol code from the TFTP server.

If the TFTP download fails, possibly due to an incorrect filename or the TFTP server being unavailable, then it can be restarted after the problem has been corrected by re-entering the **enable voip protocol** command.

See "Loading VoIP Firmware onto a PIC" on page 29-18 and "Configuration Examples" on page 29-20 for detailed information.

Loading VoIP Firmware onto a PIC

The following instructions are for loading the Voice over IP (VoIP) PIC firmware onto your PIC. They assume you have successfully installed the VoIP PIC in your router and that all LEDs indicate it is on. See the Installation and Safety Guide for your PIC for more information.

1. **Download the VoIP PIC Firmware**

   Open your web browser and enter the URL www.alliedtelesis.co.uk. Navigate to the PIC's product information page and find the firmware files you need from the support page. You will need:

   - the boot code for the PIC
     This code is responsible for loading the protocol image onto the PIC.

   - the SIP protocol image and/or the H.323 protocol image
     The protocol(s) you wish to run on PIC's installed in your router.

   Download and save the firmware files you need to your TFTP server.

2. **Load the boot code onto your router.**

   To load the boot code onto your router, use the command:

   ```
   load method=tftp destination=flash server={hostname|ipadd}
       file=filename
   ```

   For more information about loading files, see Chapter 5, Managing Configuration Files and Software Versions.

3. **If possible, load the protocol image onto your router.**

   If you have enough space in the router's flash, load the protocol image to the flash, so that the router does not need to be continually connected to an external TFTP server. Use the **load** command as described above.

4. **Set the boot code and protocol image on the router.**

   To set the boot code on the router, use the command:

   ```
   set voip bootcode=filename server={ipadd|flash}
   ```

   This command sets the boot code filename, and specifies the location of the protocol image that the boot code will load.

   To set the name of the protocol image, and specify what type of VoIP protocol the protocol image (file) supports, use the command:

   ```
   set voip file=filename protocol={h323|sip} type={fxs|fxo}
   ```

5. **Set the interface for the VoIP traffic.**

   To set the preferred router interface for VoIP traffic, use the command:

   ```
   set voip public interface=interface
   ```

   If a logical interface is not specified, 0 is assumed (that is, eth0 is equivalent to eth0-0).

6. **Load the protocol image onto the PIC or PICs.**

   Use the command:

   ```
   enable voip={h323|sip} [engine={engine}]
   ```

   The boot code loads the protocol onto all PICs unless you specify an individual PIC (engine). Once the firmware is loaded, all the LEDs turn off. The following figure shows an example of the screen output of the firmware download process.

Figure 29-5: Example output of firmware download process

```
Manager> set voip boot=c-1-0-0.bin server=10.32.16.115
Info (1110003): Operation successful.
Manager> set voip fi=hs-1-0-0.bin protocol=h323 type=fxs
Info (1110003): Operation successful.
Manager> set voip public int=eth0
Info (1110003): Operation successful.
Manager> ena voip protocol=h323
Info (1110282): VoIP PIC BAY0:Firmware is loading...
Info (1110282): VoIP PIC BAY1:Firmware is loading...
Manager>
Info (1110293): VoIP PIC BAY0:Firmware successfully loaded.
Manager>
Info (1110293): VoIP PIC BAY0:Firmware is now running.
Manager>
Info (1110293): VoIP PIC BAY1:Firmware successfully loaded.
Manager>
Info (1110293): VoIP PIC BAY1:Firmware is now running.
```

Configuration Examples

The following examples illustrate how to configure Voice over IP on a router.

■   **Using H.323 and no gatekeeper**

■   **Using H.323 and a gatekeeper**

■   **Using a SIP server**

Using H.323 and no gatekeeper

The following example shows how to configure VoIP on the router using H.323 and static entries without a gatekeeper.

Figure 29-6: Configuration of VoIP using H.323 and no gatekeeper
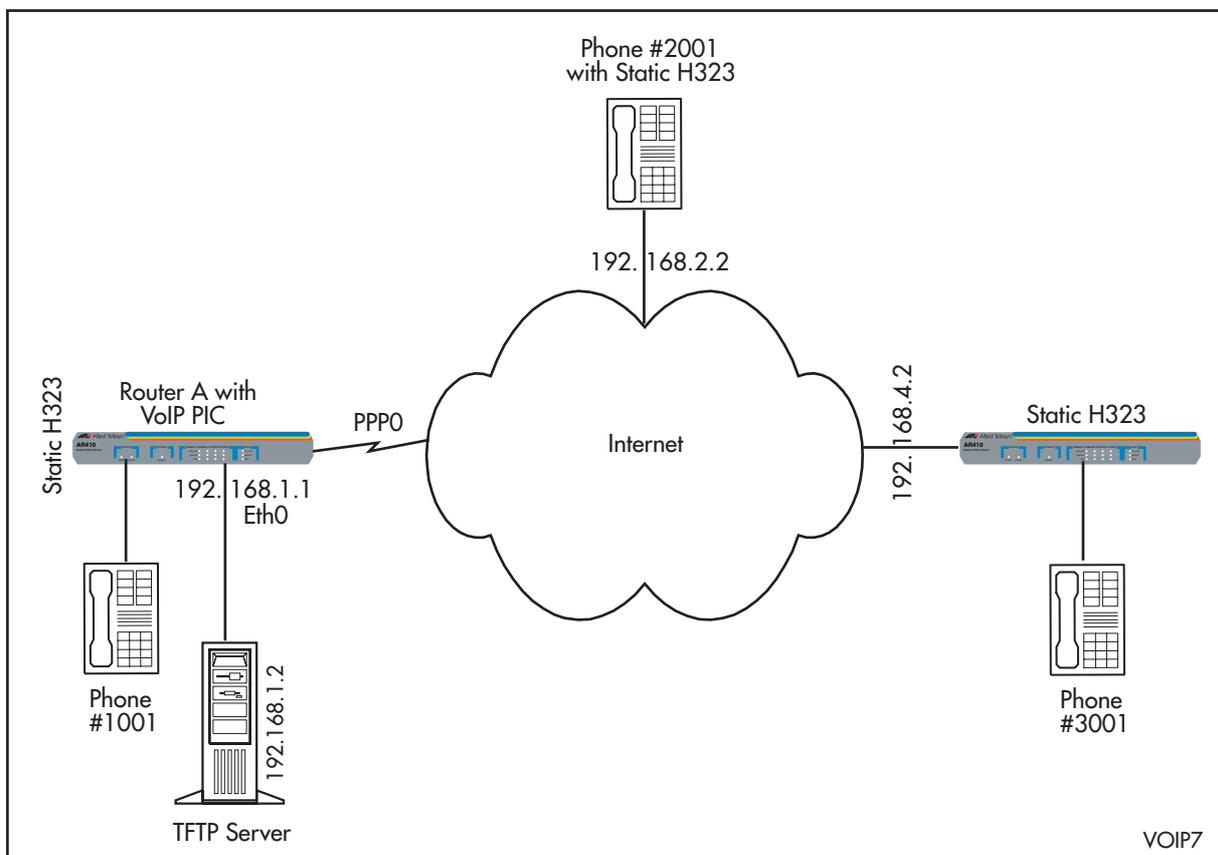
To configure VoIP using Static H.323 and no gatekeeper

Router A Setup

1.   **Set up Router A (with a VoIP PIC installed).**

        set system name=Router_A

2.   **Create a PPP link on Router A.**

        create ppp=0 over=syn0

3.   **Set syn speed (128k is recommended for good voice quality).**

        set syn=syn0 speed=128000

4. **Add IP interfaces to Router A.**

    ```
    enable ip
    add ip int=eth0 ip=192.168.1.1
    add ip int=ppp0 ip=192.168.1.2 mask=255.255.255.252
    add ip rip interface=eth0
    add ip rip interface=ppp0
    ```

5. **Set up and enable VoIP on Router A.**

    ```
    set voip boot=C-1-0-0.bin server=192.168.1.2
    set voip pub int=ppp0
    set voip file=hs-1-0-0.bin protocol=h323 type=fxs
    enable voip protocol=h323 engine=fxs0
    ```

6. **Create the H.323 interface on Router A.**

    ```
    set h323 gateway gatekeeper=none
    create h323 int=fxs0.0 ph=1001 capability=g729a
    ```

7. **Create H.323 Static Entry for phone numbers 2001 and 3001.**

    ```
    create h323 entry engine=fxs0 phone=2001
       hostip=192.168.2.2
    create h323 entry engine=fxs0 phone=3001
       hostip=192.168.4.2
    ```

Using H.323 and a gatekeeper

The following example shows how to configure VoIP on the router using H.323 with a gatekeeper.

Figure 29-7: .Configuration of VoIP using H.323 and a gatekeeper
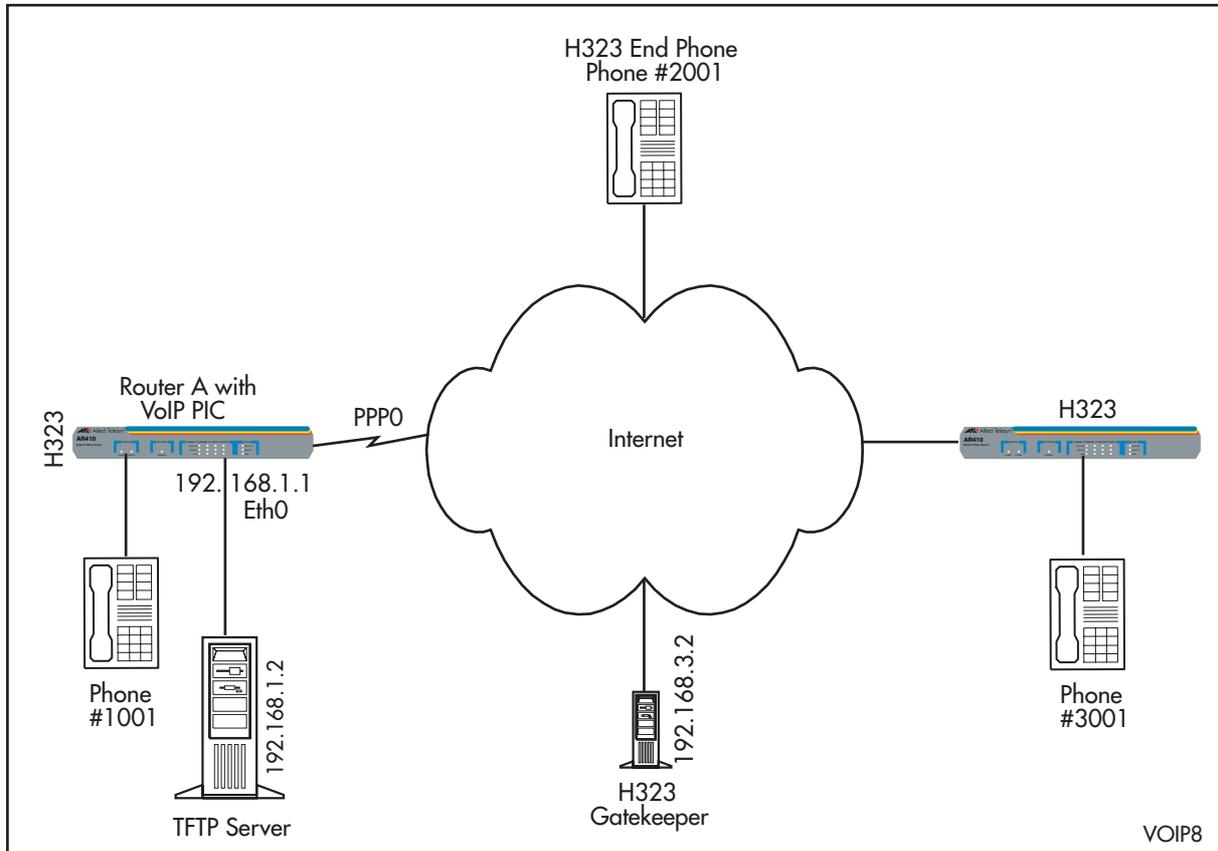
To configure VoIP using Static H.323 and a gatekeeper

Router A Setup

1. **Set up Router A (with a VoIP PIC installed).**

   ```
   set system name=Router_A
   ```

2. **Create a PPP link on Router A.**

   ```
   create ppp=0 over=syn0
   ```

3. **Set syn speed (128k is recommended for good voice quality).**

   ```
   set syn=syn0 speed=128000
   ```

4. **Add IP interfaces to Router A.**

   ```
   enable ip
   add ip int=eth0 ip=192.168.1.1
   add ip int=ppp0 ip=192.168.1.2 mask=255.255.255.252
   add ip rip interface=eth0
   add ip rip interface=ppp0
   ```

5. **Set up and enable VoIP on Router A**

```
set voip boot=C-1-0-0.bin server=192.168.1.2

set voip public interface=ppp0

set voip file=hs-1-0-0.bin protocol=h323 type=fxs

enable voip protocol=h323 engine=fxs0
```

6. **Create the H.323 interface on Router A using a Gatekeeper.**

```
set h323 gateway gatekeeper=192.168.3.2

create h323 int=fxs0.0 ph=1001 capability=g729a
```

Using a SIP server

The following example shows how to configure VoIP on the router using a SIP server.

Figure 29-8: Configuration of VoIP using a SIP server
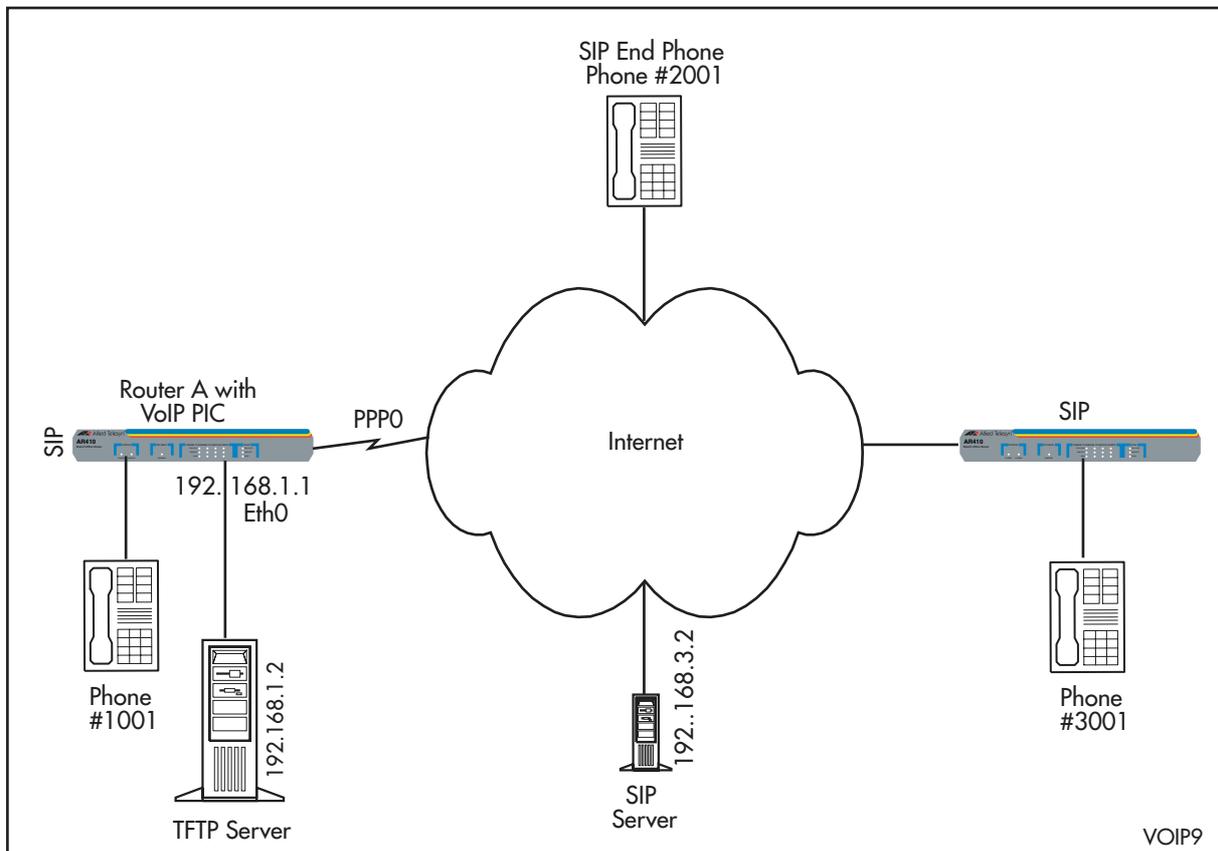
To configure VoIP using a SIP server

Router A setup

1. **Set up Router A (with a VoIP PIC installed).**

```
set system name=Router_A
```

2. **Create a PPP link on Router A.**

```
create ppp=0 over=syn0
```

3. **Set syn speed (128k is recommended for good voice quality)**

   ```
   set syn=syn0 speed=128000
   ```

4. **Add IP interfaces to Router A.**

   ```
   enable ip
   add ip int=eth0 ip=192.168.1.1
   add ip int=ppp0 ip=192.168.1.2 mask=255.255.255.252
   add ip rip interface=eth0
   add ip rip interface=ppp0
   ```

5. **Set up and enable VoIP on Router A**

   ```
   set voip boot=C-1-0-0.bin server=192.168.1.2
   set voip public interface=ppp0
   set voip file=ss-1-0-0.bin protocol=sip type=fxs
   enable voip protocol=sip engine=fxs0
   create sip interface=fxs0.0 phone=1001 domain=192.168.3.2
      proxy=192.168.3.2
   ```

6. **Create the SIP interface on Router A using a SIP server.**

   ```
   set sip interface=fxs0.0 location=192.168.3.2
   set sip interface=fxs0.0 capability=g729a
   ```

Command Reference

This section describes the commands available on the router to configure and manage Voice over IP.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page lxv of About this Software Reference in the front of the software reference manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

create h323

Syntax   CREate H323 INTerface=*interface* PHonenumber=*number*
        [CAPABility={ALL|PCMU|PCMA|G723R53|G723R63|
        G729A}[,...]] [CLIP={ON|OFF}] [DSCP=*dscppriority*]
        [DTMFrelay={H245|RTP|NONE}] [RTCP={ON|OFF}]
        [TOS=*tospriority*]

where:

■ *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay3.fxs0.0).

■ *number* is a phone number, with a maximum of 20 digits.

■ *dscppriority* is a number from 0 to 63.

■ *tospriority* is a number from 0 to 7.

Description   This command creates an H.323 logical interface on a specific physical PIC port. The port registers and uses the gatekeeper specified in the **set h323 gateway** command on page 29-38.

The **interface** parameter specifies the port where H.323 is being created.

The **phonenumber** parameter specifies the local port phone number in e.164 format. This is the only required parameter.

The **capability** parameter specifies a comma-separated list of coding methods. When making or receiving a call, the coding methods are given in the order they are specified in the list. If **all** is specified, the coding methods are given in the following order: PCMU, PCMA, G723R53, G723R63, G729A. The default is PCMU, PCMA.

The **clip** parameter specifies the Calling Line Identification Presentation (Caller ID). If **clip** is on, the port shows its phone number to the called party. If **clip** is off, the phone number is not shown. The default is **on**.

The **dscp** and **tos** parameters specify whether the RTP packets that carry voice frames across the network have a DSCP or TOS value. Increasing either value increases the priority of the RTP packets when they are switched along to their destinations. The default is 0.

The **dtmfrelay** parameter specifies how the DTMF tones are to be carried. If **h245** is specified, coding algorithms such as G.729 and G.723 that are not transparent to DTMF tones, can be carried out of band using an H.245 packet. If **rtp** is specified, packets that carry voice frames across the network have a specific TOS or DSCP value in order to receive a higher priority. The default is **none**.

The **rtcp** parameter specifies whether the real-time control protocol is on or off. If **on** is specified, the protocol is activated with RTP. If **off** is specified, the protocol is not activated. The default is **on**.

Examples   To create an H.323 logical interface on the first VoIP port of PIC 0, with phone number 0055 and preferred coding algorithms G.723R63 and G.729A, use the command:

```
cre H323 int=FXS0.0 ph=0055 capab=G723R63,G729A
```

Related Commands   **destroy h323**
set h323
show h323

create h323 entry

Syntax   CREate H323 ENTry ENGine=*engine* HOSTip=*ipaddr*
              PHonenumber=*number* [POrt=*tcpport*]

where:

- *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay3.fxs0).

- *ipaddr* is an IP address in dotted decimal notation.

- *number* is a phone number, with a maximum of 20 digits.

- *tcpport* is a TCP port number.

Description   This command creates a static entry that can be reached without using a gatekeeper.

The **engine** parameter specifies the name of the VoIP interface where the VoIP protocol is being created.

The **hostip** parameter specifies the IP address of the destination endpoint.

The **phonenumber** parameter specifies the destination phone number in e.164 format.

The **port** parameter specifies the TCP destination port used for Q.931 signalling. The default port is 1720.

Examples   To create a static entry for phone number 12345 that is related to IP address 10.10.1.5, using TCP port number 1720 on FXS engine 2, use the command:

```
cre H323 ent eng=FXS2 ph=12345 host=10.10.1.5 po=1720
```

Related Commands   **destroy h323 entry**
                        show h323 entry

create sip

Syntax   CREate SIP INTerface=*interface* PHonenumber=*number*
    DOmain=*domain* PROXYserver=*ipaddr*[:*udpport*|
    tcpport][;*ipaddr*[:*udpport*|*tcpport*]] [CAPABility={ALL|
    PCMU|PCMA|G723R53|G723R63|G729A}[,...]]
    [DSCP=*dscppriority*] [DTMFrelay={RTP|NONE}]
    [LOCATionserver=*ipaddr*[:*udpport*|
    tcpport][;*ipaddr*[:*udpport*|*tcpport*]]] [PASSword={NONE|
    password}] [RTPport=*udpport*] [TOS=*tospriority*]
    [USERName={NONE|*username*}]

where:

■   *interface* is a port interface name formed by concatenating an interface type
    and an interface instance (for example, fxs0.0). A fully qualified interface
    name may also be specified (for example, nsm0.bay3.fxs0.0).

■   *number* is a phone number, with a maximum of 20 digits.

■   *domain* can either be an IP address in dotted decimal notation or a character
    string 1 to 128 characters long. Valid characters are lowercase letters,
    decimal digits (0–9), and underscore ("_") separated by a dot.

■   *ipaddr* is an IP Address in dotted decimal notation.

■   *udpport* is a UDP port number.

■   *tcpport* is a TCP port number.

■   *dscppriority* is a number from 0 to 63.

■   *password* is a character string 1 to 16 characters long, Valid characters are
    uppercase and lowercase letters, digits, the hyphen, and the underscore.
    The string cannot contain spaces.

■   *tospriority* is a number from 0 to 7.

■   *username* is a character string 1 to 128 characters long. Valid characters are
    any printable character. The string cannot contain any spaces.

Description   This command enables the SIP protocol on a specific physical phone port. The
    port URL is: LOCPHONENUMBER@DOMAIN.

Tip: A SIP Application Layer Gateway is available to allow SIP traffic to pass
through the firewall. See "SIP Application Layer Gateway: VoIP Phone Calls"
on page 46-36 of Chapter 46, Firewall.

The **interface** parameter specifies the port where SIP is being created.

The **phonenumber** parameter specifies the local port phone number in e.164
format. This is the only required parameter.

The **domain** parameter specifies the user network domain name.

The **proxyserver** parameter specifies the server used to send an outgoing call
request. When a call is placed, an invite message is sent to the **proxyserver**. Up
to two proxy servers can be specified, so that if one fails the other can be used.

The **capability** parameter specifies a comma-separated list of coding methods. When making or receiving a call, the coding methods are given in the order they are specified in the list. If **all** is specified, the coding methods are given in the following order: PCMU, PCMA, G723R53, G723R63, G729A. The default is PCMU, PCMA.

The **dscp** and **tos** parameters specify whether the RTP packets that carry voice frames across the network have a DSCP or TOS value. Increasing either value increases the priority of the RTP packets when they are switched along to their destinations. The default is 0.

The **dtmfrelay** parameter specifies how the DTMF tones are to be carried. When using coding algorithms such as G.729 and G.723 that are not transparent to DTMF tones, these can be carried out of band using RTP packets, as described in RFC 2833. The default is **none**.

The **locationserver** parameter specifies the IP address and port of the location server. Up to two location servers can be specified, so that if one fails the other can be used.

The **password** parameter specifies the password the SIP user must supply when using the proxy server's services to authenticate the PIC. The default is **none**.

The **rtpport** parameter specifies the port number used to listen for RTP messages. The port number must be an even number from 5061 to 49151, as odd numbers are reserved for the RTCP protocol. If not set, **rtpport** is assigned dynamically.

The **username** parameter specifies the username the SIP user must supply when using the proxy server's services to authenticate the PIC. The default is **none**.

Examples   To create a SIP logical interface on the first VoIP port of PIC 0, with the phone number 0055, in the alliedtelesis.com domain, using 192.168.0.10 as both location and proxy servers, UDP signalling port 5060, the preferred coding algorithm as G723 and with the username and password for the SIP port set as "*eurord@alliedtelesis.com*" and "*welcome*", use the command:

```
cre sip int=FXS0.0 ph=0055 usern=eurord@alliedtelesis.com
   pass=welcome proxy=192.168.0.10:5060 do=alliedtelesis.com
   locat=192.168.0.10:5060 capab=G723
```

Related Commands   destroy sip
set sip
show sip

destroy h323

Syntax   `DESTroy H323 INTerface=interface`

where *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay3.fxs0.0).

Description   This command destroys a logical interface from the H.323 stack. Any ongoing calls are terminated when this command is executed.

The **interface** parameter specifies the port where H.323 is being destroyed.

Examples   To destroy the H.323 logical interface on the first VoIP port of PIC 0, use the command:

```
dest h323 int=fxs0.0
```

Related Commands   create h323
set h323
show h323

destroy h323 entry

Syntax
```
DESTroy H323 ENTry ENGine=engine PHonenumber=number
     HOSTip=ipaddr [PORT=tcpport]
```

where:

■ *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (e.g. fxs2). A fully qualified engine name may also be specified (e.g. bay0.fxs0 or nsm0.bay1.fxs0).

■ *number* is a phone number, with a maximum of 20 digits.

■ *ipaddr* is an IP address in dotted decimal notation.

■ *tcpport* is a TCP port number.

Description
This command destroys a static entry.

The **engine** parameter specifies the name of the VoIP interface where the VoIP protocol is being destroyed.

The **phonenumber** parameter specifies the destination phone number in e.164 format.

The **hostip** parameter specifies the IP address of the destination endpoint.

The **port** parameter specifies the TCP destination port used for Q.931 signalling. The default port is 1720.

Examples
To destroy a static entry for phone number 12345 that is related to IP address 10.10.1.5, using TCP port number 1720 on FXS engine 2, use the command:

```
dest H323 ent eng=FXS2 ph=12345 host=10.10.1.5 po=1720
```

Related Commands
create h323 entry
show h323 entry

destroy sip

Syntax   DESTroy SIP INTerface=*interface*

where *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay2.fxs0.0).

Description   This command destroys a logical interface from the SIP stack. Any ongoing calls are terminated when this command is executed.

The **interface** parameter specifies the port where SIP is being destroyed.

Examples   To destroy the SIP logical interface on the first VoIP port of PIC 0, use the command:

```
dest sip int=fxs0.0
```

Related Commands   create sip
set sip
show sip

disable voip

Syntax   DISable VOIP PROTocol={H323|SIP} [ENGine=*engine*]

where *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command disables the VoIP engine and reinitiates the master PIC selection process. The VoIP PIC is disabled by default.

The **protocol** parameter specifies the name of the signalling protocol stack that is disabled from the PIC.

The **engine** parameter specifies the VoIP interface being disabled.

Examples   To disable the H.323 protocol on FXS engine 2, use the command:

```
dis voip prot=H323 eng=FXS2
```

Related Commands   enable voip
set voip phone
show voip
show voip load

disable voip debug

Syntax   DISable VOIP DEBug={ALL|IP|H323|SIP|PHONE|RTP|DSP}[,...]
                [ENGine=*engine*]

where *engine* is an engine name formed by concatenating a VoIP interface type
and an engine instance (for example, fxs2). A fully qualified engine name may
also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

■

Description   This command disables debugging on the specified VoIP PIC software module.
A list of options separated by commas may be specified to enable more than
one debugging option at a time.

The **engine** parameter specifies the name of the VoIP interface where
debugging is being disabled. If **engine** is not specified, debugging is disabled
on all VoIP PICs installed on the router.

Example   To disable the debugging of the IP and SIP modules on FXS engine 2, use the
command:

        dis voip deb=ip,sip eng=FXS2

Related Commands   enable voip debug

enable voip

Syntax   ENAble VOIP PROTocol={H323|SIP} [ENGine=*engine*]

where *engine* is an engine name formed by concatenating a VoIP interface type
and an engine instance (for example, fxs2). A fully qualified engine name may
also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command loads the application image associated with the indicated VoIP
protocol to the PIC if it is not already loaded, and enables the VoIP engine. This
command can also be used to resume the firmware download.

The **protocol** parameter specifies the signalling protocol stack that is loaded
into the PIC.

The **engine** parameter specifies the name of the VoIP interface where the VoIP
protocol is enabled. If **engine** is not specified, all VoIP PICs installed on the
router are enabled.

Examples   To load and enable the H.323 protocol on FXS engine 2, use the command:

        ena voip prot=H323 eng=FXS2

Related Commands   disable voip
show voip
show voip load

enable voip debug

Syntax   ENAble VOIP DEBug={ALL|IP|H323|SIP|PHONE|RTP|DSP}[,...]
           [ASYn=*port-number*] [ENGine=*engine*]

where:

■   *port-number* is the number of an asynchronous port.

■   *engine* is an engine name formed by concatenating a VoIP interface type
     and an engine instance (for example, fxs2). A fully qualified engine name
     may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command enables debugging on the specified VoIP PIC software module.
A list of options separated by commas may be specified to enable more than
one debugging option at a time. If **all** is specified, all software modules are
debugged, which may generate enormous amounts of output and lock the
router.

The **asyn** parameter specifies the asynchronous port where the debug output is
to be sent. The port numbers start from 0. Each time this command is entered,
the destination of the debugging output may change.The default is to send
output to the terminal or Telnet session where the command is executed.

The **engine** parameter specifies the name of the VoIP interface where
debugging is being enabled. If **engine** is not specified, debugging is enabled on
all VoIP PICs installed on the router.

Example   To enable H323 module debugging on PIC 1, use the command:

    ena voip deb=H323 eng=FXS1

Related Commands   disable voip debug

reset voip

Syntax   `RESET VOIP TYpe={SW|HW} [ENGine=engine]`

where *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command performs a device reset.

The **type** parameter specifies the requested type of reset, either Hardware (HW) or Software (SW). If **sw** is specified, the router forwards the command to the engine in order to cause a device warm reboot. If **hw** is specified, the router resets the selected VoIP engine and loads the application image to the engine.

The **engine** parameter specifies the name of the VoIP interface to be reset. If **engine** is not present, all VoIP engines installed on the router are reset.

Examples   To perform a software reset of PIC 0, use the command:

```
reset voip ty=sw eng=FXS0
```

Related Commands   **set voip**
show voip

set h323

Syntax

```
SET H323 INTerface=interface [CAPABility={ALL|PCMU|PCMA|
    G723R53|G723R63|G729A}[,...]] [CLIP={ON|OFF}]
    [DSCP=dscppriority] [DTMFrelay={H245|RTP|NONE}]
    [PHonenumber=number] [RTCP={ON|OFF}] [TOS=tospriority]
```

where:

- *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay2.fxs0.0).

- *dscppriority* is a number from 0 to 63.

- *number* is a phone number, with a maximum of 20 digits.

- *tospriority* is a number from 0 to 7.

Description

This command modifies different parameters on any H.323 logical interface already created. The port registers and uses the gatekeeper specified in the SET H323 GATEWAY command.

The **interface** parameter specifies the port where H.323 is being modified.

The **capability** parameter specifies a comma-separated list of coding methods. When making or receiving a call, the coding methods are given in the order they are specified in the list. If **all** is specified, the coding methods are given in the following order: PCMU, PCMA, G723R53, G723R63, G729A. The default is PCMU, PCMA.

The **clip** parameter specifies the Calling Line Identification Presentation (Caller ID). If **clip** is on, the port shows its phone number to the called party. If **clip** is off, the phone number is not shown. The default is **on**.

The **dscp** and **tos** parameters specify whether the RTP packets that carry voice frames across the network have a DSCP or TOS value. Increasing either value increases the priority of the RTP packets when they are switched along to their destinations. The default is 0.

The **dtmfrelay** parameter specifies how the DTMF tones are to be carried. If **h245** is specified, coding algorithms such as G.729 and G.723 that are not transparent to DTMF tones, can be carried out of band using an H.245 packet. If **rtp** is specified, packets that carry voice frames across the network have a specific TOS or DSCP value in order to receive a higher priority. The default is **none**.

The **phonenumber** parameter specifies the local port phone number in e.164 format. This is the only required parameter.

The **rtcp** parameter specifies whether the real-time control protocol is on or off. If **on** is specified, the protocol is activated with RTP. If **off** is specified, the protocol is not activated. The default is **on**.

Examples

To modify a phone number parameter on the H.323 logical interface, on the second VoIP port of PIC 0, use the command:

```
set h323 int=FXS0.1 ph=0088
```

Related Commands

create h323
destroy h323
show h323

set h323 entry

Syntax   `SET H323 ENTry ENGine=engine PHonenumber=number`
              `    [HOSTip=ipaddr] [POrt=tcpport]`

where:

■   *engine* is an engine name formed by concatenating a VoIP interface type
     and an engine instance (for example, fxs2). A fully qualified engine name
     may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

■   *number* is a phone number, with a maximum of 20 digits.

■   *ipaddr* is an IP address in dotted decimal notation.

■   *tcpport* is a TCP port number.

Description   This command changes a static entry that can be reached without using a
                  gatekeeper.

The **engine** parameter specifies the name of the VoIP interface where the VoIP
protocol is being set.

The **phonenumber** parameter specifies the destination phone number in e.164
format.

The **hostip** parameter specifies the IP address of the destination endpoint.

The **port** parameter specifies the TCP destination port used for Q.931
signalling. The default port is 1720.

The **hostip** and **port** parameters are both optional, but at least one of them is
required.

Examples   To set a static entry for phone number 12345 that is related to IP address
                10.10.1.5, using TCP port number 1720 on PIC 0, use the command:

                    `set h323 ent eng=fxs0 ph=12345 host=10.10.1.5 po=1720`

Related Commands   create h323 entry
                        destroy h323 entry
                        set h323 entry

set h323 gateway

Syntax
```
SET H323 GATEway [CONnecttout=time]
    [GATEKeeper={ipaddr[:ipport] [-id][;ipaddr[:ipport][-
    id]]|AUTO|NONE}] [NAMe=alias] [Q931port=tcpport]
    [RASport=udpport] [RESPOnsetout=time] [TIMEtolive=time]
```

where:

■ *time* is a time interval expressed in seconds.

■ *ipaddr* is an IP Address in dotted decimal notation.

■ *ipport* is a TCP/UDP port number.

■ *id* is a string of 20 characters maximum that identify the gateway. Valid characters are uppercase and lowercase letters and digits. The string cannot contain spaces.

■ *alias* is a character string 1 to 40 characters long, in either lower or upper case. Valid characters are uppercase and lowercase letters and digits. The string cannot contain spaces.

■ *tcpport* is a TCP port number.

■ *udpport* is a UDP port number.

Description   This command modifies parameters relating to the H.323 stack configuration common to all ports.

The **connecttout** parameter specifies an interval from 5 to 255 seconds that the terminal waits for the other terminal to answer a call before it treats the connection as being down. The default is 90 seconds.

The **gatekeeper** parameter specifies the IP address and IP port used for the gatekeeper identification, and is used for registration and call management. Up to two gatekeepers can be specified, so that in case of failure the other can be used. If **gatekeeper** is not specified, the auto discovery procedure is used.

The **name** parameter specifies the alias used when registering the PIC with the gatekeeper.

The **q931port** parameter specifies the IP port through which the device listens for Q.931 signalling messages. The default port is 1720.

The **rasport** parameter specifies the IP port through which the device listens for RAS signalling messages. The default port is 1719.

The **responsetout** parameter specifies and interval 5–255 seconds that the terminal waits to receive an Alerting or Call Proceeding message when a call is placed before it treats the connection as being down. The default is 20 seconds.

The **timetolive** parameter specifies an interval 10–10800 seconds between two consecutive registrations. The default is 7200 seconds.

Examples   To register the VoIP FXS engines with alias "NEWGTW10" to gatekeeper 192.168.1.10 that uses RASPORT 1719 and "OpenGK" as the ID, use the command:

```
set h323 gate gatek=192.168.1.10:1719-OpenGK nam=newgtw10
    ras=1719
```

Related Commands   set h323 gateway
show h323 gateway

set sip

Syntax   SET SIP INTerface=*interface* [CAPABility={ALL|PCMU|PCMA|
            G723R53|G723R63|G729A}[,...]] [DOmain=*domain*]
            [DSCP=*dscppriority*] [DTMFrelay={RTP|NONE}]
            [LOCATionserver=*ipaddr*[:*udpport*|*tcpport*]
            [;*ipaddr*[:*udpport*|*tcpport*]]]] [PASSword={NONE|*password*}]
            [PHonenumber=*number*] [PROXYserver=*ipaddr*[:*udpport*|
            tcpport][;*ipaddr*[:*udpport*|*tcpport*]]]]
            [RTPport=*udpport*] [TOS=*tospriority*] [USERName={NONE|
            username}]

where:

■ *interface* is a port interface name formed by concatenating an interface type
   and an interface instance (for example, fxs0.0). A fully qualified interface
   name may also be specified (for example, nsm0.bay2.fxs0.0).

■ *udpport* is a UDP port number.

■ *tcpport* is a TCP port number.

■ *domain* can either be an IP address in dotted decimal notation or a character
   string 1 to 128 characters long. Valid characters are lowercase letters, digits,
   and the underscore separated by a dot.

■ *dscppriority* is a number from 0 to 63.

■ *ipaddr* is an IP Address in dotted decimal notation.

■ *password* is a character string 1 to 16 characters long. It may contain
   uppercase and lowercase letters, digits, the hyphen, and the underscore.
   The string cannot contain spaces.

■ *number* is a phone number, with a maximum of 20 digits.

■ *tospriority* is a number from 0 to 7.

■ *username* is a character string 1 to 128 characters long. Valid characters are
   any printable character. The string cannot contain any spaces.

Description   This command modifies the parameters of any already created SIP logical
interface.

The **interface** parameter specifies the port where SIP is being modified.

The **capability** parameter specifies a comma-separated list of coding methods.
When making or receiving a call, the coding methods are given in the order
they are specified in the list. If **all** is specified, the coding methods are given in
the following order: PCMU, PCMA, G723R53, G723R63, G729A. The default is
PCMU, PCMA.

The **domain** parameter specifies the user network domain name.

The **dscp** and **tos** parameters specify whether the RTP packets that carry voice
frames across the network have a DSCP or TOS value. Increasing either value
increases the priority of the RTP packets when they are switched along to their
destinations. The default is 0.

The **dtmfrelay** parameter specifies how the DTMF tones are to be carried.
When using coding algorithms such as G.729 and G.723 that are not
transparent to DTMF tones, these can be carried out of band using RTP
packets, as described in RFC 2833. The default is **none**.

The **locationserver** parameter specifies the IP address and port of the location server. Up to two location servers can be specified, so that if one fails the other can be used.

The **password** parameter specifies the password the SIP user must supply when using the proxy server's services to authenticate the PIC. The default is **none**.

The **phonenumber** parameter specifies the local port phone number in e.164 format. This is the only required parameter.

The **proxyserver** parameter specifies the server used to send an outgoing call request. When a call is placed, an invite message is sent to the **proxyserver**. Up to two proxy servers can be specified, so that if one fails the other can be used.

The **rtpport** parameter specifies the port number used to listen for RTP messages. The port number must be an even number from 5061 to 49151, as odd numbers are reserved for the RTCP protocol. If not set, **rtpport** is assigned dynamically.

The **username** parameter specifies the username the SIP user must supply when using the proxy server's services to authenticate the PIC. The default is **none**.

Examples   To change a phone number parameter on the SIP logical interface on the second VoIP port of PIC 0, use the command:

```
set sip int=FXS0.1 ph=0088
```

Related Commands   create sip
destroy sip
show sip

set sip gateway

Syntax   `SET SIP GATEway [NATIp=`*`ipaddr`*`] [DEFAUltport={`*`udpport`*`|`
`          `*`tcpport`*`}]`

where:

■ *ipaddr* is an IP Address in dotted decimal notation.

■ *udpport* is a UDP port number.

■ *tcpport* is a TCP port number.

Description   This command modifies the SIP stack configurations common to all VoIP
engines installed on the router. When the **create sip** command on page 29-28
command is used, the gateway parameters on the SIP-created entity are given
default values.

The **natip** parameter specifies the IP address of the NAT device.

The **defaultport** parameter specifies the UDP or TCP port number the PIC is
listening on. The default is 5060.

Examples   To register the VoIP FXS engines with the SIP signalling port 5061, use the
command:

    set sip gate defau=5061

Related Commands   **show sip gateway**

set voip

Syntax   SET VOIP ENGine=*engine* IP=*ipaddr* [GATEway=*ipaddr*]

where:

- *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

- *ipaddr* is an IP address in dotted decimal notation.

Description   This command modifies an IP interface on a specific engine. You only need this command to change the PIC's IP address when there is a conflict between the PIC's IP address and the router's IP address.

The **engine** parameter specifies the name of the VoIP interface where the VoIP protocol is being set.

The **ip** parameter specifies the IP address assigned to the selected PIC. Network Address Translation is applied to the PIC, so packets generated by the PICs have their source IP address replaced by the router's IP address.

The **gateway** parameter specifies the default gateway for the VoIP PIC. Note that this gateway IP address is used only by the PIC to communicate with the router. The gateway must be in the same Class C subnet as the PIC's IP address. By default, the PIC's IP address is 192.168.255.picIndex where *picIndex* is the index of the PIC bay (for example, bay0 PIC index is 1, bay1 PIC index is 2 etc.), and the gateway IP address is 192.168.255.100.

See the **show voip** command on page 29-59 for the PIC IP address settings.

Examples   To set the IP interface with address 192.168.0.10 and mask 255.255.255.0 on PIC 0, use the command:

```
set voip eng=FXS0 ip=192.168.0.10 gate=192.168.0.10
```

Related Commands   **show voip**

set voip ap

Syntax

```
SET VOIP AP INTERFACE=interface [BUFFLEN=blen]
    [BUFFTHR=bthr] [CAPABILITY={ALL|PCMU|PCMA|G723R53|
    G723R63|G729A|T38} [,...]] [COUNTRY={ITALY|JAPAN|
    EUROPE|HOLLAND|AUSTRALIA|NEWZEALAND|USA|CHINA|KOREA}]
    [CRITICALDIGITTIME=msec] [IMPEDANCE={600R|600C|900C|
    CPLX1|CPLX2}] [INTERDIGITTIME=msec] [LEC=lecframe]
    [OFFHOOKTIME=msec] [ONHOOKTIME=msec] [RTPPORT=udpport]
    [RXGAIN=gain] [TXGAIN=gain]
```

where:

■ *interface* is an interface name formed by concatenating an interface type and an interface instance (for example, fxs0). A fully qualified interface name may also be specified.

■ *blen* is a decimal number from 30 to 500.

■ *bthr* is a decimal number from 0 to *blen*.

■ *msec* is a time interval in milliseconds.

■ *lecframe* is a decimal number from 1 to 64.

■ *udpport* is a UDP port number.

■ *gain* is the Gain/Attenuation from -12 to +12 dB in 3 dB steps.

Description   This command modifies the PIC port configuration.

The **interface** parameter specifies the number of the interface or the interface name where the AP is being modified.

The **bufflen** parameter specifies the total length, between 30 and 500 msec, of the circular buffer between the network and the FXS interface. The default is **120** msec.

The **buffthr** parameter specifies the accumulated lengths of voice frames, between 0 and the value of **bufflen** before the frames are transferred to the FXS interface. The default is **60** msec.

The **capability** parameter specifies a comma-separated list of coding methods. When making or receiving a call, the coding methods are given in the order they are specified in the list. If **all** is specified, the coding methods are given in the following order: **pcmu**, **pcma**, **g723r53**, **g723r63**, **g729a**, **t38**. The default is **pcmu**, **pcma**.

The **country** parameter specifies the National Signalling Protocol setting for any event validation characteristic, e.g. ringing frequency and cadence, tone frequency and cadence etc. Available values are **italy**, **japan**, **europe**, **holland**, **australia**, **newzealand**, **usa**, **china** and **korea**. The default is configured by the router at the start-up of the VOIP engine.

The **criticaldigittime** parameter specifies the maximum allowed time between the on-hook event and the first digit entry, between 3000 and 16000 milliseconds. Setting the value to 0 resets the limit. The default is **16000** msec.

The **impedance** parameter specifies the FXS equivalent circuit that should match the connected phone circuit to guarantee the maximum quality and lowest line echo. The default is **600r**.

The **interdigittime** parameter specifies the maximum allowed time between digit entries between 3000 and 4000 milliseconds. Setting the value to 0 resets the limit. The default is **4000** msec.

The **lec** parameter specifies the line echo cancellation, specified as the number of frames. Because each frame takes 125 μSec. and 64 frames are the upper limit, the maximum echo cancellation is 8 milliseconds.

The **offhooktime** parameter specifies the validation time for the off-hook event, between 200 and 250 milliseconds. Setting the value to 0 resets the limit. The default is **250** msec.

The **onhooktime** parameter specifies the validation time for the on-hook event, between 200 and 350 milliseconds. Setting the value to 0 resets the limit. The default is **350** msec.

The **rtpport** parameter specifies the port number used to listen for RTP messages. The port number must be an even number from 5061 to 49151, odd numbers are reserved for the RTCP protocol. If not set, the **rtpport** is assigned dynamically.

The **rxgain** and **txgain** parameters specify the gain applied to the audio signal from and to the network respectively. The default is **0** dB.

Examples   To change the inter-digit time on the first VoIP port of PIC 0 to 3000 msec, use the command:

```
set voip ap interface=fxs0 interdigittime=3000
```

Related Commands   **show voip ap**

set voip bootcode

Syntax   SET VOIP BOotcode=*filename* SERVER={*ipadd*|FLASH}

where:

■ *filename* is a file name in the format *filename.bin*. Valid characters are lowercase letters, digits, and the hyphen.

■ *ipadd* is an IPv4 address in dotted decimal format.

Description   This command sets the filename of the boot code and the IP address of the TFTP server to download the protocol image to.

The **bootcode** parameter specifies the filename of the boot code. The boot code may be stored on the TFTP server or in the router's flash memory.

The **server** parameter specifies the IP address of the TFTP server that stores the application code. If the application code is stored in the router's flash memory, specify **server=flash**.

Examples   To set the filename of the boot code, use the command:

```
set voip bo=C-1-1-1.bin server=202.36.163.22
```

To set the filename of the boot code and download the application code from flash memory, use the command:

```
set voip bo=C-1-1-1.bin server=flash
```

Related Commands   set voip file

set voip file

Syntax   SET VOIP FIle=*filename* PROTocol={H323|SIP}
                   TYpe={FXS|FXO}

where *filename* is a file name in the format *filename.bin*. Valid characters are lowercase letters, digits, and the hyphen.

Description   This command sets the filename of the application code for a selected protocol.

The **file** parameter specifies the application filename for a selected protocol. The filename is stored on the TFTP server or in flash memory.

The **protocol** parameter specifies the signalling protocol stack.

The **type** parameter specifies the VoIP PIC onto which the protocol is loaded.

Examples   To set the application filename for the H323 protocol and load the file onto the FXS PIC, use the command:

```
set voip fi=hs-1-0-1.bin prot=H323 ty=fxs
```

Related Commands   set voip bootcode

set voip phone

Syntax
```
SET VOIP PHone INTerface=interface [[BUFFLen=blen]
    [BUFFThr=bthr] [COUNTRYname={AUSTria|AUStralia|CHIna|
    FRance|GERMANY1|GERMANY2|HOLland|ITALy|JAPan|KORea|
    NEWZealand|SPain|UK|USA1|USA2|}] [CADence={RING|TRING|
    TDIAL|TBUSY|TDISC|TWAIT} [CFreq=frequency-value]
    CValue={cadence-values}|[,...]] [DIGITTout=dtout]
    [FValue=frequency-value] [IMPEDance={600R|600C1|600C2|
    900R|900C1|900C2|900C3|CPLX1|CPLX2|CPLX3|CPLX4|CPLX5|
    CPLX6|CPLX7|CPLX8|GLOBAlcplx}] [LEC=lecframe]
    [RXgain=gain] [TXgain=gain] [VAD={ON|OFF}]
```

where:

■ *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay2.fxs0.0).

■ *blen* is a decimal number from 30 to 500.

■ *bthr* is a decimal number from 0 to *blen*.

■ *cadence-values* is a comma separated list of up to 8 decimal numbers, each from 0 to 5000 milliseconds.

■ *dtout* is the digit collection timeout period from 1 to 255 seconds.

■ *frequency-value* is a comma separated list of up to 2 decimal numbers, each from 17 to 1000 Hz.

■ *lecframe* is a decimal number from 1 to 64.

■ *gain* is the Gain/Attenuation from -12 to +12 dB in 3 dB steps.

Description This command sets different parameters for FXS phone port configuration.

The **interface** parameter specifies the port where the phone is being configured.

The **bufflen** parameter specifies the total length, between 30 and 500 msec, of the circular buffer between the network and the FXS interface. The default is 120 msec.

The **buffthr** parameter specifies the accumulated lengths of voice frames between 0 and the value of **bufflen** before the frames are transferred to the FXS interface. The default is 0 msec.

The **countryname** parameter specifies the National Signalling Protocol setting for event validation characteristics, ringing threshold, tone detection, impedance, etc. The default is configured by the router when the VoIP engine starts up.

Specific values for National Signalling Protocol settings for each country are in tables from

Table 29-3: Australia Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	0.4 - 0.2 - 0.4 - 2.0
Dial Tone	425 Hz	Continuous
Busy Tone	400 Hz	0.375 - 0.375
Ringing Back Tone	400 Hz	0.4 - 0.2 - 0.4 - 2.0
Disc Tone	400 Hz	0.375 - 0.375
Wait Tone	400 Hz	0.375 - 0.375
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-4: Austria Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	50 Hz	1.0 - 5.0
Dial Tone	420 Hz	Continuous
Busy Tone	420 Hz	0.4 - 0.4
Ringing Back Tone	420 Hz	1.0 - 5.0
Disc Tone	420 Hz	0.4 - 0.4
Wait Tone	420 Hz	0.4 - 0.4
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-5: China Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	20 Hz	1.0 - 4.0
Dial Tone	350 + 440 Hz	Continuous
Busy Tone	450HZ	0.35 - 0.35
Ringing Back Tone	450 Hz	1.0 - 4.0
Disc Tone	450 Hz	0.35 - 0.35
Wait Tone	450 Hz	0.35 - 0.35
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	0 dB	

Table 29-6: France Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	50 Hz	1.5 - 3.5
Dial Tone	440 Hz	Continuous
Busy Tone	440 Hz	0.4 - 0.4
Ringing Back Tone	440 Hz	1.5 - 3.5
Disc Tone	440 Hz	0.4 - 0.4
Wait Tone	440 Hz	0.4 - 0.4
Impedance	600Ω	
Tx Gain	-2 dB	
Rx Gain	-9 dB	

Table 29-7: Germany1 Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	0.25 - 4.0 - 1.0 - 4.0
Dial Tone	425 Hz	Continuous
Busy Tone	425 Hz	0.48 - 0.48
Ringing Back Tone	425 Hz	0.25 - 4.0 - 1.0 - 4.0
Disc Tone	425 Hz	0.48 - 0.48
Wait Tone	425 Hz	0.48 - 0.48
Impedance	220Ω + 820Ω // 115 nF	
Tx Gain	+3 dB	
Rx Gain	-10 dB	

Table 29-8: Germany2 Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	0.5 - 4.0 - 1.0 - 4.0
Dial Tone	425 Hz	Continuous
Busy Tone	425 Hz	0.15 - 0.475
Ringing Back Tone	425 Hz	0.5 - 4.0 - 1.0 - 4.0
Disc Tone	425 Hz	0.15 - 0.475
Wait Tone	425 Hz	0.15 - 0.475
Impedance	220Ω+ 820Ω // 115 nF	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-9: Holland Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	1.0 - 4.0
Dial Tone	425 Hz	Continuous
Busy Tone	425 Hz	0.5 - 0.5
Ringing Back Tone	425 Hz	1.0 - 4.0
Disc Tone	425 Hz	0.5 - 0.5
Wait Tone	425 Hz	0.5 - 0.5
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-10: Italy Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	1.0 - 4.0
Dial Tone	425 Hz	0.2 - 0.2 - 0.6 - 1.0
Busy Tone	425 Hz	0.5 - 0.5
Ringing Back Tone	425 Hz	1.0 - 4.0
Disc Tone	425 Hz	0.5 - 0.5
Wait Tone	425 Hz	0.5 - 0.5
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-11: Japan Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	20 Hz	1.0 - 2.0
Dial Tone	400 Hz	Continuous
Busy Tone	400 Hz	0.5 - 0.5
Ringing Back Tone	400 Hz	1.0 - 2.0
Disc Tone	400 Hz	0.5 - 0.5
Wait Tone	400 Hz	0.5 - 0.5
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-9 dB	

Table 29-12: Korea Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	20 Hz	1.0 - 2.0
Dial Tone	350 + 440 Hz	Continuous
Busy Tone	480 + 620 Hz	0.5 - 0.5
Ringing Back Tone	440 + 480 Hz	1.0 - 2.0
Disc Tone	480 + 620 Hz	0.5 - 0.5
Wait Tone	480 + 620 Hz	0.5 - 0.5
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-9 dB	

Table 29-13: New Zealand Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	0.4 - 0.2 - 0.4 - 2.0
Dial Tone	400 Hz	Continuous
Busy Tone	400 Hz	0.5 - 0.5
Ringing Back Tone	400 + 450 Hz	0.4 - 0.2 - 0.4 - 2.0
Disc Tone	400 Hz	0.5 - 0.5
Wait Tone	400 Hz	0.5 - 0.5
Impedance	370Ω + 620Ω // 310 nF	
Tx Gain	+3 dB	
Rx Gain	-9 dB	

Table 29-14: Spain Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	1.5 - 3.0
Dial Tone	425 Hz	Continuous
Busy Tone	425 Hz	0.2 - 0.2
Ringing Back Tone	425 Hz	1.5 - 3.0
Disc Tone	425 Hz	0.2 - 0.2
Wait Tone	425 Hz	0.2 - 0.2
Impedance	600Ω	
Tx Gain	0 dB	
Rx Gain	-7 dB	

Table 29-15: UK Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	25 Hz	0.4 - 0.2 - 0.4 - 2.0
Dial Tone	350 + 440 Hz	Continuous
Busy Tone	400 Hz	0.375 - 0.375
Ringing Back Tone	400 + 450 Hz	0.4 - 0.2 - 0.4 - 2.0
Disc Tone	400 Hz	0.375 - 0.375
Wait Tone	400 Hz	0.375 - 0.375
Impedance	$370\Omega + 620\Omega$ // 310 nF	
Tx Gain	+3 dB	
Rx Gain	-9 dB	

Table 29-16: USA1 Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	20 Hz	2.0 - 4.0
Dial Tone	350 + 440 Hz	Continuous
Busy Tone	480 + 620 Hz	0.5 - 0.5
Ringing Back Tone	440 + 480 Hz	2.0 - 4.0
Disc Tone	480 + 620 Hz	0.5 - 0.5
Wait Tone	480 + 620 Hz	0.5 - 0.5
Impedance	600Ω	
Tx Gain	+3 dB	
Rx Gain	-3 dB	

Table 29-17: USA2 Parameters

Parameter	Value	On - Off Sequence (sec)
Ring Frequency	20 Hz	1.0 - 4.0
Dial Tone	350 + 440 Hz	Continuous
Busy Tone	480 + 620 Hz	0.5 - 0.5
Ringing Back Tone	440 + 480 Hz	1.0 - 4.0
Disc Tone	480 + 620 Hz	0.5 - 0.5
Wait Tone	480 + 620 Hz	0.5 - 0.5
Impedance	$350\Omega + 1000\Omega$ // 210 nF	
Tx Gain	0 dB	
Rx Gain	0 dB	

The **cadence** parameter changes the country-specific value, and specifies the tone cadences that can be changed. A signal or tone cadence can be specified with a series of on and off time intervals. This waveform is then repeated as long as the signal or tone is active. Table 29-18 on page 29-52 shows the **cadence** options that can be changed. If **cadence** is specified, either **cvalue** or **fvalue** or both parameters must also be specified.

Table 29-18: Changeable **cadence** parameter options

Cadence Type	Changes the
RING	Ring Signal cadence when there is an incoming call.
TRING	Ring Tone cadence when the called party phone is ringing.
TDIAL	Dial Tone cadence when the system is ready to collect digits to make a call.
TBUSY	Busy Tone cadence when the called party phone is busy.
TDISC	Disconnect Tone cadence when the called party phone or the VoIP server cannot be reached.
TWAIT	Busy Tone cadence when a call is already in progress and there is a new incoming call.
RINGFREQ	Ring Signal frequency when there is an incoming call.
TRINGFREQ	Ring Tone frequency when the called party phone is ringing.
TDIALFREQ	Dial Tone frequency when the system is ready to collect digits to make a call.
TBUSYFREQ	Busy Tone cadence when the called party phone is busy.
TDISCFREQ	Disconnect Tone frequency when the called party phone or the VoIP server cannot be reached.
TWAITFREQ	Busy Tone cadence when a call is already in progress and there is a new incoming call.
RINGFREQ	Ring Signal frequency when there is an incoming call.
TRINGFREQ	Ring Tone frequency when the called party phone is ringing.

The **cfreq** parameter specifies the frequency of the dial tone for the country specified in **countryname**. The frequency can be a comma-separated list of up to 2 frequency values.

The **cvalue** parameter is required when **cadence** is specified because it defines on/off periods for **cadence** as a comma-separated list of decimal numbers, for example CVALUE=on1,off1... on4,off4.

The **digittout** parameter specifies how long in seconds until digit collection terminates. The timeout period can be skipped by pressing the "#" key. The default is 3 seconds.

The **fvalue** parameter specifies the frequency value, and is required if **cadence** is specified. The default is 0 dB.

The **impedance** parameter specifies the FXS equivalent circuit that should match the connected phone circuit to guarantee the maximum quality and lowest line echo.

The **lec** parameter specifies the line echo cancellation, specified as the number of frames. Because each frame takes 125 μSec. and 64 frames are the upper limit, the maximum echo cancellation is 8 milliseconds. The default is 64 frames.

The **rxgain** and **txgain** parameters specify the gain applied to the audio signal from and to the network respectively.

The **vad** parameter specifies whether the Voice Activity Detection (VAD) feature that detects silent periods is on or off. If on, the PIC does not send voice packets during periods of silence. If off, frames are always sent across the network. The default is **on**.

Examples To set the transmit and receive gain to -3 dB on the first VoIP port of PIC 0, use the command:

```
set voip ph int=FXS0.0 tx=-3 rx=-3
```

Related Commands show voip phone

set voip public interface

Syntax SET VOIP PUBlic INTerface=*interface*

where *interface* is a port interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15 (for example, eth0). If a logical interface is not specified, 0 is assumed (that is, eth0 is equivalent to eth0-0).

Description This command sets the selected router interface as the preferred VoIP interface. This interface sends and receives all VoIP data flows.

The **interface** parameter specifies the name of the logical interface, and implicitly, the attached layer 2 interface. The interface must currently be assigned to the IP module.

Example To set the eth0 interface as the VoIP interface, use the command:

```
set voip pub int=eth0
```

Related Commands enable voip
disable voip
set voip phone
show voip
show voip load

show h323

Syntax   SHow H323 INTerface=*interface*

where *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay2.fxs0.0).

Description   This commands shows the H.323 logical engine configuration on the specified interface.

Figure 29-9: Example output from the **show h323** command

```
H323 Module Information
-------------------------------------------------------------
Port 0
    Phone Number      1000
    Registered        YES
    Reg. Time         Mon 3 Feb 00:00:09 2003
    CLIP              ON
    PRIORITY          TOS - 0
    DTMFRELAY         NONE
    RTCP              ON
    CAPABILITY        PCMU
                      PCMA
                      G723R53
                      G723R63
                      G729A
                      T38

-------------------------------------------------------------
```

Table 29-19: Parameters in the output of the **show h323** command

Parameter	Meaning
Phone Number	The port phone number.
Registered	Whether the port is successfully registered at least to one gatekeeper.
Reg. Time	The date and time that the port was registered, or registration was confirmed, with the gatekeeper.
CLIP	Whether the Calling Line ID Presentation is "ON" or "OFF". If "ON", the port shows its phone number to the called party and the phone number is sent in the CALL SETUP message.
PRIORITY	Whether the RTP/RTCP packets are sent with a TOS or DSCP value across the network.
DTMFRELAY	The coding algorithm used to carry DTMF tones.
RTCP	Whether the RTCP channel is open or not. If "ON", the RTCP channel is open.
CAPABILITY	The list of capabilities used during call setup. The first one has the highest priority.

Examples   To show the H.323 logical interface configuration of PIC 1, use the command:

```
sh h323 int=fxs1.0
```

Related Commands   create h323
destroy h323
set h323

show h323 entry

Syntax   SHow H323 ENTry ENGine=*engine*

where *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command shows the H.323 entries for the requested PIC.

Examples   To show all the defined static entries of PIC 0, use the command:

```
sh h323 ent eng=fxs2
```

Figure 29-10: Example output from the **show h323 entry** command

```
Static phone address Information
-----------------------------------------------------------------------------
Entry No.    Dest. Phonenumber    Dest. IP Address    Dest. Port
1            12345                10.10.1.5           1720
2            55566                10.10.1.8           1720

-----------------------------------------------------------------------------
```

Table 29-20: Parameters in the output of the **show h323 entry** command

Parameter	Meaning
Entry No.	The entry Id.
Dest. Phonenumber	The destination phone number.
Dest. IP Address	The destination host IP address.
Dest. Port	The TCP destination port used for Q.931 signalling.

Related Commands   create h323 entry
destroy h323 entry

show h323 gateway

Syntax   SHow H323 GATEway

Description   This command shows the H.323 Gateway settings for the specified engine.

Figure 29-11: Example output from the **show h323 gateway** command

```
H323 Gateway Information
----------------------------------------------
Gateway
   Name              -
   Gatekeeper        149.35.48.203:1719
   Timetolive        7200
   Response Timeout  20
   Connect Timeout   90
   RAS Port          1719
   Q931 Port         1720
----------------------------------------------
```

Table 29-21: Parameters in the output of the **show h323 gateway** command

Parameter	Meaning
Name	The H.323 alias name used to register to the gatekeeper.
Gatekeeper	The gatekeeper/s where the port is registered.
Timetolive	The interval in seconds between adjacent registrations.
Response Timeout	The interval in seconds that the device waits for an ALERTING message from the called terminal before tearing the call down.
Connect Timeout	The interval in seconds that the device waits for a CONNECT message from the called terminal before tearing the call down.
RAS Port	The port where the device listens for RAS messages.
Q931 Port	The port where the device listens for Q931 messages.

Examples   To show the gateway configuration of the active engines, use the command:

    sh h323 gate

Related Commands   set h323 gateway

show sip

Syntax SHOW SIP INTERFACE=*interface*

where *interface* is a port interface name formed by concatenating an interface type and an interface instance (for example, fxs0.0). A fully qualified interface name may also be specified (for example, nsm0.bay2.fxs0.0).

Description This commands shows the SIP logical interface configuration for the PIC interface specified.

Figure 29-12: Example output from the **show sip** command

```
SIP Module information
-------------------------------------------------------------
Interface 0
    Phone Number      000555
    Authorisation     UserName: "eurord@alliedtelesis.com"
                      Password: "welcome"
    Domain            alliedtelesis.com
    Location Server   192.168.0.5:5060    /UDP
    Proxy Server      192.168.0.5         /UDP
    TOS               0
    Registered        YES
    Capability        PCMU
                      G723R53
                      T38
    RTP port          dynamic assignment
-------------------------------------------------------------
```

Table 29-22: Parameters in the output of the **show sip** command

Parameter	Meaning
Phone Number	The port phone number.
Authorisation	The authorised Username and Password.
Domain	The user's network domain name.
Location Server	The IP address of the server where the port is registered.
Proxy Server	The IP address of the server where the port sends outgoing call requests.
TOS	The TOS value.
Registered	Whether the port is successfully registered to the location servers
Capability	The list of capabilities used during call setup. The first one has the highest priority.
RTP Port	The RTP port number.

Examples To show the first SIP logical interface configuration of PIC0, use the command:

```
show sip interface=fxs0.0
```

Related Commands **create sip**
destroy sip
set sip

show sip gateway

Syntax   SHOW SIP GATEWAY

Description   This command shows the SIP Gateway settings.

Figure 29-13: Example output from the **show sip gateway** command

```
SIP Gateway Information
---------------------------------------------------
Gateway
   Nat IP       None
   Default Port 5060
---------------------------------------------------
```

Table 29-23: Parameters in the output of the **show sip gateway** command

Parameter	Meaning
Nat IP	The IP address of the NAT router.
Default Port	The local UDP/TCP port used for SIP signalling.

Examples   To show the gateway configuration of the active VoIP FXS engines, use the command:

```
show sip gateway
```

Related Commands   **set sip gateway**

show voip

Syntax   SHow VOIP [ENGine=*engine*]

where *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command shows the VoIP PIC configuration and status.

The **engine** parameter specifies the port for which configuration information is required. If **engine** is not specified, all VoIP settings are shown.

Figure 29-14: Example output from the **show voip** command

```
VoIP Module Configuration
-------------------------------------------------------------------------------
Bootcode Filename .... C-1-0-0.bin
H323 FXS Filename ........ HS-1-0-0.bin         H323 FXO Filename ...
SIP  FXS Filename ........ SS-1-0-0.bin         SIP  FXO Filename ...
Public Interface ..... eth0

BAY0.FXS0
   Type           FXS
   Enabled        Yes
   IP             192.168.255.1
   Mask           255.255.255.0
   Gateway        192.168.255.100
   Protocol       H323
   Master         Yes
   Debug          Enabled
      Module      IP
      Module      H323
      Module      PHONE

BAY1.FXS0
   Type           FXS
   Enabled        YES
   IP             192.168.255.2
   Mask           255.255.255.0
   Gateway        192.168.255.100
   Protocol       H323
   Master         No
   Debug          Enabled
      Module      IP

NSM0.BAY2.FXS0
   Type           FXS
   Enabled        No
   IP             192.168.255.5
   Mask           255.255.255.0
   Gateway        192.168.255.100
   Protocol       NONE
   Master         No
   Debug          Disabled
-------------------------------------------------------------------------------
```

Table 29-24: Parameters in the output of the **show voip** command

Parameter	Meaning
Type	Whether the engine type is FXS or FXO.
Enabled	The engine state.
IP	The local IP address given to the selected engine.
Mask	The local network mask address given to the selected engine.
Gateway	The default gateway IP address given to the engine.
Protocol	The engine protocol stack name.
Master	The engine Master selection.
Debug	Whether debugging is enabled or disabled.
Module	The engine software module name, either "IP", "H323", "SIP", "PHONE", "RTP", or "DSCP".

Examples   To show the configuration and status of PIC 1, use the command:

```
sh voip eng=fxs1
```

Related Commands   **disable voip**
enable voip
set voip phone
show voip load

show voip ap

Syntax   SHow VOIP AP INTerface=*interface*

where *interface* is an interface name formed by concatenating an interface type
and an interface instance (for example, fxs0). A fully qualified interface name
may also be specified.

Description   This commands shows the AP configuration for the PIC interface specified.

Figure 29-15: Example output from the **show voip ap** command

```
Analogue Port information
-------------------------------------------------------------
Interface 0
   Interface type:              FXS Simple Loop Call method
   Country:                     Italy
   Capabilities:                G723
                                G729
                                PCMU
                                PCMA
   Critical digit time:         16000 mSec.
   Inter-digit time:            4000 mSec.
   On-hook validation time:     350 mSec.
   Off-hook validation time:    250 mSec.
   Impedance:                   600R
   Tx gain:                     0 dB.
   Rx gain:                     0 dB.
   Max echo cancel. delay:      64 * 125 µSec.
   Packet. buffer length:       120 mSec.
   Packet. preload threshold:   0 mSec.
-------------------------------------------------------------
```

Table 29-25: Parameters in the output of the **show voip ap** command

Parameter	Meaning
Interface type	The functional description of the interface.
Country	The national signalling protocol for the country set.
Capabilities	The specified coding method.
Critical digit time	The time-out time between the on-hook event and the first digit entry.
Inter-digit time	The maximum time-out time between digit entries.
On-hook validation time	The on-hook event validation time.
Off-hook validation time	The off-hook event validation time.
Impedance	The FXS equivalent circuit that should match the connected phone circuit to guarantee the maximum quality and lowest line echo.
Tx gain	The gain applied to the audio signal (from analogue to digital section path).
Rx gain	The gain applied to audio the signal (from digital to analogue section path).

Table 29-25: Parameters in the output of the **show voip ap** command

Parameter	Meaning
Max echo cancel. delay	The maximum echo cancellation delay time.
Packet. buffer length	The packetisation buffer size.
Packet preload threshold	The packetisation preloading threshold time.

Examples   To show the first VoIP port configuration of PIC 0, use the command:

```
sh voip ap int=nsm0.bay3.fxs0
```

Related Commands   **set voip ap**

show voip counter engine

Syntax   SHow VOIP COUnter [ENGine=*engine*]

where *engine* is an engine name formed by concatenating a VoIP interface type and an engine instance (for example, fxs2). A fully qualified engine name may also be specified (for example, bay0.fxs0 or nsm0.bay1.fxs0).

Description   This command displays counters for the specified VOIP interface or if no engine is specified, the counters for all VOIP interfaces on the router are displayed. (Figure 29-16 on page 29-63, Table 29-26 on page 29-64

Figure 29-16: Example output from the **show voip counter engine** command

```
VoIP Module Counters
--------------------------------------------------------------------------------
BAY0
 rxConfigMsg ................. 5

Config Layer Message Counters:
   rxStartTcp ................. 0        txStartTcp ............................. 0
   rxStopTcp .................. 0        txStopTcp .............................. 0
   rxStartUdp ................. 0        txStartUdp ............................. 0
   rxStopUdp .................. 0        txStopUdp .............................. 0
   rxTftpState ................ 0        txTftpDownload ......................... 0
   rxGetConfigParam ........... 0
   rxLogMsgs .................. 3
   rxDebugMsg ................. 0

Config Layer Response Counters
   StartTcpError .............. 0        StopTcpError ........................... 0
   StartUdpError .............. 0        StopUdpError ........................... 0
   Bad Config Msgs ............ 0        Command expires ........................ 0
   Response oK ................ 0        Response Error ......................... 0
   Parameter Not Found ........ 0

Data Message Counters
   Incoming TCP data .......... 0        Outgoing TCP data ...................... 0
   Incoming UDP data ....... 2847        Outgoing UDP data .................. 2848
   txLocalForwardPkt .......... 0
--------------------------------------------------------------------------------
```

Table 29-26: Parameters in the output of the **show voip counter engine** command

Parameter	Meaning
rxConfigMsg	The total number of configuration messages received from the specified VoIP PIC.
rxStartTcp	The total number of 'start listening' TCP requests from the specified VoIP PIC.
rxStopTcp	The total number of 'stop listening' TCP requests from the specified VoIP PIC.
rxStartUdp	The total number of 'start listening' UDP requests from the specified VoIP PIC.
rxStopUdp	The total number of 'stop listening' UDP requests from the specified VoIP PIC.
rxTftpState	The total number of TFTP states received from the specified VOIP PIC.
rxGetConfigParam	The total number of configuration parameter requests received from the specified VoIP PIC.
rxLogMsgs	The total number of log messages received from the specified VoIP PIC.
rxDebugMsg	The total number of log messages received from the specified VoIP PIC.
txStartTcp	The total number of 'start listening' TCP responses sent to the specified VoIP PIC.
txStopTcp	The total number of 'stop listening' TCP responses sent to the specified VoIP PIC.
txStartUdp	The total number of 'start listening' UDP responses sent to the specified VoIP PIC.
txStopUdp	The total number of 'stop listening' UDP responses sent to the specified VoIP PIC.
txTftpDowload	The total number of TFTP download states received from the specified VoIP PIC.
StartTcpError	The total number of 'start listening' TCP request errors from the specified VoIP PIC.
StartUdpError	The total number of 'start listening' UDP request errors from the specified VoIP PIC.
StopTcpError	The total number of 'stop listening' TCP request errors from the specified VoIP PIC.
StopUdpError	The total number of 'stop listening' UDP request errors from the specified VoIP PIC.
Bad Config Msgs	The total number of unknown configuration messages received from the specified VoIP PIC.
Command expires	The total number of commands with no response received from the specified VoIP PIC.
Response OK	The total number of commands with the response 'OK' received from the specified VoIP PIC.
Response Error	The total number of commands with the response 'ERROR' received from the specified VoIP PIC.
Parameter Not Found	The total number of 'parameter not found' messages sent to the specified VoIP PIC.
Incoming TCP data	The total number of incoming TCP packets received by the specified VoIP PIC.
Outgoing TCP data	The total number of outgoing TCP packets sent by the specified VoIP PIC.
Incoming UDP data	The total number of incoming UDP packets received by the specified VoIP PIC.
Outgoing UDP data	The total number of outgoing UDP packets received by the specified VoIP PIC.
txLocalForwardPkt	The total number of packets forwarded to another local PIC from the specified VoIP PIC.

Examples   To display the engine counters for VOIP PIC 0, use the command:

```
sh voip cou eng=fxs0
```

Related Commands   **show voip**

show voip instance

Syntax   SHow VOIP INSTANCE

Description   This command displays the mappings between the engine instances and fully qualified interface names. The user can specify VoIP interfaces using interface instances or fully qualified interface names.

Figure 29-17: Example output from the **show voip instance** command

```
VoIP Engines
Engine                FQN                     Interfaces        FQN
-------------------------------------------------------------------------
fxs0                  bay0.fxs0               fxs0.0            bay0.fxs0.0
                                              fxs0.1            bay0.fxs0.1
fxs1                  bay1.fxs0               fxs1.0            bay1.fxs0.0
                                              fxs1.1            bay1.fxs0.1
fxs2                  nsm0.bay2.fxs0          fxs2.0            nsm0.bay2.fxs0.0
                                              fxs2.1            nsm0.bay2.fxs0.1
-------------------------------------------------------------------------
```

Table 29-27: Parameters in the output of the **show voip instance** command

Parameter	Meaning
Engine	The engine instance name (the VoIP PIC).
FQN	Fully Qualified Name.
Interface	The VoIP interfaces on the PIC card.

show voip load

Syntax   SHow VOIP LOAd [ENGine=*engine*]

where *engine* is an interface name formed by concatenating an interface type and an interface instance (for example, fxs0). A fully qualified interface name may also be specified.

Description   This command shows the VoIP PIC application code download state.

The **engine** parameter specifies the name of the VoIP interface for which the application download state information is required.

Figure 29-18: Example output from the **show voip load** command

```
VoIP TFTP Client Configuration
------------------------------------------------------------
BAY0.FXS0
    Type           FXS
    Revision       1.0
    Version        1-0-0
    Binary Name    HS-1-0-0.bin
    TFTP Server IP 192.168.1.1
    TFTP State     Running
    TFTP Percentage 50%
------------------------------------------------------------
```

Table 29-28: Parameters in the output of the **show voip load** command

Parameter	Meaning
Type	The engine type, either "FXS" or "FXO".
Revision	The engine revision.
Version	The TFTP Client software version.
Binary Name	The application code filename.
TFTP Server IP	The given TFTP server IP address.
TFTP State	The TFTP download state, either "Stopped", "Running", "End", or "Error".
TFTP Percentage	The percentage of application code downloaded.

Examples   To show the application download state of PIC 1, use the command:

```
sh voip loa eng=fxs1
```

Related Commands   disable voip
enable voip
set voip phone
show voip load

show voip phone

Syntax   SHow VOIP PHone INTerface=*interface*

where *interface* is a port interface name formed by concatenating an interface
type and an interface instance (for example, fxs0.0). A fully qualified interface
name may also be specified (for example, nsm0.bay2.fxs0.0).

Description   This commands shows the PIC Phone port configuration for the interface
specified.

Figure 29-19: Example output from the **show voip phone** command

```
FXS Ports Configuration
---------------------------------------------------------------
Phone 0
  -------------------------------------------------------------
  Country                   ITALY
  -------------------------------------------------------------
  Ring     Freq (Hz)  Cadence (msec)
  -------------------------------------------------------------
            25        1000 4000
  -------------------------------------------------------------
  Tone     Freq (Hz)  Cadence (msec)
  -------------------------------------------------------------
  Ring     425        1000 4000
  Dial     425        1000   0
  Busy     425        500   500
  Disc     425        500   500
  Wait     425        500   500
  -------------------------------------------------------------
  Gain
    Tx (dB)                 0
    Rx (dB)                 0
  -------------------------------------------------------------
  Input Buffer
    Length (msec)           120
    Threshold (msec)        0
  -------------------------------------------------------------
  Impedance
    Impedence               600R
  -------------------------------------------------------------
  General
    VAD                     ON
    Digit Tout (sec)        3
    Lec Length (nframe)     64
  -------------------------------------------------------------
```

Table 29-29: Parameters in the output of the **show voip phone** command

Parameter	Meaning
Country	Geographical region where the PIC is located.
Ring	Ring parameters for Ring Cadence and Ring Frequency.
Tone	Tone parameters for Ring, Busy, Dial, Disconnect, and Wait.
Gain	The gain applied to the audio signal. TXGAIN is to the network, RXGAIN is from the network.
Input Buffer	The length of the **bufflen** and **buffthr** input buffers.
Impedance	The resistance required to guarantee maximum voice quality and avoid echo.
VAD	Whether Voice Activation and silence Detection is active.
Digit Tout	The time in seconds before digit collection terminates.
Lec Length	The line echo cancellation length expressed in frames. Each frame is 0.125 μsec.

Examples   To show the first VoIP phone port configuration of PIC 0, use the command:

```
sh voip ph int=FXS0.0
```

Related Commands   **set voip phone**