Chapter 49

Voice over IP (VoIP)

Introduction

This chapter explains Voice over IP (VoIP), VoIP protocols, and the benefits and applications of VoIP.

VoIP Overview

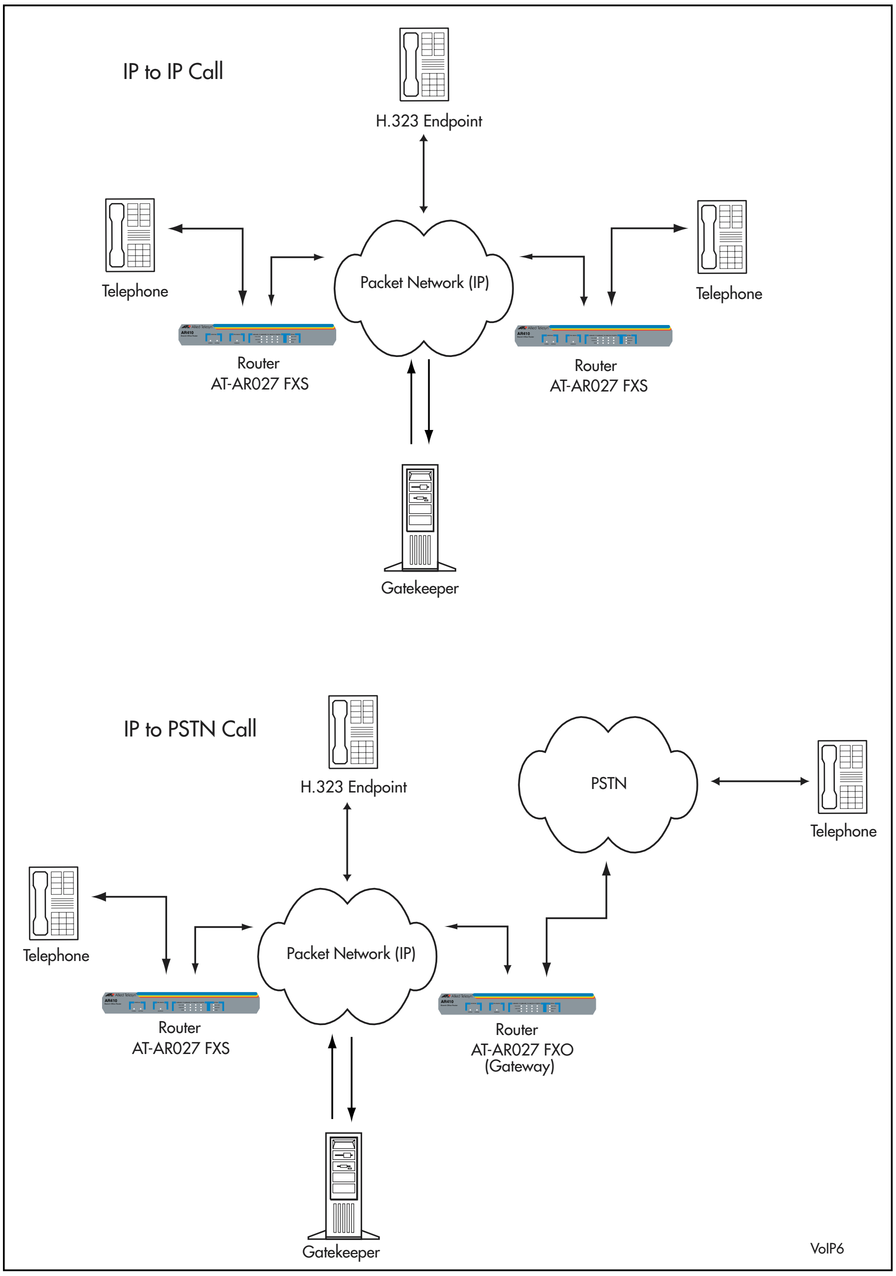
Voice over IP provides the ability to make phone calls over IP-based networks.

The VoIP PIC can communicate with the following devices:

- Another terminal on the IP network such as the VoIP PIC.
- Any LAN 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 49-1 on page 49-4 illustrates two possible VoIP call scenarios; an IP to IP call, and an IP to PSTN call.

Figure 49-1: VoIP PIC Scenarios.



VoIP Benefits and Applications

Benefits

VoIP has many benefits, the main ones are listed below.

- 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.
- Common infrastructure tools are no longer needed, e.g. physical ports for voice mail services.
- There are open standards that means businesses and service providers can have equipment from multiple vendors on site.

Applications

There are many useful VoIP applications, some of which are listed below.

- 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 49-1 on page 49-6 lists the tone names and their corresponding meanings.

Table 49-1: 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 can, and must, be customised for the specific country. The parameters that can be used to define the above-mentioned 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) can detect silence and avoid 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 to make the remote party understand that 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, and the Session Initiation Protocol (SIP). The H.323 protocol is discussed in greater detail below.

H.323

The *H.323 protocol* specifies the components, protocols and procedures 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.

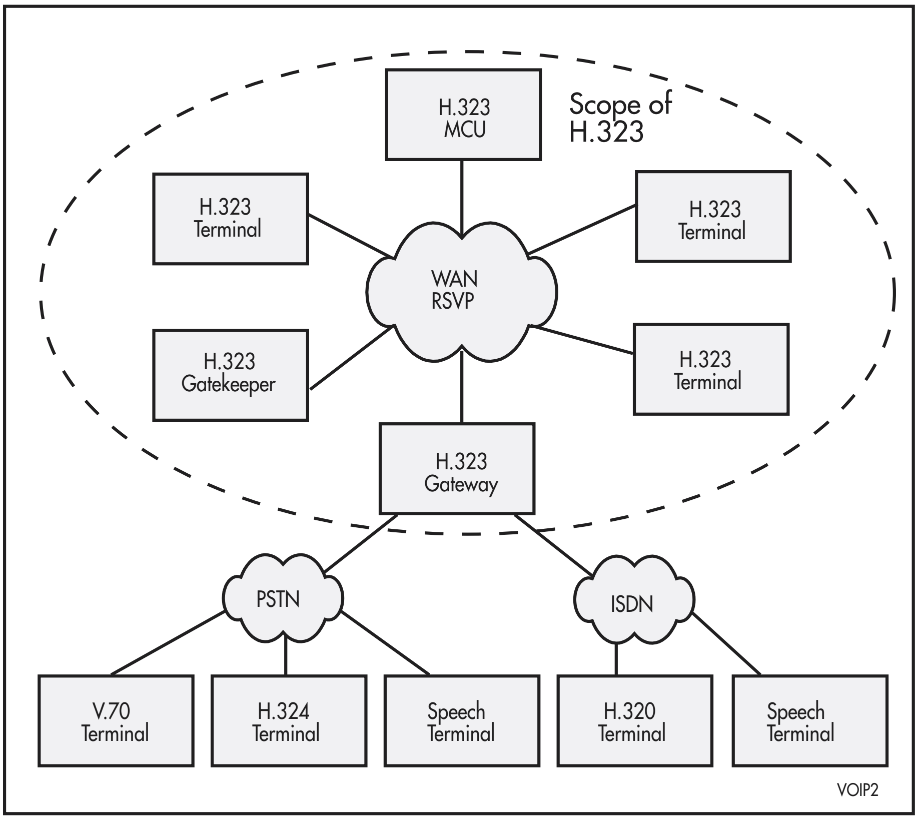
H.323 Components

The H.323 standard specifies four components:

- Terminals.
- Gateways.
- Gatekeepers.
- Multipoint control units (MCUs).

When these components are networked together they provide point-to-point and point-to-multipoint multimedia-communication services. Figure 49-2 on page 49-8 illustrates the H.323 components.

Figure 49-2: H.323 components.



Terminals

An H.323 *terminal* can either be a personal computer (PC) or a stand-alone device, running an H.323 stack and multimedia communications applications. The terminals support audio communications and can optionally support video or data communications.

Terminal Characteristics

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 procedure is used by endpoints to determine with which gatekeeper to register. It can be a manual or automatic procedure. Manual discovery configures endpoints with the gatekeeper’s IP address, so the endpoints can register immediately, but only with the defined gatekeeper. Auto discovery enables an endpoint that may not know its gatekeeper to find out 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 listed below:

- Audio CODECs.
- Video CODECs.
- H.225 registration, admission, and status (RAS).
- H.225 call signalling.
- H.245 control signalling.
- Real-time transfer protocol (RTP).
- Real-time control protocol (RTCP).

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.

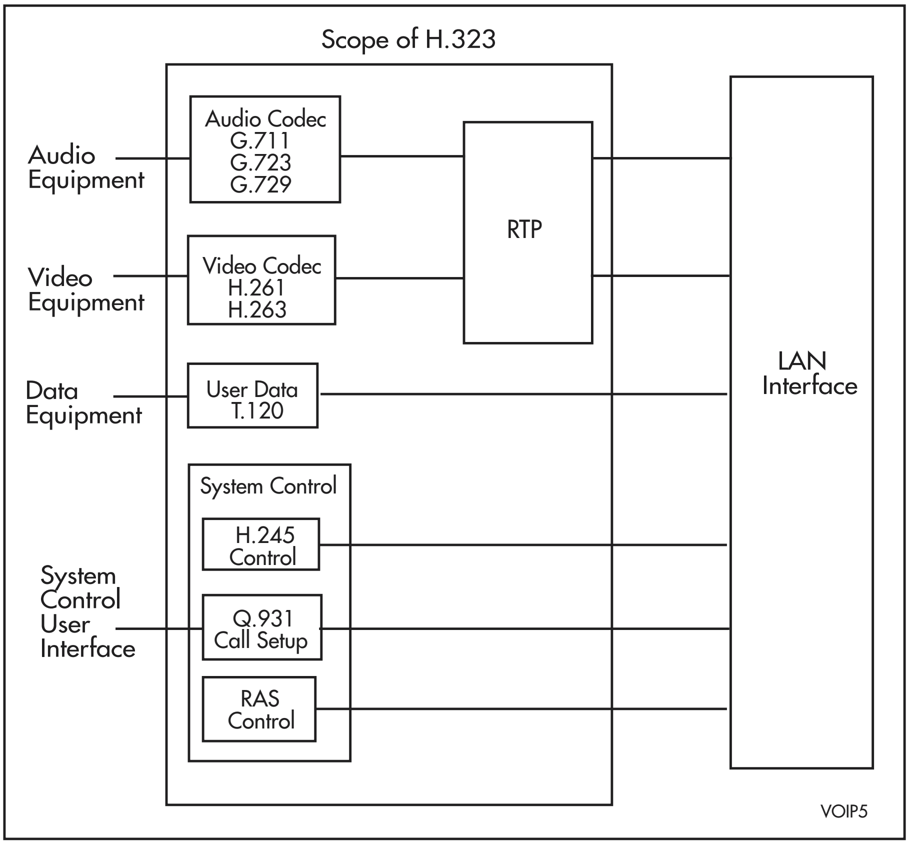
H.245 Control Signalling

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

Figure 49-3 on page 49-12 illustrates the relationship between H.323 components.

Figure 49-3: 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.

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.

SIP Components

There are two components within SIP. The SIP *User Agent* and the SIP *Network Server*. The User Agent is effectively the end system component for the call and the SIP 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), known as client and server respectively. 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 SIP Server element also provides more than one type of server. There are effectively three forms of server that can exist in the network: the *SIP stateful proxy server*, the *SIP stateless proxy server,* and the *SIP 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 Firewall chapter 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 PICs phone number as being at the private address. This private address is not accessible to hosts outside the PICs 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 to modify the NAT feature.

SIP also provides the following 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.

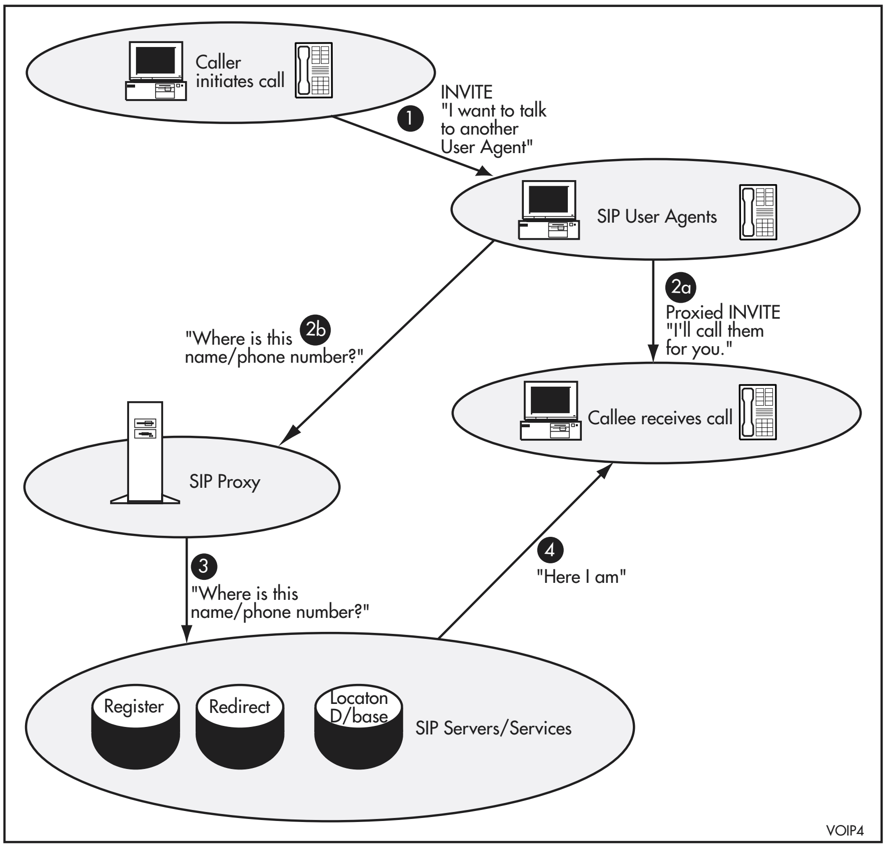
SIP Operation

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 do this, such as 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.

Figure 49-4 on page 49-16 illustrates SIP operation.

Figure 49-4: 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

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

A SIP request message consists of three elements:

- Request Line.
- Header.
- Message Body.

A SIP response message consists of three 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):

- Invite - invites a user to join a call.
- Bye - terminates the call between two of the 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

The packetisation of voice and the handling of the VoIP protocols is a specialised and intensive task. To relieve the router CPU of this onerous task, VoIP interfaces are implemented using a semi-autonomous VoIP *engine*. Each engine supports one or more VoIP interfaces depending upon 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 may be used to see the names of all the VoIP interfaces in a router. From the command line, engine and interface commands may use the abbreviated names (e.g. fxs1.0), but configuration scripts should use fully qualified names (e.g. 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 release files. Each time a router is restarted the boot code must be downloaded by the engine from an external TFTP server.

If the firmware file is stored in the router's flash, then an external TFTP server is not necessary.

Before the engine can download the application code, the boot code must first be downloaded from the router's flash memory. The command SET VOIP BOOTCODE 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 the router's flash. The name of the application code file(s) must be configured using the SET VOIP FILE command.

So that the engine may communicate with the TFTP server, it needs an IP address. 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 may be changed using the SET VOIP command. The SET VOIP PUBLIC command indicates to the engine the router 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 PROTOCOL command to initiate the firmware download. This proceeds 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, say due to an incorrect filename or the unavailability of the TFTP server, then it can be restarted once the problem has been corrected by re-entering the ENABLE VOIP PROTOCOL command.

See "*Downloading VoIP Firmware*" on page 49-19 and "*Configuration Examples*" on page 49-20 for detailed information.

Downloading VoIP Firmware

The following instructions are for downloading the Voice over IP (VoIP) PIC firmware onto your PIC. The instructions assume you have successfully installed the VoIP PIC into your router and made sure all the LEDs show as being on. See your Port Interface Card (PIC) Quick Install Guide for more information.

To download the VoIP PIC firmware, do the following:

Insert the VoIP PIC into the router (as per your PIC Quick Install Guide).

Open the browser of your choice, enter the URL www.alliedtelesyn.com and navigate to *Products*, then to the *Accessories and Other Products* page. Click the *Show Products* button beside Port Interface Cards, and click the *AT-AR027/FXS* link. You are able to download all the files you require from here.

Download and save the firmware file to a location of your choice. Then load the boot code to the router's FLASH. If you have enough space in FLASH, also load the application code to allow downloading without an external TFTP server.

1. Set the boot file on the router, using the command:

```
SET VOIP BOOTCODE=filename SERVER={ipaddr|flash}
```

where:

- *filename* is a file name of the form `filename.bin`. Valid characters are lowercase letters, digits (0-9) and the hyphen (-).
- *ipaddr* is a TFTP server IPv4 address in dotted decimal format. Use the PING command to make sure the IP address is reachable by the router.
- *flash* is the application code already stored in the router's flash.

This file should already be in the router's flash. Set the SERVER parameter to "flash" if you wish to download the application code from flash.

2. Set the protocol image filename in the TFTP server, using the command:

```
SET VOIP FILE=filename PROTOCOL={H323|SIP} TYPE={FXS|FXO}
```

where:

- *filename* is a file name of the form `filename.bin`. Valid characters are lowercase letters, digits (0-9) and the hyphen (-).

3. Set the preferred router interface for the VoIP traffic, using the command:

```
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 in the range 0 to 15 (e.g. eth0). If a logical interface is not specified, 0 is assumed (i.e. eth0 is equivalent to eth0-0).

4. Initiate the download of the H.323 or SIP protocol image, using the command:

ENABLE VOIP PROTOCOL={H323|SIP} [ENGINE={*engine*}]

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).

If the TFTP download fails, say due to an incorrect filename or the unavailability of the TFTP server, then it can be restarted once the problem has been corrected by re-entering the ENABLE VOIP PROTOCOL command.

Once the firmware is downloaded, all the LEDs should turn off. The figure below shows an example of the screen output of the firmware download process.

Figure 49-5: Example output of firmware download process

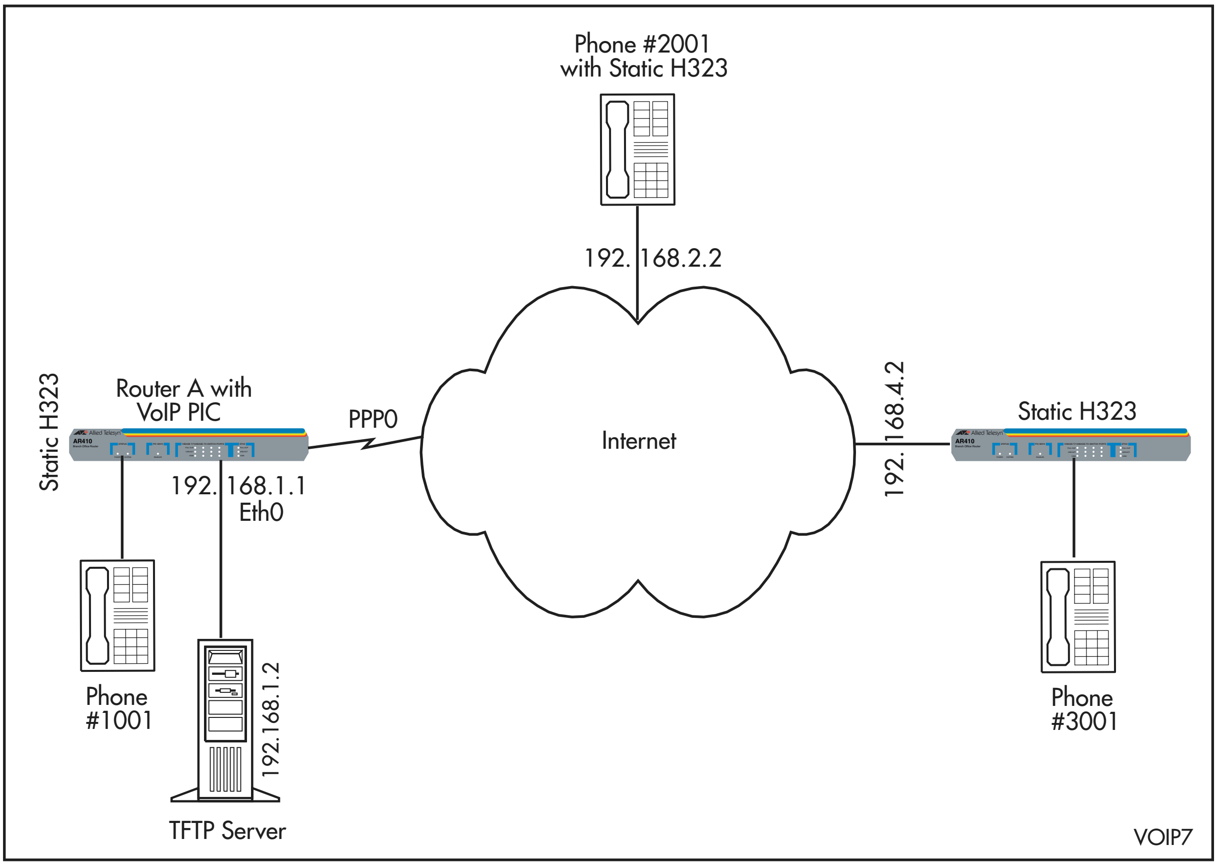
```
Manager> set voip boot=c-1-0-0.bin server=10.32.16.115
Info (1110003): Operation successful.
Manager> set voip fi=hs-1-0-0.bin protocol=h323
Info (1110003): Operation successful.
Manager> set voip public int=eth0
Info (1110003): Operation successful.
Manager> ena voip protocol=h323
Info (1110282): 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

Example One

The following example illustrates the steps required to configure VoIP on the switch, using H.323 and static entries, without a gatekeeper.

Figure 49-6: Configuration of VoIP using H.323 and no gateway.



To configure VoIP using Static H323 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 H323 interface on Router A.**

```
set h323 gateway gatekeeper=none
```

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

7. **Create H323 Static Entry for phone numbers 2001 and 3001.**

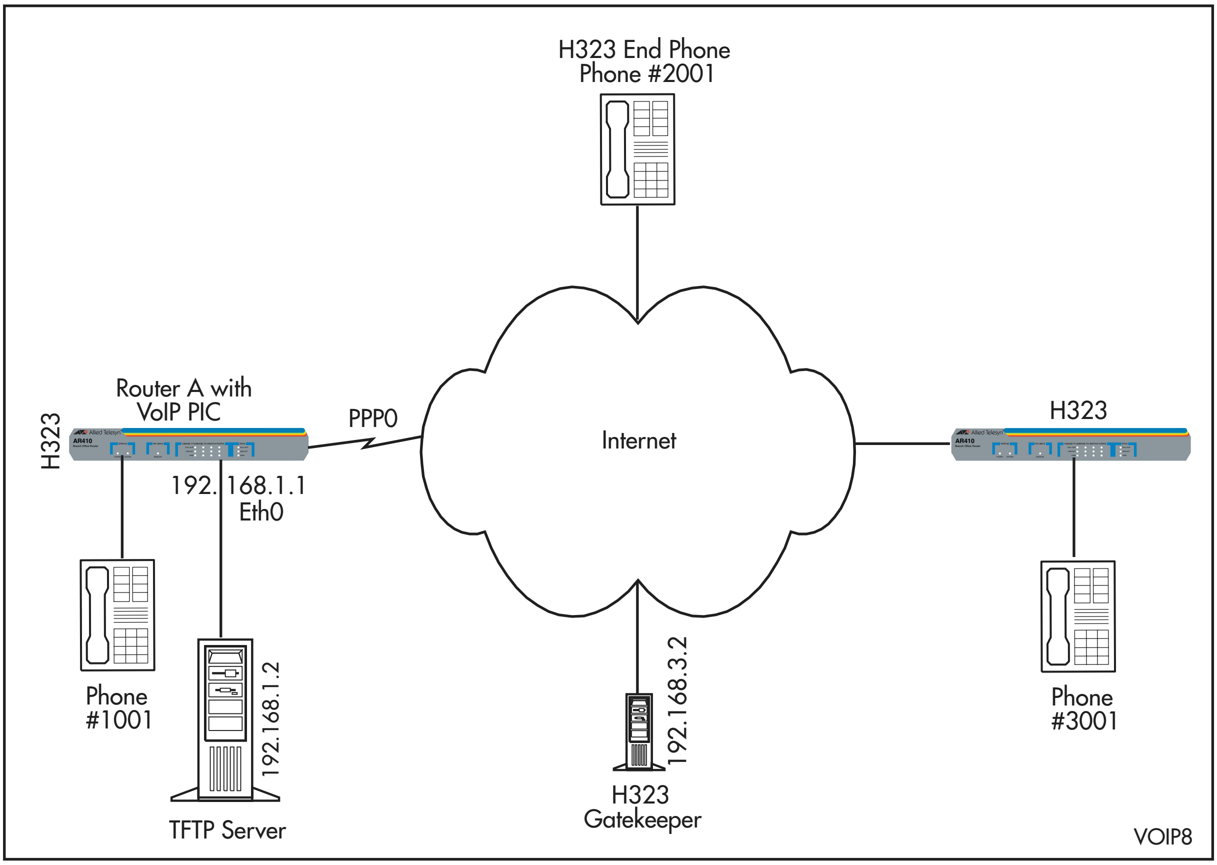
```
create h323 entry engine=fxs0 phone=2001
    hostip=192.168.2.2
```

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

Example Two

The following example illustrates the steps required to configure VoIP on the switch, using H.323 and a gatekeeper.

Figure 49-7: .Configuration of VoIP using H.323 and a gatekeeper.



To configure VoIP using Static H323 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
```

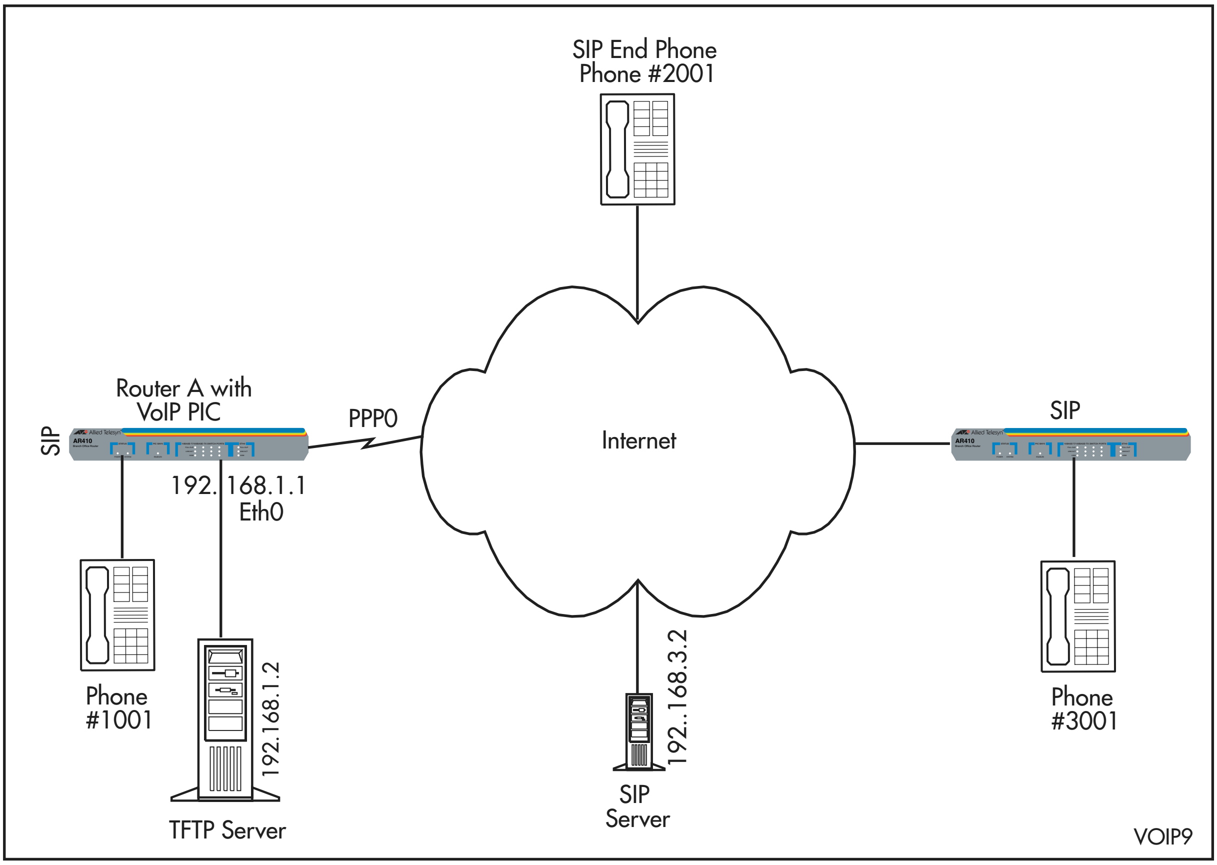
6. **Create the H323 interface on Router A using a Gatekeeper.**

```
set h323 gateway gatekeeper=192.168.3.2
create h323 int=fxs0.0 ph=1001 capability=g729a
```

Example Three

The following example illustrates the steps required to configure VoIP on the switch, using a SIP server.

Figure 49-8: 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 switch to configure and manage Voice over IP.

See *Conventions on page xciv of Preface* 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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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.

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 set to ON, the port shows its phone number to the called party. If CLIP is set to 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 the DSCP or TOS 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 H.245 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 to get a higher priority when switched along the path to their destinations. 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:

```
CREATE H323 INTERFACE=FXS0.0 PHONENUMBER=0055
   CAPABILITY=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 (e.g. fxs2). A fully qualified engine name may also be specified (e.g. 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:

```
CREATE H323 ENTRY ENGINE=FXS2 PHONENUMBER=12345
   HOSTIP=10.10.1.5 PORT=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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 lower case letters (a–z), 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 (0- 9), the hyphen (-), and the underscore character ("_"). 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.

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 the DSCP or TOS 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 in RTP packets, as described in RFC2833. 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 in the range 5061- 49151, as odd numbers are reserved for the RTCP protocol. If not set, the 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 alliedtelesyn.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@alliedtelesyn.com*" and "*welcome*", use the command:

```
CREATE SIP INTERFACE=FXS0.0 PHONENUMBER=0055
   USERNAME=eurord@alliedtelesyn.com PASSWORD=welcome
   PROXYSERVER=192.168.0.10:5060 DOMAIN=alliedtelesyn.com
   LOCATIONSERVER=192.168.0.10:5060 CAPABILITY=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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:

```
DESTROY H323 INTERFACE=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 concatenting 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:

```
DESTROY H323 ENTRY ENGINE=FXS2 PHONENUMBER=12345
   HOSTIP=10.10.1.5 PORT=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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:

```
DESTROY SIP INTERFACE=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 concatenting 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).

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:

```
DISABLE VOIP PROTOCOL=H323 ENGINE=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:

- *port-number* is the number of an asynchronous port.
- *engine* is an engine name formed by concatenting 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).

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 the ENGINE parameter 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:

```
DISABLE VOIP DEBUG=IP,SIP ENGINE=FXS2
```

Related Commands ENABLE VOIP DEBUG

ENABLE VOIP

Syntax `ENABLE VOIP PROTOCOL={H323|SIP} [ENGINE=`*`engine`*`]`

where:

- *engine* is an engine name formed by concatenting 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).

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 the ENGINE parameter 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:

```
ENABLE VOIP PROTOCOL=H323 ENGINE=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 concatenting 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).

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. If IP is specified, all IP interfaces on the PIC are debugged. If H323 is specified, the H323 protocol stack is debugged. If SIP is specified, the SIP protocol stack is debugged. If PHONE is specified, the VoIP engine is debugged. If RTP is specified, the RTP/RTCP protocol stack is debugged. If DSP is specified, the DSP manager is debugged.

Enabling all debug options with ENABLE VOIP FXS DEBUG=ALL may generate enormous amounts of output, causing the router to lock up.

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 was executed.

The ENGINE parameter specifies the name of the VoIP interface where debugging is being enabled. If the ENGINE parameter 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:

```
ENABLE VOIP DEBUG=H323 ENGINE=FXS1
```

Related Commands DISABLE VOIP DEBUG

RESET VOIP

Syntax `RESET VOIP TYPE={SW|HW} [ENGINE=`*`engine`*`]`

where:

- *engine* is an engine name formed by concatenting 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).

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 the ENGINE parameter 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 TYPE=SW ENGINE=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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 set to ON, the port shows its phone number to the called party. If CLIP is set to 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 the DSCP or TOS 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 H.245 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 has a specific TOS or DSCP value to get a higher priority when switched along the path to their destinations. 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 INTERFACE=FXS0.1 PHONENUMBER=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 concatenting 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 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 ENTRY ENGINE=FXS0 PHONENUMBER=12345 HOSTIP=10.10.1.5
    PORT=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 (0-9). 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 (0-9). 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 how long, in seconds, the terminal waits for the other terminal to answer a call before treating the connection as down. The time must be in the range 5-255 seconds. 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 no Gatekeeper is 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 how long, in seconds, the terminal waits to receive an Alerting or Call Proceeding message when a call is placed, before treating the connection as down. The time must be in the range 5-255 seconds. The default is 20 seconds.

The TIMETOLIVE parameter specifies the interval between two consecutive registrations, between 10 and 10800 seconds. 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 GATEWAY GATEKEEPER=192.168.1.10:1719-OpenGK
    NAME=NEWGTW10 RASPORT=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. 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 letter in lower case (a-z), digits (0- 9) and the underscore character ("_") 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 (0- 9), the hyphen (-), and the underscore character (“_”). 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 the

DSCP or TOS 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 in RTP packets, as described in RFC2833. 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 in the range 5061- 49151, as odd numbers are reserved for the RTCP protocol. If not set, the 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 INTERFACE=FXS0.1 PHONENUMBER=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 49-28 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 GATEWAY DEFAULTPORT=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 concatenting 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).
- *ipaddr* is an IP address in dotted decimal notation.

Description

This command modifies an IP interface on a specific engine.

The only time you need to use this command is to change the PICs IP address is when there is a conflict between the PICs 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 solely used by the PIC to communicate with the router. The gateway must be in the same Class C subnet of the PICs IP address. By default, the PICs IP address is 192.168.255.picIndex where picIndex is the index of the PIC bay (e.g. 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 49-55 for the PICs 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 ENGINE=FXS0 IP=192.168.0.10 GATEWAY=192.168.0.10
```

Related Commands SHOW VOIP

SET VOIP BOOTCODE

Syntax `SET VOIP BOOTCODE=`*`filename`* `SERVER={`*`ipaddr`*`|`*`flash`*`}`

where:

- *filename* is a file name of the form `filename.bin`. Valid characters are lowercase letters, digits (0-9), and the hyphen (-).
- *ipaddr* is an IPv4 address in dotted decimal format.
- *flash* is the application code stored in the router's flash.

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.

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, specify SERVER=*flash*.

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

```
SET VOIP BOOTCODE=C-1-1-1.bin SERVER=202.36.163.22
```

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

```
SET VOIP BOOTCODE=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 of the form `filename.bin`. Valid characters are lowercase letters, digits (0-9), 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 the router's flash.

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 FILE=hs-1-0-1.bin PROTOCOL=H323 TYPE=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. nsm0.bay2.fxs0.0).
- *blen* is a decimal number in the range 30 to 500.
- *bthr* is a decimal number in the range 0 to *blen*.
- *cadence-values* is a comma separated list of up to 8 decimal numbers, each in the range 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 in the range 17 to 1000 Hz.
- *lecframe* is a decimal number in the range 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 any event validation characteristics, ringing threshold, tone detection, impedance etc. Available values are AUSTRIA, AUSTRALIA, CHINA, FRANCE, GERMANY1, GERMANY2, HOLLAND, ITALY, JAPAN, KOREA, NEWZEALAND, SPAIN, UK, USA1, and USA2. The default is configured by the router when the VoIP engine starts up.

The specific values for National Signalling Protocol settings for each country are shown in the tables from page 49-44 to page 49-48.

Table 49-2: 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 49-3: 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 49-4: 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 49-5: 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 49-6: 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 49-7: 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 49-8: 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 49-9: 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 49-10: 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 49-11: 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 49-12: 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 49-13: 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 49-14: 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	370Ω + 620Ω // 310 nF	
Tx Gain	+3 dB	
Rx Gain	-9 dB	

Table 49-15: 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 49-16: 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	350Ω + 1000Ω // 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 49-17 on page 49-49 shows the CADENCE parameter options that can be changed. If the CADENCE parameter is specified, either CVALUE or FVALUE, or both parameters must also be specified.

Table 49-17: 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 specifies the ON/OFF periods for the specified cadence, as a comma-separated list of decimal numbers: CVALUE=on1,off1... on4,off4. The CVALUE parameter must be specified when the CADENCE parameter is specified.

The DIGITTOUT parameter specifies the length of time, in seconds, after which the 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. The FVALUE must be specified when the CADENCE parameter 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 is specified, the PIC does not send voice packets during periods of silence. If OFF is specified, 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 PHONE INTERFACE=FXS0.0 TXGAIN=-3 RXGAIN=-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 in the range 0 to 15 (e.g. eth0). If a logical interface is not specified, 0 is assumed (i.e. 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 PUBLIC INTERFACE=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. nsm0.bay2.fxs0.0).

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

Figure 49-9: 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 49-18: Parameters displayed 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:

```
SHOW H323 ENGINE=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 concatenting 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).

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:

```
SHOW H323 ENTRY ENGINE=FXS2
```

Figure 49-10: 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 49-19: Parameters displayed 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 49-11: 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 49-20: Parameters displayed 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:

```
SHOW H323 GATEWAY
```

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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. nsm0.bay2.fxs0.0).

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

Figure 49-12: Example output from the SHOW SIP command.

```
SIP Module information
------------------------------------------------------------
Interface 0
   Phone Number      000555
   Authorisation     UserName: "eurord@alliedtelesyn.com"
                     Password: "welcome"
   Domain            alliedtelesyn.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 49-21: Parameters displayed 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 49-13: Example output from the SHOW SIP GATEWAY command.

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

Table 49-22: Parameters displayed 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 concatenting 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).

Description This command shows the VoIP PIC configuration and status.

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

Figure 49-14: 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 49-23: Parameters displayed in the output of the SHOW VOIP command.

Parameter	Meaning
Type	The engine type, either “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:

```
SHOW VOIP ENGINE=FXS1
```

Related Commands DISABLE VOIP
ENABLE VOIP
SET VOIP PHONE
SHOW VOIP LOAD

SHOW VOIP COUNTER ENGINE

Syntax `SHOW VOIP COUNTER [ENGINE=`*`engine`*`]`

where:

- *engine* is an engine name formed by concatenting 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).

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 49-15 on page 49-57, Table 49-24 on page 49-58

Figure 49-15: 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 49-24: Parameters displayed 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:

```
SHOW VOIP COUNTER ENGINE=FXS0
```

Related Commands SHOW VOIP

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 (e.g. 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 49-16: 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 49-25: Parameters displayed 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:

```
SHOW VOIP LOAD ENGINE=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 (e.g. fxs0.0). A fully qualified interface name may also be specified (e.g. nsm0.bay2.fxs0.0).

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

Figure 49-17: 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 49-26: Parameters displayed in the output of the SHOW VOIP PHONE command.

Parameter	Meaning
Country	Selected geographical region where the PIC is located: either “ITALY”, “AUSTRIA”, “FRANCE”, “GERMANY1”, “GERMANY2”, “UK”, “SPAIN”, “HOLLAND”, “JAPAN”, “CHINA”, “KOREA”, “USA1”, “USA2”, “AUSTRALIA”, “NEW ZEALAND”.
Ring	The ring parameters for Ring Cadence and Ring Frequency.
Tone	The 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, either “ON” or “OFF”.
Digit Tout	The length of 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:

```
SHOW VOIP PHONE INTERFACE=FXS0.0
```

Related Commands SET VOIP PHONE