

Using Cisco 3600 and Cisco 2600 Series Routers as H.323 VoIP Gateways

H.323 VoIP Gateway Feature Summary

In previous Cisco IOS Releases, the Cisco AS5300 voice service provider features included enhancements for both the functionality and configuration of gateways and Voice over IP (VoIP) gatekeepers. The architecture of these features provided the Quality of Service (QoS), stability, and functionality necessary for carrier class, real-time IP communications services. In this release, basic gateway Registration, Admission, and Status (RAS) protocol capability is extended to Cisco 3600 and Cisco 2600 series routers. Other features, such as authentication, authorization, and accounting (AAA) enhancements for security and accounting services, interactive voice response (IVR), Integrated Services Digital Network (ISDN) redirect number support, and rotary call pattern support, will be offered in future Cisco IOS releases.

Benefits

- Carrier-class voice quality
- End-to-end VoIP solutions
- Quality, scalability, and services

List of Terms

AAA—Authentication, authorization, and accounting. AAA is a suite of network security services that provides the primary framework through which access control can be set up on your Cisco router or access server.

dial peer—An addressable call endpoint. In Voice over IP (VoIP), there are two types of dial peers: POTS and VoIP.

DNS—Domain name system used in address translation to convert H.323 IDs, URLs, or e-mail IDs to IP addresses. DNS is also used to assist in the location of remote gatekeepers and to reverse-map raw IP addresses to host names of administrative domains.

DNIS—Dialed number identification service (the called number). Feature of trunk lines where the called number is identified; this called number information is used to route the call to the appropriate service.

DSP—Digital signal processor.

DTMF—Dual tone multi-frequency.

E.164—International Telecommunication Union (ITU-T) recommendation for international telecommunication numbering. This recommendation provides the number structure and functionality for the 3 categories of numbers used for international public telecommunication—geographic areas, global services, and networks.

E&M—E&M stands for recEive and transMit (or Ear and Mouth). E&M is a trunking arrangement generally used for two-way switch-to-switch or switch-to-network connections. Cisco's E&M interface is an RJ-48 connector that allows connections to PBX trunk lines (tie lines).

endpoint—An H.323 terminal or gateway. An endpoint can call and be called. It generates and/or terminates the information stream.

gatekeeper—An H.323 entity on the LAN that provides address translation and control access to the LAN for H.323 terminals and gateways. The gatekeeper can provide other services to the H.323 terminals and gateways, such as bandwidth management and locating gateways. A gatekeeper maintains a registry of devices in the multimedia network. The devices register with the gatekeeper at startup, and request admission to a call from the gatekeeper.

gateway—An H.323 endpoint on the LAN that provides real-time, two-way communications between H.323 terminals on the LAN and other ITU-T terminals in the WAN, or to another H.323 gateway. A gateway allows H.323 terminals to communicate with non-H.323 terminals by converting protocols. A gateway is the point at which a circuit-switched call is encoded and repackaged into IP packets.

H.323—An ITU-T standard that describes packet-based video, audio, and data conferencing. H.323 is an umbrella standard that describes the architecture of the conferencing system, and refers to a set of other standards (H.245, H.225.0, and Q.931) to describe its actual protocol.

H.323 RAS—Registration, admission, and status. The RAS signaling protocol performs registration, admissions, bandwidth changes, status and disengage procedures between the VoIP gateway and the gatekeeper.

POTS—Plain Old Telephone Service. Basic telephone service supplying standard single line telephones, telephone lines, and access to the public switched telephone network.

PSTN—Public switched telephone network. PSTN refers to the local telephone company.

QoS—Quality of service, which refers to the measure of service quality provided to the user.

Technology prefix—Discriminators used to distinguish between gateways having specific capabilities within a given zone. In the exchange between the gateway and the gatekeeper, the technology prefix is used to select a gateway after the zone has been selected. Technology prefixes can be used to tell the gatekeeper that a certain technology is associated with a particular call (for example, 15# could mean a fax transmission). No standard defines what the numbers in a technology prefix mean; by convention, technology prefixes are designated by a pound (#) symbol as the last character.

VoIP—Voice over IP. The ability to carry normal telephone-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP is a blanket term that generally refers to Cisco's standards based (for example, H.323) approach to IP voice traffic.

zone—A collection of all terminals, gateways, and multipoint control units (MCUs) managed by a single gatekeeper. A zone includes at least one terminal, and can include gateways or MCUs. A zone has only one gatekeeper. A zone can be independent of LAN topology and can be comprised of multiple LAN segments connected using routers or other devices.

Restrictions

The H.323 gateway feature and supporting software applications in Cisco IOS Release 12.0(2)XD require VCWare version 2.4.

Platforms

This feature is supported on the following platforms:

- Cisco AS5300 universal access server
- Cisco 2600 series routers
- Cisco 3600 series routers

Prerequisites

Before you can configure your Cisco 3600 series router to serve as an H.323 VoIP gateway, you must first:

- Establish a working IP network. For more information about configuring IP, refer to the “IP Overview,” “Configuring IP Addressing,” and “Configuring IP Services” chapters in the Cisco IOS Release 12.0 *Network Protocols Configuration Guide, Part 1*.
- Install the 1-slot or 2-slot (NM-1V/NM-2V) voice network module into the appropriate bay of your Cisco router. For more information about the physical characteristics of the voice network module, or how to install it, refer to the installation documentation, *Voice Network Module and Voice Interface Card Configuration Note*, that came with your voice network module.
- Configure Voice over IP. For more information about configuring Voice over IP, refer to the *Voice over IP for the Cisco 3600 and Cisco 2600 Series Software Configuration Guide*.

Supported MIBS and RFCs

None.

Functional Description

The Cisco VoIP gateway is a high-performance H.323-compliant gateway optimized for VoIP applications. Supporting up to two T1/E1 digital channels, it connects with existing telephones and fax machines through the Public Switched Telephone Network (PSTN), key systems, and PBXes, so that the process of placing calls over the IP network is transparent to users.

The gateway capability allows a Cisco 3600 or Cisco 2600 series router to function as an H.323 endpoint. Therefore, the gateway provides admission control, and address lookup and translation.

Gateway RAS Implementation

The Registration, Admission, and Status (RAS) signaling protocol performs registration, admissions, status, and disengage procedures between the H.323 VoIP gateway and the H.323 VoIP gatekeeper.

The following additions have been made to basic gateway functionality to implement RAS:

- Technology Prefixes
- Additional Keyword for the Session Target Command

Note For additional information regarding the H.323 VoIP gatekeeper, refer to *Configuring the Cisco AS5300 for Voice Service Provider Features* documentation.

Technology Prefixes

Technology prefixes are used to distinguish between gateways having specific capabilities within a given zone. In the exchange between the gateway and the gatekeeper, the technology prefix is used to select a gateway after the zone has been selected. You use the **tech-prefix** dial-peer configuration command to define technology prefixes.

In most cases, there is a dynamic protocol exchange between the gateway and the gatekeeper that enables the gateway to inform the gatekeeper about technology prefixes and where to forward calls. If, for some reason, that dynamic registry feature is not in effect, you can statically configure the gatekeeper to query the gateway for this information by configuring the **gw-type-prefix** command on the gatekeeper. Use the **show gatekeeper gw-type-prefix** to display how the gatekeeper has mapped the technology prefixes to local gateways.

For more information about configuring H.323 gateways, refer to Cisco IOS Release 11.3(6)NA 2 *Configuring H.323 VoIP Gateway for Cisco Access Platforms*. For more information about configuring H.323 gatekeepers, refer to Cisco IOS Release 11.3(6) 2 *Configuring H.323 VoIP Gatekeeper for Cisco Access Platforms*.

Additional Keyword for the Session Target Command

The **session target** dial-peer command indicates the address of the remote gateway where the call is terminated. A new keyword has been added to the session target command to indicate that the RAS protocol is being used—meaning that a gatekeeper will be consulted to translate the E.164 address to an IP address.

You can now define the address of a remote gateway in the following ways:

- Using an IP address: **session target ipv4:A.B.C.D**
- Using DNS: **session target dns: gateway@domain**
- Using RAS: **session target ras**

For more information about this revised command, refer to the Command Reference section.

Configuration Task List

To configure a Cisco 3600 or Cisco 2600 series router to perform as an H.323 VoIP gateway using RAS, perform the following tasks:

- Configure Voice Port Parameters
- Configure the H.323 Gateway

Configure Voice Port Parameters

To configure voice port parameters for an E&M voice port, use the following commands:

Step	Command	Purpose
1	configure terminal	Enter global configuration mode.
2	voice-port slot-number/subunit-number/port	Enter voice-port configuration mode to configure parameters for the specified voice port. The voice port is identified by its slot number, subunit number, and port number.
3	operation {2-wire 4-wire}	Specify an E&M cabling scheme for this voice-port.
4	type {1 2 3 5}	Specify the E&M port type.

For more information about configuring voice ports, refer to the “Configuring Voice Ports” chapter in the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Configuration Guide*.

Verify Voice Port Parameters

To verify voice port parameters, use the **show voice port** command.

The following is sample output from the **show voice port** command for an E&M voice port on the Cisco 3600 series:

```
router# show voice port 1/0/0
E&M Slot is 1, Sub-unit is 0, Port is 0
Type of VoicePort is E&M
Operation State is UP
Administrative State is UP
<--snip-->

Voice card specific Info Follows:
Signal Type is wink-start
Operation Type is 4-wire
Impedance is set to 600r Ohm
E&M Type is 5
Dial Type is dtmf
<--snip-->
```

For more information about the **show voice port** command, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Command Reference*.

Configure the H.323 Gateway

To configure the H.323 Gateway, you need to perform the following tasks:

- Configure POTS and VoIP Dial Peers
- Enable VoIP Gateway Functionality
- Configure Gateway Interface Parameters

Configure POTS and VoIP Dial Peers

The first step in configuring the H.323 gateway is to define the applicable POTS and VoIP dial peers. The POTS dial peer informs the system which voice port to direct incoming VoIP calls to and (optionally) defines that RAS-initiated calls will have a technology prefix prepended to the destination telephone number. The VoIP dial peer defines how to direct calls that originate from a local voice port into the VoIP cloud to the session target. The session target indicates the address of the remote gateway where the call is terminated. As mentioned, there are several different ways to define the destination gateway address: by statically configuring the IP address of the gateway, by defining the DNS of the gateway, or by using RAS. If you use RAS, that gateway determines the destination target by querying the RAS gatekeeper.

To configure POTS dial peer for the H.323 gateway, use the following commands:

Step	Command	Purpose
1	configure terminal	Enter global configuration mode.
2	dial-peer voice tag-number pots	Enter dial-peer configuration mode to configure parameters for the specified dial peer. The specific dial peer is identified by the tag number.
3	destination-pattern [tech-prefix#]string[t]	Specify the telephone number (E.164 or otherwise) associated with this dial peer. Use the <i>tech-prefix</i> argument to define a technology prefix. By adding this technology prefix, you are enabling the system to identify destination-patterns containing this prefix as RAS-initiated calls. Note If the gateway will be terminating RAS-initiated calls from another gateway—meaning that the other gateway is using RAS—you must add a technology prefix to the destination pattern.
4	port slot-number/subunit-number/port	Specify the voice port through which incoming VoIP calls will be received.
5	exit	Exit the dial-peer configuration mode.

For more information about POTS dial peers, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Configuration Guide*. For more information about any of the commands used to configure VoIP dial peers, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Command Reference*.

To configure VoIP dial peer for the H.323 gateway, use the following commands:

Step	Command	Purpose
1	configure terminal	Enter global configuration mode.
2	dial-peer voice <i>tag-number</i> voip	Enter dial-peer configuration mode to configure parameters for the specified dial peer. The specific dial peer is identified by the tag number.
3	destination-pattern [+] <i>string[t]</i>	Specify the telephone number (E.164 or otherwise) associated with this dial peer.
4	tech-prefix <i>tech-prefix</i>	Specify the technology prefix to be prepended to the destination pattern.
5	session-target ras	Specify that the RAS protocol is being used to determine the IP address of the session target—meaning that a gatekeeper will be consulted to translate the E.164 address to an IP address.
6	exit	Exit the dial-peer configuration mode.

For more information about POTS dial peers, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Configuration Guide*. For more information about any of the commands used to configure POTS dial peers, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Command Reference*.

Verify Dial Peer Configuration

Verify the POTS and VoIP dial-peer configuration by using the **show dial-peer voice** command.

The following example shows output from the **show dial-peer voice** command for a VoIP dial peer using RAS:

```
5300-1# show dial-peer voice 1234

VoiceOverIpPeer1234
 tag = 1234, destination-pattern = 1234',
 answer-address = ',
 group = 1234, Admin state is up, Operation state is up,
 incoming called-number = ', connections/maximum = 0/unlimited,
 application associated:
 type = voip, session-target = ras',
 technology prefix: 8#
 ip precedence = 0, UDP checksum = disabled,
 session-protocol = cisco, req-qos = controlled-load,
 acc-qos = best-effort,
 fax-rate = voice, codec = g729r8,
 Expect factor = 10, Icpif = 30,
 VAD = enabled, Poor QOV Trap = disabled,
<--snip-->
```

Enable VoIP Gateway Functionality

The next step in configuring an H.323 gateway is to enable VoIP gateway functionality by using the **gateway** command.

To enable gateway functionality, use the following commands:

Step	Command	Purpose
1	configure terminal	Enter global configuration mode.
2	gateway	Enable the VoIP gateway.

Configure Gateway Interface Parameters

The next step in configuring an H.323 gateway is to configure the gateway interface parameters. First define which interface will be presented to the VoIP network as this gateway's H.323 interface. Only one interface is allowed to be the gateway interface. You can select either the interface that is connected to the gatekeeper or a loopback interface. The interface that is connected to the gatekeeper is usually a LAN interface (for example, Fast Ethernet, Ethernet, FDDI, or Token Ring).

After you define the gateway interface, configure the gateway to discover the gatekeeper either through multicasting or by directing it to a specific host. Then configure the gateway's H.323 identification number and any technology prefixes that this gateway should register with the gatekeeper.

To define the interface to be used as the H.323 gateway interface and configure the H.323 gateway interface parameters, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	interface <i>type slot/port</i>	Enter interface configuration mode to configure parameters for the specified interface.
2	ip address <i>ip-address subnet-mask</i>	Specify the IP address for this interface.
3	h323-gateway voip interface	Designate this interface as the H.323 gateway interface.
4	h323-gateway voip h323-id <i>interface-id</i>	Specify an H.323 name (ID) for the gateway associated with this interface. This ID is used by this gateway when this gateway communicates with the gatekeeper. Usually, this H.323 ID is the name given to the gateway with the gatekeeper domain name appended to the end.
5	h323-gateway voip id <i>gatekeeper {ipaddr ip-address [port]} multicast</i>	Specify the name (ID) of the gatekeeper associated with this gateway and how the gateway finds it. The gatekeeper ID configured here must exactly match the gatekeeper ID in the gatekeeper configuration. The gateway determines the location of the gateway in one of two ways: either by a defined IP address or through multicast.

Step	Command	Purpose
6	h323-gateway voip tech-prefix <i>prefix</i>	Specify a technology prefix. A technology prefix is used to identify a type of service that this gateway is capable of providing. Note If a gateway is capable of handling multiple services, specify each service with a tech-prefix command.
7	exit	Exit interface configuration mode.
8	exit	Exit global configuration mode.

Verify the Gateway Interface Configuration

- Use the **show gateway** privileged EXEC command to check that the gateway has found and registered itself with the gatekeeper.

In the following example, gateway GW13.cisco.com has successfully registered itself with gatekeeper GK15.cisco.com:

```
hostname#show gateway
Gateway GW13.cisco.com is registered to Gatekeeper GK15.cisco.com
```

- Use the **show gatekeeper gw-type-prefix** command to verify that the gateway has successfully registered its technology prefixes with the gatekeeper. This command displays all registers and status tech-prefixes on a Cisco gatekeeper.

In the following example, gatekeeper 'GK15.cisco.com' has 4 technology prefixes. Prefix 31# is statically defined on the gatekeeper, while 23#, 22#, and 13# have been dynamically registered by gateways.

```
hostname#show gatekeeper gw-type-prefix
GATEWAY TYPE PREFIX TABLE
=====
Prefix:31#*
  Statically-configured gateways (not necessarily currently registered):
    1.9.53.13:1720
  Zone:GK15.cisco.com Registered gateways for this prefix:
    1.9.53.13:1720 GW13

Prefix:23#*
  Zone:GK15.cisco.com Registered gateways for this prefix:
    1.9.74.23:1720 GW23-c2611

Prefix:22#*
  Zone:GK15.cisco.com Registered gateways for this prefix:
    1.9.74.22:1720 GW22-c2611

Prefix:13#*
  Zone:GK15.cisco.com Registered gateways for this prefix:
    1.9.53.13:1720 GW13
```

Tips

- Use the **debug ras** command to display the types and addressing of RAS messages sent and received. The debug output lists the message type using mnemonics defined in ITU-T specification H.225.

In the following example, gateway GW13.cisco.com sends a RAS registration request message (RRQ) to gatekeeper GK15.cisco.com at IP address 1.9.53.15. GW13.cisco.com then receives a registration confirmation (RCF) message from the gatekeeper. If there is no response, it could mean that the gatekeeper is offline or improperly addressed. If you receive a reject message (RRJ), it could mean that the gatekeeper is unable to handle another gateway or that the registration information is incorrect.

```
hostname#debug ras
*Mar 13 19:53:34.231:      RASLib::ras_sendto:msg length 105 from
                        1.9.53.13:8658 to 1.9.53.15:1719
*Mar 13 19:53:34.231:      RASLib::RASSendRRQ:RRQ (seq# 36939) sent
                        to 1.9.53.15
*Mar 13 19:53:34.247:      RASLib::RASRecvData:successfully rcvd
                        message of length 105 from 1.9.53.15:1719
*Mar 13 19:53:34.251:      RASLib::RASRecvData:RCF (seq# 36939) rcvd
                        from [1.9.53.15:1719] on sock [0x6168356C
```

- Use the **debug h225 asn1** commands to display additional information about the actual contents of the H.225 RAS messages.

In the following example, the **debug h225 asn1** command displays the actual contents of the RAS registration message exchange between gateway GW13.cisco.com and gatekeeper GK15.cisco.com. The **debug h225 asn1** command also displays the technology prefixes that gateway GW13.cisco.com is registering:

```
hostname#debug h225 asn1
H.225 ASN1 Messages debugging is on
3640GW.13#value RasMessage ::= registrationRequest :
*Mar 13 20:16:45.295:  {
*Mar 13 20:16:45.295:    requestSeqNum 037001,
*Mar 13 20:16:45.295:    protocolIdentifier { 0 0 8 2250 0 1 },
*Mar 13 20:16:45.295:    discoveryComplete TRUE,
*Mar 13 20:16:45.295:    callSignalAddress
*Mar 13 20:16:45.295:    {
*Mar 13 20:16:45.295:      ipAddress :
*Mar 13 20:16:45.295:      {
*Mar 13 20:16:45.295:        ip '0109350D'H,
*Mar 13 20:16:45.295:        port 01720
*Mar 13 20:16:45.295:      }
*Mar 13 20:16:45.295:    },
*Mar 13 20:16:45.295:    rasAddress
*Mar 13 20:16:45.295:    {
*Mar 13 20:16:45.295:      ipAddress :
*Mar 13 20:16:45.295:      {
*Mar 13 20:16:45.295:        ip '0109350D'H,
*Mar 13 20:16:45.295:        port 04635
*Mar 13 20:16:45.295:      }
*Mar 13 20:16:45.295:    },
*Mar 13 20:16:45.295:    terminalType
*Mar 13 20:16:45.295:    {
*Mar 13 20:16:45.295:      gateway
*Mar 13 20:16:45.295:      {
*Mar 13 20:16:45.295:        protocol
*Mar 13 20:16:45.295:        {
*Mar 13 20:16:45.295:          voice :
*Mar 13 20:16:45.295:          {
*Mar 13 20:16:45.295:            supportedPrefixes
*Mar 13 20:16:45.295:            {
```

```

*Mar 13 20:16:45.299:          {
*Mar 13 20:16:45.299:          prefix e164 : "13#"
*Mar 13 20:16:45.299:          }
*Mar 13 20:16:45.299:      }
*Mar 13 20:16:45.299:      }
*Mar 13 20:16:45.299:      },
*Mar 13 20:16:45.299:      mc FALSE,
*Mar 13 20:16:45.299:      undefinedNode FALSE
*Mar 13 20:16:45.299:  },
*Mar 13 20:16:45.299:  terminalAlias
*Mar 13 20:16:45.303:  {
*Mar 13 20:16:45.303:      h323-ID : "GW13.cisco.com"
*Mar 13 20:16:45.303:  },
*Mar 13 20:16:45.303:  gatekeeperIdentifier "GK15.cisco.com",
*Mar 13 20:16:45.303:  endpointVendor
*Mar 13 20:16:45.303:  {
*Mar 13 20:16:45.303:      vendor
*Mar 13 20:16:45.303:      {
*Mar 13 20:16:45.303:          t35CountryCode 0181,
*Mar 13 20:16:45.303:          t35Extension 00,
*Mar 13 20:16:45.303:          manufacturerCode 018
*Mar 13 20:16:45.303:      }
*Mar 13 20:16:45.303:  }
*Mar 13 20:16:45.303: }
*Mar 13 20:16:45.303:0CC09088 06000891 4A000180 01000109 350D06B8 01000109 350D121B
0880013C
05050100 40460000 01400D00 47005700 31003300 40006300 69007300 63006F00
2E006300 6F006D1A 0047004B 00310035 002E0063 00690073 0063006F 002E0063
006F006D 00B50000 12
10C09088 06000891 4A000100 01400D00 47005700 31003300 40006300 69007300
63006F00 2E006300 6F006D1A 0047004B 00310035 002E0063 00690073 0063006F
002E0063 006F006D 1E003600 30004400 37004500 37003300 43003000 30003000
30003000 30003000 46
value RasMessage ::= registrationConfirm :
*Mar 13 20:16:45.335: {
*Mar 13 20:16:45.335:   requestSeqNum 037001,
*Mar 13 20:16:45.335:   protocolIdentifier { 0 0 8 2250 0 1 },
*Mar 13 20:16:45.335:   callSignalAddress
*Mar 13 20:16:45.335:   {
*Mar 13 20:16:45.335:   },
*Mar 13 20:16:45.335:   terminalAlias
*Mar 13 20:16:45.335:   {
*Mar 13 20:16:45.335:       h323-ID : "GW13.cisco.com"
*Mar 13 20:16:45.339:   },
*Mar 13 20:16:45.339:   gatekeeperIdentifier "GK15.cisco.com",
*Mar 13 20:16:45.339:   endpointIdentifier "60D7E73C0000000F"
*Mar 13 20:16:45.339: }
*Mar 13 20:16:45.339:
3640GW.13#

```

Configuration Examples

This section contains the following configuration examples:

- Configuring an H.323 Gateway
- Configuring a RAS Gatekeeper

Configuring an H.323 Gateway

The following example shows how to configure a Cisco 3600 series router as an H.323 gateway:

```
! Configure the voice-port parameters.
! This voice-port is an analog E&M-wink port using 4-wire, type 5 interface
!
voice-port 2/0/0
  operation 4-wire
  type 5
!
! Setup a pots dial peer to direct calls incoming VoIP calls to the voice-port.
! This dial peer defines that the RAS initiated call will be received with a tech
! prefix of 13#
!
dial-peer voice 13200 pots
  destination-pattern 13#13200
  port 2/0/0
!
! Setup a VoIP dial-peer to direct calls originated from a local voice-port
! into the VoIP cloud. In this example, the session target indicates
! that the destination target is determined by querying the RAS gatekeeper.
! The tech-prefix command means that the H.323 gateway will ask the RAS gatekeeper to
! direct calls using the technology prefix of 14#.
!
dial-peer voice 14 voip
  destination-pattern 14...
  tech-prefix 14#
  session target ras
!
! Enable Gateway functionality with global config command.
!
gateway
!
! Choose an interface to be this gateway's H.323 interface. In this example, the
! gateway is directed toward a specific host. Then define this gateway's H.323 ID, and
! configure any tech prefixes that this gateway should register with the gatekeeper.
! In this example, gateway GW13 tells gatekeeper GK15 to route any calls with a pattern
! than begins with 13# to GW13. Dial-peer 14 expects that some other gateway has
! register tech-prefix 14#.
!
interface Ethernet0/0
  ip address 172.9.53.13 255.255.255.0
  h323-gateway voip interface
  h323-gateway voip id GK15.cisco.com ipaddr 172.9.53.15 1719
  h323-gateway voip h323-id GW13@cisco.com
  h323-gateway voip tech-prefix 13#
!
```

Configuring a RAS Gatekeeper

For RAS to work on an H.323 gateway, you need to configure a corresponding RAS gatekeeper. The following example configures a Cisco 3600 series router as a RAS gatekeeper. For more information about configuring gatekeepers, refer to the Cisco IOS Release 11.3(6)NA2 document, *Configuring the Cisco AS5300 for Voice Service Provider Features*.

```

! Define this Ethernet port as the RAS gatekeeper.
interface Ethernet0/0
 ip address 172.9.53.15 255.255.255.0
!
gatekeeper
!
! Specify the name of the local zone that this gatekeeper manages. Specify the IP
! address that the gatekeeper advertises.
zone local GK15.cisco.com cisco.com 172.9.53.15
!
! Statically define a remote zone and the associated gatekeeper's IP address.
zone remote GK21.cisco.com cisco.com 172.9.74.21 1719
!
! Statically define the E.164 prefixes that a remote zone handles. This causes GK15 to
! direct any call with a called number that matches 22* (22 and any number of trailing
! digits) to GK21. This is not the same as a tech prefix. If a call comes in with an
! E.164 pattern of (220) 555-1234, it will be routed to GK21 because the pattern
! matches 22*.
zone prefix GK21.cisco.com 22*
zone prefix GK21.cisco.com 23*
!
! Statically define a tech prefix routing. Any call that comes in to the gatekeeper
! with a technology prefix of 88# (the * catches any following E.164 address), is
! directed to the gateway at IP address 172.9.53.13. This is a static technology prefix
! definition. The gateway can also dynamically register its tech-prefixes with the
! gatekeeper.
gw-type-prefix 88#* gw ipaddr 172.9.53.13 1720
!
! Activate the gatekeeper function by activating the port.
no shutdown
!

```

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference.

- **gateway**
- **h323-gateway voip h.323 id**
- **h323-gateway voip id**
- **h323-gateway voip interface**
- **h323-gateway voip tech-prefix**
- **session target**
- **show gateway**
- **tech-prefix**

gateway

To enable the H.323 VoIP gateway, use the **gateway** global configuration command. Use the **no** form of this command to cancel the registration of this gateway with the gatekeeper.

gateway
no gateway

Syntax Description

This command has no keywords or arguments.

Default

The gateway is not registered.

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Use the gateway command to enable H.323 VoIP gateway functionality. After you enable the gateway, it will attempt to discover a gatekeeper by using the H.323 RAS GRQ message. If you enter the **no gateway** command, the H.323 VoIP gateway will cancel this registration with the gatekeeper using the H.323 RAS URQ message.

Example

The following example enables the H.323 gateway:

```
configure terminal
gateway
```

h323-gateway voip h.323-id

To configure the H.323 name of the gateway identifying this gateway to its associated gatekeeper, use the **h323-gateway voip h.323-id** interface configuration command. Use the **no** form of this command to disable this defined gateway name.

```
h323-gateway voip h323-id interface-id
no h323-gateway voip h323-id interface-id
```

Syntax Description

<i>interface-id</i>	H.323 name (ID) used by this gateway when this gateway communicates with its associated gatekeeper. Usually, this ID is the name of the gateway with the gatekeeper's domain name appended to the end: <i>name@domain-name</i> .
---------------------	--

Default

No gateway identification is defined.

Command Mode

Interface configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Example

The following example configures Ethernet interface 0/0 as the gateway interface. In this example, the gateway ID is GW13@cisco.com.

```
interface Ethernet0/0
 ip address 172.9.53.13 255.255.255.0
 h323-gateway voip interface
 h323-gateway voip id GK15.cisco.com ipaddr 172.9.53.15 1719
 h323-gateway voip h323-id GW13@cisco.com
 h323-gateway voip tech-prefix 13#
```

Related Commands

```
h323-gateway voip id
h323-gateway voip interface
h323-gateway voip tech-prefix
```

h323-gateway voip id

To define the name and location of the gatekeeper for this gateway, use the **h323-gateway voip id** interface configuration command. Use the **no** form of this command to disable this gatekeeper identification.

```
h323-gateway voip id gatekeeper-id {ipaddr ip-address [port-number] | multicast}  
no h323-gateway voip id gatekeeper-id {ipaddr ip-address [port-number] | multicast}
```

Syntax Description

<i>gatekeeper-id</i>	Indicates the H.323 identification of the gatekeeper. This value must exactly match the gatekeeper ID in the gatekeeper configuration. The recommended format is <i>name.doman-name</i> .
ipaddr	Indicates that the gateway will use an IP address to locate the gatekeeper.
<i>ip-address</i>	Defines the IP address used to identify the gatekeeper.
port-number	(Optional) Specifies the optional UDP port number through which the gateway communicates with the gatekeeper. If not specified, the gateway uses port 1719 as the default.
multicast	Indicates that the gateway will use multicast to locate the gatekeeper.

Default

No gatekeeper identification is defined.

Command Mode

Interface configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

This command tells the H.323 gateway associated with this interface which H.323 gatekeeper to talk to and where to locate it. The gatekeeper ID configured here must exactly match the gatekeeper ID in the gatekeeper configuration.

Example

The following example configures Ethernet interface 0/0 as the gateway interface. In this example, the gatekeeper ID is GK15.cisco.com and its IP address is 172.9.53.15 (using port 1719).

```
interface Ethernet0/0  
 ip address 172.9.53.13 255.255.255.0  
 h323-gateway voip interface  
 h323-gateway voip id GK15.cisco.com ipaddr 172.9.53.15 1719  
 h323-gateway voip h323-id GW13@cisco.com  
 h323-gateway voip tech-prefix 13#
```

Related Commands

h323-gateway voip h323-id
h323-gateway voip interface
h323-gateway voip tech-prefix

h323-gateway voip interface

To configure this interface as an H.323 interface, use the **h323-gateway voip interface** interface configuration command. Use the **no** form of this command to disable H.323 functionality for this interface.

h323-gateway voip interface
no h323-gateway voip interface

Syntax Description

This command has no arguments or keywords.

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA 2.

Example

The following example configures Ethernet interface 0/0 as the gateway interface. In this example, the **h323-gateway voip interface** command configures this interface as an H.323 interface.

```
interface Ethernet0/0
 ip address 172.9.53.13 255.255.255.0
 h323-gateway voip interface
 h323-gateway voip id GK15.cisco.com ipaddr 172.9.53.15 1719
 h323-gateway voip h323-id GW13@cisco.com
 h323-gateway voip tech-prefix 13#
```

Related Commands

h323-gateway voip h323-id
h323-gateway voip id
h323-gateway voip tech-prefix

h323-gateway voip tech-prefix

To define the technology prefix that the gateway will register with the gatekeeper, use the **h323-gateway voip tech-prefix** interface configuration command. Use the **no** form of this command to disable this defined technology prefix.

```
h323-gateway voip tech-prefix prefix  
no h323-gateway voip tech-prefix prefix
```

Syntax Description

<i>prefix</i>	Defines the numbers used as the technology prefix. Each technology prefix can contain up to 11 characters. Although not strictly necessary, a pound (#) symbol is frequently used as the last digit in a technology prefix. Valid characters 0 though 9, the pound (#) symbol, and the asterisk (*).
---------------	--

Default

Disabled

Command Mode

Interface configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

This command defines a technology prefix that the gateway will then register with the gatekeeper. Technology prefixes can be used as a discriminator so that the gateway can tell the gatekeeper that a certain technology is associated with a particular call (for example, 15# could mean a fax transmission), or it can be used like an area code for more generic routing. No standard currently defines what the numbers in a technology prefix mean. By convention, technology prefixes are designated by a pound (#) symbol as the last character.

Note Cisco gatekeepers use the asterisk (*) as a reserved character. If you are using Cisco gatekeepers, do not use the asterisk as part of the technology prefix.

Example

The following example configures Ethernet interface 0/0 as the gateway interface. In this example, the technology prefix is defined as 13#.

```
interface Ethernet0/0  
ip address 172.9.53.13 255.255.255.0  
h323-gateway voip interface  
h323-gateway voip id GK15.cisco.com ipaddr 172.9.53.15 1719  
h323-gateway voip h323-id GW13@cisco.com  
h323-gateway voip tech-prefix 13#
```

Related Commands

h323-gateway voip id
h323-gateway voip interface
h323-gateway voip h323-id

session target

To specify a network-specific address for a specified dial peer, use the **session target** dial-peer configuration command. Use the **no** form of this command to disable the specified target address.

For the Cisco 3600 series Voice over IP:

```
session target {ipv4:destination-address | dns:[$$$. | $d$. | $e$. | $u$.] host-name |
loopback:rtp | loopback:compressed | loopback:uncompressed | ras}
no session target
```

Syntax Description

ipv4:destination-address	IP address of the dial peer.
dns:host-name	<p>Indicates that the domain name server will be used to resolve the name of the IP address. Valid entries for this parameter are characters representing the name of the host device.</p> <p>(Optional) You can use one of the following 3 wildcards with this keyword when defining the session target for VoIP peers:</p> <ul style="list-style-type: none"> • \$\$\$.—Indicates that the source destination pattern will be used as part of the domain name. • \$d\$.—Indicates that the destination number will be used as part of the domain name. • \$e\$.—Indicates that the digits in the called number will be reversed, periods will be added inbetween each digit of the called number, and that this string will be used as part of the domain name. • \$u\$.—Indicates that the unmatched portion of the destination pattern (such as a defined extension number) will be used as part of the domain name.
loopback:rtp	Indicates that all voice data will be looped back to the originating source. This is applicable for VoIP peers.
loopback:compressed	Indicates that all voice data will be looped back in compressed mode to the originating source. This is applicable for POTS peers.
loopback:uncompressed	Indicates that all voice data will be looped-back in uncompressed mode to the originating source. This is applicable for POTS peers.
ras	Indicates that the Registration, Admission and Status (RAS) signaling function protocol is being used—meaning that a gatekeeper will be consulted to translate the E.164 address to an IP address.

Default

The default for this command is enabled with no IP address or domain name defined.

Command Mode

Dial-peer configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(1)T.

Use the **session target** command to specify a network-specific address or domain name for a dial peer. Whether you select a network-specific address or a domain name depends on the session protocol you select.

The **session target loopback** command is used for testing the voice transmission path of a call. The loopback point will depend on the call origination and the loopback type selected.

The **session target dns** command can be used with or without the specified wildcards. Using the optional wildcards can reduce the number of VoIP dial-peer session targets you need to configure if you have groups of numbers associated with a particular router.

Use the **session target ras** command to specify that the RAS protocol is being used to determine the IP address of the session target.

Examples

The following example configures a session target using DNS for a host, “voice_router,” in the domain “cisco.com:”

```
dial-peer voice 10 voip
  session target dns:voice_router.cisco.com
```

The following example configures a session target using DNS, with the optional **\$u\$**. wildcard. In this example, the destination pattern has been configured to allow for any 4-digit extension, beginning with the numbers 1310222. The optional wildcard **\$u\$**. indicates that the router will use the unmatched portion of the dialed number—in this case, the 4-digit extension, to identify the dial peer. As in the previous example, the domain is “cisco.com.”

```
dial-peer voice 10 voip
  destination-pattern 1310222...
  session target dns:$u$.cisco.com
```

The following example configures a session target using dns, with the optional **\$d\$**. wildcard. In this example, the destination pattern has been configured for 13102221111. The optional wildcard **\$d\$**. indicates that the router will use the destination pattern to identify the dial peer in the “cisco.com” domain.

```
dial-peer voice 10 voip
  destination-pattern 13102221111
  session target dns:$d$.cisco.com
```

The following example configures a session target using DNS, with the optional **\$e\$**. wildcard. In this example, the destination pattern has been configured for 12345. The optional wildcard **\$e\$**. indicates that the router will reverse the digits in the destination pattern, add periods between the digits, and then use this reverse-exploded destination pattern to identify the dial peer in the “cisco.com” domain.

```
dial-peer voice 10 voip
  destination-pattern 12345
  session target dns:$e$.cisco.com
```

The following example configures a session target using RAS:

```
dial-peer voice 11 voifr
  destination-pattern 13102221111
  session target ras
```

Related Commands

destination-pattern
session protocol

show gateway

To display the current H.323 VoIP gateway status, use the **show gateway** privileged EXEC command.

```
show gateway
```

Syntax Description

This command has no keywords or arguments.

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Example Output

The following is sample output from the **show gateway** command. This sample output shows that gateway voip2@vm1lab is enabled and registered to gatekeeper gk1vm1lab.

```
router#show gateway  
Gateway voip2@vm1lab is registered to Gatekeeper gk1.vm1lab
```

tech-prefix

To specify a particular technology prefix to be prepended to the destination pattern of a specific dial peer, use the `tech-prefix` dial-peer configuration command. Use the **no** form of this command to disable the defined technology prefix for this dial peer.

tech-prefix *number*
no tech-prefix *number*

Syntax Description

<i>number</i>	Defines the numbers used as the technology prefix. Each technology prefix can contain up to 11 characters. Although not strictly necessary, a pound (#) symbol is frequently used as the last digit in a technology prefix. Valid characters 0 through 9, the pound (#) symbol, and the asterisk (*).
---------------	---

Default

No technology prefix is defined.

Command Mode

Dial-peer configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Technology prefixes are used to distinguish between gateways having specific capabilities within a given zone. In the exchange between the gateway and the gatekeeper, the technology prefix is used to select a gateway after the zone has been selected. Use the **tech-prefix** command to define technology prefixes.

Technology prefixes can be used as a discriminator so that the gateway can tell the gatekeeper that a certain technology is associated with a particular call (for example, 15# could mean a fax transmission), or it can be used like an area code for more generic routing. No standard defines what the numbers in a technology prefix mean; by convention, technology prefixes are designated by a pound (#) symbol as the last character.

In most cases, there is a dynamic protocol exchange between the gateway and the gatekeeper that enables the gateway to inform the gatekeeper about technology prefixes and where to forward calls. If, for some reason, that dynamic registry feature is not in effect, you can statically configure the gatekeeper to query the gateway for this information by configuring the **gw-type-prefix** command on the gatekeeper. Use the **show gatekeeper gw-type-prefix** to display how the gatekeeper has mapped the technology prefixes to local gateways.

Note Cisco gatekeepers use the asterisk (*) as a reserved character. If you are using Cisco gatekeepers, do not use the asterisk as part of the technology prefix.

The gateway prepends the technology prefix to the called number that matches the VoIP dial peer's destination pattern. It prepends the technology prefix when an admission request is made to the gatekeeper and when a subsequent call setup match is sent to the endpoint identified by the gatekeeper. This is why the technology prefix configured for a dial peer must match the technology prefix on the terminating gateway.

Examples

The following example defines a technology prefix of 14# for the specified dial peer. In this example, the technology prefix means that the H.323 gateway will ask the RAS gatekeeper to direct calls using the technology prefix of 14#.

```
dial-peer voice 10 voip
  destination-pattern 14...
  tech-prefix 14#
```

Related Commands

gw-type-prefix
show gatekeeper gw-type-prefix

Debug Commands

This section describes the following debug commands:

- `debug cch323 h225`
- `debug cch323 h245`
- `debug cch323 ras`
- `debug h225`
- `debug ras`

debug cch323 h225

Use the **debug cch323 h225** command to trace the state transition of the H.225 state machine based on the processed events. Use the **no** form of this command to disable debugging output.

debug cch323 h225
no debug cch323 h225

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Sample Output

The following output shows a series of states and events for the H.225 state machine:

```
20:59:17:Set new event H225_EVENT_SETUP
20:59:17:H225 FSM:received event H225_EVENT_SETUP while at state H225_IDLE
20:59:17:Changing from H225_IDLE state to H225_SETUP state
20:59:17:cch323_h225_receiver:received msg of type SETUPCFM_CHOSEN
20:59:17:H225 FSM:received event H225_EVENT_SETUP_CFM_IND while at state
H225_SETUP
20:59:17:Changing from H225_SETUP state to H225_ACTIVE state
20:59:17:Set new event H225_EVENT_H245_SUCCESS
20:59:17:H225 FSM:received event H225_EVENT_H245_SUCCESS while at state
H225_ACTIVE
20:59:20:Set new event H225_EVENT_RELEASE
20:59:20:H225 FSM:received event H225_EVENT_RELEASE while at state
H225_ACTIVE
20:59:20:Changing from H225_ACTIVE state to H225_WAIT_FOR_DRQ state
20:59:20:Set new event H225_EVENT_RAS_SUCCESS
20:59:20:H225 FSM:received event H225_EVENT_RAS_SUCCESS while at state
H225_WAIT_FOR_DRQ
20:59:20:Changing from H225_WAIT_FOR_DRQ state to H225_IDLE state
```

State Descriptions

Definitions of the different states of the H.225 state machine are as follows:

- **H225_IDLE**—Initial state of the H.225 state machine. The H.225 state machine is in this state before issuing a call setup request (for the outbound IP call case) or ready to receive an incoming IP call.
- **H225_SETUP**—Call setup state. The state machine transitions to this state after sending out a call setup request, or after the reception of an incoming call indication.

- H225_ALERT—Call alerting state. The state machine transitions to this state after sending the *alerting* message or after the reception of an *alerting* message from the peer.
- H225_CALLPROC—Call proceeding state.
- H225_ACTIVE—Call connected state. In this state, the call is active. The state machine transitions to this state after sending the *connect* message to the peer or after the reception of the *connect* message from the peer.
- H225_WAIT_FOR_ARQ—The H.225 state machine is waiting for the completion of the ARQ process from the RAS state machine.
- H225_WAIT_FOR_DRQ—The H.225 state machine is waiting for the completion of the DRQ process from the RAS state machine.
- H225_WAIT_FOR_H245—The H.225 state machine is waiting for the success or failure from the H.245 state machine.

Events Description

Definitions of the different events of the H.225 state machine are as follows:

- H225_EVENT_NONE— No event.
- H225_EVENT_ALERT—The H.225 state machine sends an alerting message to the peer.
- H225_EVENT_ALERT_IND—The H.225 state machine receives an *alerting* message from the peer.
- H225_EVENT_CALLPROC—The H.225 state machine sends a call proceeding message to the peer
- H225_EVENT_CALLPROC_IND—The H.225 state machine receives a call proceeding message from the peer.
- H225_EVENT_REJECT—The H.225 state machine rejects the call setup request from the peer.
- H225_EVENT_REJECT_IND—The H.225 state machine requests that a call setup to the peer is rejected.
- H225_EVENT_RELEASE—The H.225 state machine sends a release complete message to the peer.
- H225_EVENT_RELEASE_IND—The H.225 state machine receives a release complete message from the peer.
- H225_EVENT_SETUP—The H.225 state machine sends a setup message to the peer.
- H225_EVENT_SETUP_IND—The H.225 state machine receives a setup message from the peer.
- H225_EVENT_SETUP_CFM—The H.225 state machine sends a connect message to the peer.
- H225_EVENT_SETUP_CFM_IND—The H.225 state machine receives a connect message from the peer.
- H225_EVENT_RAS_SUCCESS—Indicates that the pending RAS operation is successful.
- H225_EVENT_RAS_FAILED—Indicates that the pending RAS operation failed.
- H225_EVENT_H245_SUCCESS—Indicates that the pending H.245 operation is successful.
- H225_EVENT_H245_FAILED—Indicates that the pending H.245 operation failed.

debug cch323 h245

Use the **debug cch323 h245** command to trace the state transition of the H.245 state machine based on the processed events. Use the **no** form of this command to disable debugging output.

debug cch323 h245
no debug cch323 h245

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

The H.245 state machines include the following 3 state machines:

- Master Slave Determination state machine
- Capability Exchange state machine
- Open Logical Channel state machine

Sample Output

The following output shows a series of states and events for the H.245 state machine:

```
20:58:23:Changing to new event H245_EVENT_MSD
20:58:23:H245 MS FSM:received event H245_EVENT_MSD while at state
H245_MS_NONE
20:58:23:changing from H245_MS_NONE state to H245_MS_WAIT state
20:58:23:Changing to new event H245_EVENT_CAP
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP while at state
H245_CAP_NONE
20:58:23:changing from H245_CAP_NONE state to H245_CAP_WAIT state
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_INDICATION
20:58:23:Changing to new event H245_EVENT_MS_IND
20:58:23:H245 MS FSM:received event H245_EVENT_MS_IND while at state
H245_MS_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_CAP_TRANSFER_INDICATION
20:58:23:Changing to new event H245_EVENT_CAP_IND
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_IND while at state
H245_CAP_WAIT
20:58:23:cch323_h245_receiver:received msg of type
M_H245_MS_DETERMINE_CONFIRM
20:58:23:Changing to new event H245_EVENT_MS_CFM
20:58:23:H245 MS FSM:received event H245_EVENT_MS_CFM while at state
H245_MS_WAIT
20:58:23:changing from H245_MS_WAIT state to H245_MS_DONE state
```

```

0:58:23:cch323_h245_receiver:received msg of type M_H245_CAP_TRANSFER_CONFIRM
20:58:23:Changing to new event H245_EVENT_CAP_CFM
20:58:23:H245 CAP FSM:received event H245_EVENT_CAP_CFM while at state
H245_CAP_WAIT
20:58:23:changing from H245_CAP_WAIT state to H245_CAP_DONE state
20:58:23:Changing to new event H245_EVENT_OLC
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC while at state
H245_OLC_NONE
20:58:23:changing from H245_OLC_NONE state to H245_OLC_WAIT state
20:58:23:cch323_h245_receiver:received msg of type
M_H245_UCHAN_ESTABLISH_INDICATION
20:58:23:Changing to new event H245_EVENT_OLC_IND
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_IND while at state
H245_OLC_WAIT
20:58:23:cch323_h245_receiver:received msg of type M_H245_UCHAN_ESTAB_ACK
20:58:23:Changing to new event H245_EVENT_OLC_CFM
20:58:23:H245 OLC FSM:received event H245_EVENT_OLC_CFM while at state
H245_OLC_WAIT
20:58:23:changing from H245_OLC_WAIT state to H245_OLC_DONE state

```

State Definitions

Definitions of the different states of the H.245 state machine are as follows:

- H245_MS_NONE— Initial state of the master slave determination state machine.
- H245_MS_WAIT—A Master Slave Determination message is sent, waiting for the reply.
- H245_MS_DONE— The result is in.
- H245_CAP_NONE—Initial state of the capabilities exchange state machine.
- H245_CAP_WAIT—A cap exchange message is sent, waiting for reply.
- H245_CAP_DONE—The result is in.
- H245_OLC_NONE—Initial state of the open logical channel state machine.
- H245_OLC_WAIT—OLC message sent, waiting for reply.
- H245_OLC_DONE—OLC done.

Event Definitions

Definitions of the different events of the H.245 state machine are as follows:

- H245_EVENT_MSD—The H.245 state machine sends an MSD message.
- H245_EVENT_MS_CFM—The H.245 state machine sends an MSD acknowledge message.
- H245_EVENT_MS_REJ—The H.245 state machine sends MSD reject message.
- H245_EVENT_MS_IND— The H.245 state machine receives an MSD message.
- H245_EVENT_CAP—The H.245 state machine sends a CAP message.
- H245_EVENT_CAP_CFM—The H.245 state machine sends a CAP acknowledge message.
- H245_EVENT_CAP_REJ—The H.245 state machine sends a CAP reject.
- H245_EVENT_CAP_IND—The H.245 state machine receives a CAP message.
- H245_EVENT_OLC—The H.245 state machine sends an OLC message.
- H245_EVENT_OLC_CFM—The H.245 state machine sends an OLC acknowledge message.

- H245_EVENT_OLC_REJ—The H.245 state machine sends an OLC reject message.
- H245_EVENT_OLC_IND—The H.245 state machine receives an OLC message.

debug cch323 ras

Use the **debug cch323 ras** command to trace the state transition of the RAS state machine based on the processed events. Use the **no** form of this command to disable debugging output.

debug cch323 ras
no debug cch323 ras

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

RAS operates in two state machines. One global state machine controls the overall RAS operation of the gateway. The other state machine is a per-call state machine that controls the active calls.

Sample Output

The following output shows a series of states and events for the RAS state machine:

```
20:58:49:Changing to new event CCH323_RAS_EVENT_SEND_RRQ
cch323_run_ras_sm:received event CCH323_RAS_EVENT_SEND_RRQ while at
CCH323_RAS_STATE_IDLE state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_RRQ state
cch323_ras_receiver:received msg of type RCF_CHOSEN
cch323_run_ras_sm:received event CCH323_RAS_EVENT_RCF while at CCH323_RAS_STATE_RRQ
state
cch323_run_ras_sm:changing to CCH323_RAS_STATE_IDLE state
20:58:59:cch323_perccall_ras_sm:received event CCH323_RAS_EVENT_NEWCALL while at
CCH323_RAS_STATE_IDLE state
20:58:59:cch323_perccall_ras_sm:changing to new state CCH323_RAS_STATE_ARQ
cch323_ras_receiver:received msg of type ACF_CHOSEN
20:58:59:cch323_perccall_ras_sm:received event CCH323_RAS_EVENT_ACF while at
CCH323_RAS_STATE_ARQ state
20:58:59:cch323_perccall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
20:59:02:cch323_perccall_ras_sm:received event CCH323_RAS_EVENT_CALLDISC while
at CCH323_RAS_STATE_ACTIVE state
20:59:02:cch323_perccall_ras_sm:changing to new state CCH323_RAS_STATE_DRQ
cch323_ras_receiver:received msg of type DCF_CHOSEN
20:59:02:cch323_perccall_ras_sm:received event CCH323_RAS_EVENT_DCF while at
CCH323_RAS_STATE_DRQ state
20:59:02:cch323_perccall_ras_sm:changing to new state CCH323_RAS_STATE_IDLE
20:59:04:cch323_perccall_ras_sm:received event CCH323_RAS_EVENT_IRR while at
CCH323_RAS_STATE_ACTIVE state
20:59:04:cch323_perccall_ras_sm:changing to new state
CCH323_RAS_STATE_ACTIVE
```

State Definitions

Definitions of the different events of the RAS state machine are as follows:

- CCH323_RAS_STATE_NONE—Initial state of the RAS state machine.
- CCH323_RAS_STATE_GRQ—State machine is in the GRQ state. In this state, the gateway is in the process of discovering a gatekeeper.
- CCH323_RAS_STATE_RRQ—State machine is in the RRQ state. In this state, the gateway is in the process of registering with a gatekeeper.
- CCH323_RAS_STATE_IDLE—Global state machine is in the idle state.
- CCH323_RAS_STATE_URQ—State machine is in the URQ state. In this state, the gateway is in the process of unregistering with a gatekeeper.
- CCH323_RAS_STATE_ARQ—Per-call state machine is in the process of admitting a new call.
- CCH323_RAS_STATE_ACTIVE—Per-call state machine is in the call active state.
- CCH323_RAS_STATE_DRQ—Per-call state machine is in the process of disengaging an active call.

Event Definitions

Definitions of the different events of the RAS state machine are as follows:

- CCH323_RAS_EVENT_NONE—No event.
- CCH323_RAS_EVENT_GWUP—Gateway is coming up.
- CCH323_RAS_EVENT_GWDWN—Gateway is going down.
- CCH323_RAS_EVENT_NEWCALL:—New call.
- CCH323_RAS_EVENT_CALLDISC—Call disconnect.
- CCH323_RAS_EVENT_GCF—Received GCF.
- CCH323_RAS_EVENT_GRJ—Received GR.J
- CCH323_RAS_EVENT_ACF—Received ACF.
- CCH323_RAS_EVENT_ARJ—Received ARJ.
- CCH323_RAS_EVENT_SEND_RRQ—Send RRQ.
- CCH323_RAS_EVENT_RCF—Received RCF.
- CCH323_RAS_EVENT_RRJ—Received RRJ.
- CCH323_RAS_EVENT_SEND_URQ—Send URQ.
- CCH323_RAS_EVENT_URQ—Received URQ.
- CCH323_RAS_EVENT_UCF—Received UCF.
- CCH323_RAS_EVENT_SEND_UCF—Send .UCF.
- CCH323_RAS_EVENT_URJ—Received URJ.
- CCH323_RAS_EVENT_BCF—Received BCF.
- CCH323_RAS_EVENT_BRJ—Received BRJ.
- CCH323_RAS_EVENT_DRQ—Received DRQ.
- CCH323_RAS_EVENT_DCF—Received DCF.

- CCH323_RAS_EVENT_SEND_DCF—Send DCF.
- CCH323_RAS_EVENT_DRJ—Received DRJ.
- CCH323_RAS_EVENT_IRQ—Received IRQ.
- CCH323_RAS_EVENT_IRR—Send IRR.
- CCH323_RAS_EVENT_TIMEOUT—Message timeout.

debug h225

Use the **debug h225** command to display additional information about the actual contents of H.225 RAS messages. Use the **no** form of this command to disable debug output.

```
debug h225 {asn1 | events}  
no debug h225 {asn1 | events}
```

Syntax Description

asn1	Indicates that only the ASN.1 contents of any H.225 message sent or received will be displayed.
events	Indicates that key Q.931 events that occur when placing a H.323 call from one gateway to another will be displayed.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2

Both versions of the debug H.225 command display information about H.225 messages. H.225 messages are used to exchange RAS information between gateways and gatekeepers as well as to exchange Q.931 information between gateways.

The **debug h225 events** command displays key Q.931 events that occur when placing a H.323 call from one gateway to another. Q.931 events are carried in H.225 messages. This command enables you to monitor Q.931 state changes such as setup, alert, connected, and released.

Note Although the debug information includes the hexadecimal output of the entire H.225 message, only the key state changes are decoded.

The **debug h225 asn1** command displays the ASN.1 contents of any H.225 message sent or received that contains ASN.1 content. Not all H.225 messages contain ASN.1 content. Some messages contain both Q.931 information and ASN.1 information; if you enter this command, only ASN.1 information will be displayed.

Sample Output

The following sample output for the **debug h225 events** command shows a call being placed from gateway GW13 to gateway GW14. Before the call was placed, the gateway exchanged RAS messages with the gatekeeper. Because RAS messages do not contain Q.931 information, these messages do not appear in this output.

```
3640GW.13# debug h225 events
H.225 Event Messages debugging is on
3640GW.13#

*Mar  2 02:47:14.689:      H225Lib::h225TConn:connect in progress on socket [2]
*Mar  2 02:47:14.689:      H225Lib::h225TConn:Q.931 Call State is initialized to be
[Null].
*Mar  2 02:47:14.697:Hex representation of the SETUP TPKT to
send.0300004D080200DC05040380C0A36C09913133233133333303070099131342331343330307E00260500
80060008914A000102004B1F5E5D8990006C0000000005BF7454000C0700000000000000
*Mar  2 02:47:14.701:
*Mar  2 02:47:14.701:      H225Lib::h225SetupRequest:Q.931 SETUP sent from socket [2]
*Mar  2 02:47:14.701:      H225Lib::h225SetupRequest:Q.931 Call State changed to [Call
Initiated].
*Mar  2 02:47:14.729:Hex representation of the received
TPKTO3000021080280DC013401017E0012050340060008914A000100000109350E2B28
*Mar  2 02:47:14.729:
*Mar  2 02:47:14.729:      H225Lib::h225RecvData:Q.931 ALERTING received from socket
[2]
*Mar  2 02:47:14.729:      H225Lib::h225RecvData:Q.931 Call State changed to [Call
Delivered].
*Mar  2 02:47:17.565:Hex representation of the received
TPKTO3000034080280DC07040380C0A37E0023050240060008914A0001000109350E2B2802004B1F5E5D899
0006C00000000005BF7454
*Mar  2 02:47:17.569:
*Mar  2 02:47:17.569:      H225Lib::h225RecvData:Q.931 CONNECT received from socket [2]
*Mar  2 02:47:17.569:      H225Lib::h225RecvData:Q.931 Call State changed to [Active].
*Mar  2 02:47:23.273:Hex representation of the received
TPKTO300001A080280DC5A080280107E000A050500060008914A0001
*Mar  2 02:47:23.273:
*Mar  2 02:47:23.273:      H225Lib::h225RecvData:Q.931 RELEASE COMPLETE received from
socket [2]
*Mar  2 02:47:23.273:      H225Lib::h225RecvData:Q.931 Call State changed to [Null].
*Mar  2 02:47:23.293:Hex representation of the RELEASE COMPLETE TPKT to
send.0300001A080200DC5A080280107E000A050500060008914A0001
*Mar  2 02:47:23.293:
*Mar  2 02:47:23.293:      H225Lib::h225TerminateRequest:Q.931 RELEASE COMPLETE sent
from socket [2]. Call state changed to [Null].
*Mar  2 02:47:23.293:      H225Lib::h225TClose:TCP connection from socket [2] closed
```

The following output shows the same call being placed from gateway GW13 to gateway GW14 using the **debug h225 asn1** command. The output is very long but you can track the following information:

- The admission request to the gatekeeper.
- The admission confirmation from the gatekeeper.
- The ASN.1 portion of the H.225/Q.931 setup message from the calling gateway to the called gateway.
- The ASN.1 portion of the H.225/Q.931 setup response from the called gateway, indicating that the call has proceeded to alerting state.
- The ASN.1 portion of the H.225/Q.931 message from the called gateway, indicating that the call has been connected.

- The ASN.1 portion of the H.225/Q.931 message from the called gateway, indicating that the call has been released.
- The ANS.1 portion of the H.225 RAS message from the calling gateway to the gatekeeper, informing it that the call has been disengaged.
- The ASN.1 portion of the H.225 RAS message from the gatekeeper to the calling gateway, confirming the disengage request.
- The ASN.1 portion of the H.225/Q.931 release complete message sent from the called gateway to the calling gateway.

```

3640GW.13# debug h225 asn1
H.225 ASN1 Messages debugging is on
3640GW.13#

value RasMessage ::= admissionRequest :
*Mar 2 02:48:18.445: {
*Mar 2 02:48:18.445:   requestSeqNum 03320,
*Mar 2 02:48:18.445:   callType pointToPoint :NULL,
*Mar 2 02:48:18.445:   callModel direct :NULL,
*Mar 2 02:48:18.445:   endpointIdentifier "60D6BA4C00000001",
*Mar 2 02:48:18.445:   destinationInfo
*Mar 2 02:48:18.445:   {
*Mar 2 02:48:18.445:     e164 : "14#14300"
*Mar 2 02:48:18.445:   },
*Mar 2 02:48:18.449:   srcInfo
*Mar 2 02:48:18.449:   {
*Mar 2 02:48:18.449:     e164 : "13#13300"
*Mar 2 02:48:18.449:   },
*Mar 2 02:48:18.449:   bandwidth 0640,
*Mar 2 02:48:18.449:   callReferenceValue 0224,
*Mar 2 02:48:18.449:   conferenceID '4B1F5E5D899000720000000005C067A4'H,
*Mar 2 02:48:18.449:   activeMC FALSE,
*Mar 2 02:48:18.449:   answerCall FALSE
*Mar 2 02:48:18.449: }
*Mar 2 02:48:18.449:25800CF7 00F00036 00300044 00360042 00410034 00430030 00300030
00300030
00300030 00310103 80470476 33010380 46046633 40028000 E04B1F5E 5D899000
72000000 0005C067 A400
29000CF7 40028000 0109350E 06B80077
value RasMessage ::= admissionConfirm :
*Mar 2 02:48:18.469: {
*Mar 2 02:48:18.469:   requestSeqNum 03320,
*Mar 2 02:48:18.469:   bandwidth 0640,
*Mar 2 02:48:18.469:   callModel direct :NULL,
*Mar 2 02:48:18.469:   destCallSignalAddress ipAddress :
*Mar 2 02:48:18.469:   {
*Mar 2 02:48:18.469:     ip '0109350E'H,
*Mar 2 02:48:18.469:     port 01720
*Mar 2 02:48:18.469:   },
*Mar 2 02:48:18.469:   irrFrequency 0120
*Mar 2 02:48:18.473: }
*Mar 2 02:48:18.473:value H323-UserInformation ::=
*Mar 2 02:48:18.481:{
*Mar 2 02:48:18.481: h323-uu-pdu
*Mar 2 02:48:18.481: {
*Mar 2 02:48:18.481:   h323-message-body setup :
*Mar 2 02:48:18.481:   {
*Mar 2 02:48:18.481:     protocolIdentifier { 0 0 8 2250 0 1 },
*Mar 2 02:48:18.481:     sourceInfo
*Mar 2 02:48:18.481:     {
*Mar 2 02:48:18.481:       terminal
*Mar 2 02:48:18.481:       {
*Mar 2 02:48:18.481:         }
*Mar 2 02:48:18.481:     },
*Mar 2 02:48:18.481:   },
*Mar 2 02:48:18.481: }

```

```

*Mar 2 02:48:18.481:          mc FALSE,
*Mar 2 02:48:18.481:          undefinedNode FALSE
*Mar 2 02:48:18.481:        },
*Mar 2 02:48:18.481:        activeMC FALSE,
*Mar 2 02:48:18.481:        conferenceID '4B1F5E5D899000720000000005C067A4'H,
*Mar 2 02:48:18.481:        conferenceGoal create :NULL,
*Mar 2 02:48:18.485:        callType pointToPoint :NULL,
*Mar 2 02:48:18.485:        sourceCallSignalAddress ipAddress :
*Mar 2 02:48:18.485:          {
*Mar 2 02:48:18.485:            ip '00000000'H,
*Mar 2 02:48:18.485:            port 00
*Mar 2 02:48:18.485:          }
*Mar 2 02:48:18.485:        }
*Mar 2 02:48:18.485:      }
*Mar 2 02:48:18.485:    }
*Mar 2 02:48:18.485:  }
*Mar 2 02:48:18.485:00800600 08914A00 0102004B 1F5E5D89 90007200 00000005 C067A400
0C070000
00000000 00
value H323-UserInformation ::=
*Mar 2 02:48:18.525:{
*Mar 2 02:48:18.525:  h323-uu-pdu
*Mar 2 02:48:18.525:  {
*Mar 2 02:48:18.525:    h323-message-body alerting :
*Mar 2 02:48:18.525:    {
*Mar 2 02:48:18.525:      protocolIdentifier { 0 0 8 2250 0 1 },
*Mar 2 02:48:18.525:      destinationInfo
*Mar 2 02:48:18.525:      {
*Mar 2 02:48:18.525:        mc FALSE,
*Mar 2 02:48:18.525:        undefinedNode FALSE
*Mar 2 02:48:18.525:      },
*Mar 2 02:48:18.525:      h245Address ipAddress :
*Mar 2 02:48:18.525:      {
*Mar 2 02:48:18.525:        ip '0109350E'H,
*Mar 2 02:48:18.525:        port 011050
*Mar 2 02:48:18.525:      }
*Mar 2 02:48:18.525:    }
*Mar 2 02:48:18.525:  }
*Mar 2 02:48:18.525:}
*Mar 2 02:48:18.525:value H323-UserInformation ::=
*Mar 2 02:48:22.753:{
*Mar 2 02:48:22.753:  h323-uu-pdu
*Mar 2 02:48:22.753:  {
*Mar 2 02:48:22.753:    h323-message-body connect :
*Mar 2 02:48:22.753:    {
*Mar 2 02:48:22.753:      protocolIdentifier { 0 0 8 2250 0 1 },
*Mar 2 02:48:22.753:      h245Address ipAddress :
*Mar 2 02:48:22.753:      {
*Mar 2 02:48:22.753:        ip '0109350E'H,
*Mar 2 02:48:22.753:        port 011050
*Mar 2 02:48:22.753:      },
*Mar 2 02:48:22.753:      destinationInfo
*Mar 2 02:48:22.753:      {
*Mar 2 02:48:22.753:        terminal
*Mar 2 02:48:22.753:        {
*Mar 2 02:48:22.753:          },
*Mar 2 02:48:22.757:        mc FALSE,
*Mar 2 02:48:22.757:        undefinedNode FALSE
*Mar 2 02:48:22.757:      },
*Mar 2 02:48:22.757:      conferenceID '4B1F5E5D899000720000000005C067A4'H
*Mar 2 02:48:22.757:    }
*Mar 2 02:48:22.757:  }
*Mar 2 02:48:22.757:}
*Mar 2 02:48:22.757:value H323-UserInformation ::=
*Mar 2 02:48:27.109:{
*Mar 2 02:48:27.109:  h323-uu-pdu

```

```

*Mar 2 02:48:27.109: {
*Mar 2 02:48:27.109:   h323-message-body releaseComplete :
*Mar 2 02:48:27.109:   {
*Mar 2 02:48:27.109:     protocolIdentifier { 0 0 8 2250 1 }
*Mar 2 02:48:27.109:   }
*Mar 2 02:48:27.109: }
*Mar 2 02:48:27.109:}
*Mar 2 02:48:27.109:value RasMessage ::= disengageRequest :
*Mar 2 02:48:27.117: {
*Mar 2 02:48:27.117:   requestSeqNum 03321,
*Mar 2 02:48:27.117:   endpointIdentifier "60D6BA4C00000001",
*Mar 2 02:48:27.117:   conferenceID '4B1F5E5D899000720000000005C067A4'H,
*Mar 2 02:48:27.121:   callReferenceValue 0224,
*Mar 2 02:48:27.121:   disengageReason normalDrop :NULL
*Mar 2 02:48:27.121: }
*Mar 2 02:48:27.121:3C0CF81E 00360030 00440036 00420041 00340043 00300030 00300030
00300030
00300031 4B1F5E5D 89900072 00000000 05C067A4 00E020
400CF8
value RasMessage ::= disengageConfirm :
*Mar 2 02:48:27.133: {
*Mar 2 02:48:27.133:   requestSeqNum 03321
*Mar 2 02:48:27.133: }
*Mar 2 02:48:27.133:value H323-UserInformation ::=
*Mar 2 02:48:27.133:{
*Mar 2 02:48:27.133: h323-uu-pdu
*Mar 2 02:48:27.133: {
*Mar 2 02:48:27.133:   h323-message-body releaseComplete :
*Mar 2 02:48:27.133:   {
*Mar 2 02:48:27.133:     protocolIdentifier { 0 0 8 2250 1 }
*Mar 2 02:48:27.133:   }
*Mar 2 02:48:27.133: }
*Mar 2 02:48:27.133:}
*Mar 2 02:48:27.133:05000600 08914A00 01
.

```

debug ras

Use the **debug ras** command to display the types and addressing of RAS messages sent and received. The debug output lists the message type using mnemonics defined in ITU-T specification H.225. Use the **no** form of this command to disable debug output.

debug ras
no debug ras

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Command Mode

Privileged EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.3(6)NA2.

Use the **debug ras** command to display the types and addressing of RAS messages sent and received. The debug output lists the message type using mnemonics defined in ITU-T specification H.225.

Sample Output

In the following output, gateway GW13.cisco.com sends a RAS registration request message (RRQ) to gatekeeper GK15.cisco.com at IP address 172.9.53.15. GW13.cisco.com then receives a registration confirmation (RCF) message from the gatekeeper. If there is no response, it could mean that the gatekeeper is offline or improperly addressed. If you receive a reject message (RRJ), it could mean that the gatekeeper is unable to handle another gateway or that the registration information is incorrect.

```
3640GW.13#debug ras
*Mar 13 19:53:34.231:   RASLib::ras_sendto:msg length 105 from
                        172.9.53.13:8658 to 1.9.53.15:1719
*Mar 13 19:53:34.231:   RASLib::RASSendRRQ:RRQ (seq# 36939) sent
                        to 172.9.53.15
*Mar 13 19:53:34.247:   RASLib::RASRecvData:successfully rcvd
                        message of length 105 from 172.9.53.15:1719
*Mar 13 19:53:34.251:   RASLib::RASRecvData:RCF (seq# 36939) rcvd
                        from [172.9.53.15:1719] on sock [0x6168356C
```

