

# Basic Two Zone Cisco Gateway-to-Gatekeeper Configuration

Document ID: 21063

## Contents

### Introduction

#### Before You Begin

- Conventions
- Prerequisites
- Components Used

#### Configure

- General Requirements for the Gateway-to-Gatekeeper Configuration
- Network Diagram
- Configurations

#### Verify

#### Troubleshoot

- Troubleshooting Commands

#### Related Information

## Introduction

This document studies a VoIP network with a two-zone topology managed by two Cisco Gatekeepers with one Cisco Gateway in each zone. The aim of this document is to provide a basic configuration that allows the user to avoid some known problems and create a reliable basis for the Gatekeeper-based network. This document includes background technical information on the configured features, design guidelines, and basic verification and troubleshooting strategies.

It is important to note that in the configuration below, the four routers are located on the same LAN. However, in your real topology, all devices can be in different parts of your network.

## Before You Begin

### Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

These configurations were tested with this equipment:

- Four Cisco 2600s with Cisco IOS<sup>®</sup> Software Release 12.2.8.5 ENTERPRISE PLUS/H323 MCM

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you work in a live network, ensure that you understand the potential impact of any command before you use it.

# Configure

In this section, you are presented with the information to configure the features described in this document.

**Note:** To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only) .

## General Requirements for the Gateway-to-Gatekeeper Configuration

There are several conditions to be met before the Gateway is able to get correct address resolution from the Gatekeeper.

There are two important points to be verified, as follows:

- All Gateways should be registered to corresponding Gatekeepers.
- All Gatekeepers should have the correct dial-plan.

### Registration

Successful registration is the first mandatory step. These additional factors should be taken into account:

- If the Gateway has Foreign Exchange Station (FXS) interfaces, then on the Plain Old Telephone Service (POTS) dial-peers, add the **no register e164** command. This avoids the problem with Gateway registration described in Cisco bug ID CSCdw60626 (registered customers only) . Instead of direct FXS port registration with e164 numbers, it is possible to add a zone prefix for the Gateway and to base the routing decisions on the zone prefixes.
- Usually, it is preferable to define a technology prefix for the Gateway. Although the presence of the technology prefixes predominantly affects the call routing, it is also desirable for reliable registration.

For more information on Gateway-to-Gatekeeper registration issues, refer to Troubleshooting Gatekeeper Registration Issues.

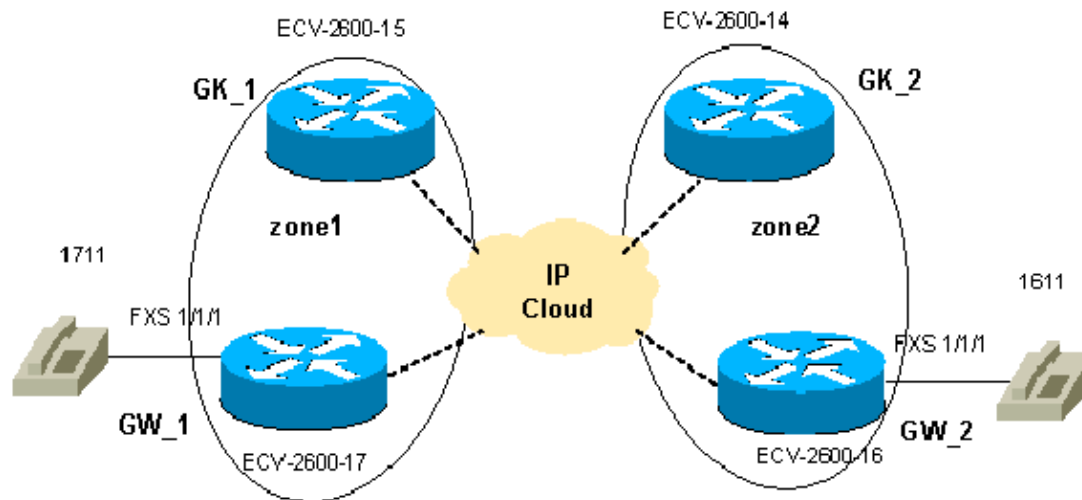
### Call Routing

- For reliable call routing, all Gateways should be registered with some technology prefix. The purpose of the technology prefix is to distinguish between different types of calls and corresponding types of Gateways. So, although it is possible to use the technology prefix for routing decisions, the better practice is to use the technology prefix to distinguish the type of call and route based on the zone prefixes. With this approach, all VoIP Gateways can be configured with the same technology prefix (for instance 1#\*, as in the example presented in this document).
- It is preferable to explicitly configure the primary Gateway for the zone prefix.
- Bind the H.323 Signaling to a specific IP address on the Cisco IOS Gateway or router. When the Cisco IOS Gateway has multiple active IP interfaces, some of the H.323 messages may be sourced from one IP address, and other parts of it may reference a different source address. The **h323-gateway voip bind srcaddr** command is necessary if the loopback interface is used to identify the Gateway, or there is a firewall and accounting servers in the network. This command was introduced in Cisco IOS Software Release 12.1.2T and is documented in Configuring H.323 Support for Virtual Interfaces.

For more information on Gatekeeper call routing, refer to Understanding Cisco IOS H.323 Gatekeeper Call Routing.

## Network Diagram

This document uses the network setup shown in the diagram below.



## Configurations

This document uses these configurations.

The verification of the configurations of the Gatekeeper and Gateway is an important part of troubleshooting Gateway-to-Gatekeeper problems. To simplify comprehension of the configurations, all unrelated configuration commands have been removed.

- GW\_1 – ECV-2600-17
- GW\_2 – ECV-2600-16
- GK\_1 ECV-2600-15
- GK\_2 ECV-2600-14

### GW\_1 – ECV-2600-17

```
IOS (tm) C2600 Software (C2600-JSX-M), Version 12.2(7a),  
RELEASE SOFTWARE (fcl)  
!  
hostname ECV-2610-17  
!  
!  
interface Ethernet0/0  
 ip address 10.52.218.49 255.255.255.0  
h323-gateway voip interface  
  
!---- This command enables VoIP GW functions on the interface.  
  
h323-gateway voip id gk-zone1.test.com ipaddr 10.52.218.47  
 1718  
  
!---- This command defines the GK this GW works with.  
  
h323-gateway voip h323-id gw_1  
  
!---- This command defines the GW alias for the GK.  
  
h323-gateway voip tech-prefix 1#  
  
!---- It is desirable to have tech prefix on the GW for
```

```

!---- reliable registration and call routing.

h323-gateway voip bind srcaddr 10.52.218.49

!---- This command is not necessary in this simple topology,
!---- but for complex networks, it is recommended to use it.

    ??
!
voice-port 1/1/0
!
voice-port 1/1/1
!
!
dial-peer voice 1 voip
 destination-pattern 16..
 session target ras

!---- All IP addresses for the destination pattern 16.. should
!---- be resolved through the requests to the GK.

!
dial-peer voice 2 pots
 destination-pattern 1711
 port 1/1/1
no register e164

!---- This command prevents registration of this number with
!---- the GK. The GW is registered with the GK with this alias only.

!
gateway
!
end

```

### GW\_2 – ECV-2600-16

```

!
hostname ECV-2610-16
!
!
interface Ethernet0/0
 ip address 10.52.218.48 255.255.255.0
h323-gateway voip interface
h323-gateway voip id gk-zone2.test.com ipaddr 10.52.218.46 1718

h323-gateway voip h323-id gw_2
h323-gateway voip tech-prefix 1#
h323-gateway voip bind srcaddr 10.52.218.48
!
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dial-peer voice 1 voip
 destination-pattern 17..
 session target ras
!
dial-peer voice 2 pots
 destination-pattern 1611
 port 1/1/1
no register e164
!

```

```
gateway
!  
!  
end
```

### GK\_1 ECV-2600-15

```
!  
hostname ECV-2610-15  
!  
interface Ethernet0/0  
 ip address 10.52.218.47 255.255.255.0  
!  
gatekeeper  
zone local gk-zone1.test.com test.com 10.52.218.47  
  
!---- This command defines the local zone. The GK name and  
!---- zone name have the same meaning.  
  
zone remote gk-zone2.test.com test.com 10.52.218.46 1719  
  
!---- This command defines the name of the remote GK (zone).  
  
zone prefix gk-zone2.test.com 16..  
  
!---- This command explicitly defines the number length with  
!---- the number of dots.  
  
zone prefix gk-zone1.test.com 17.. gw-priority 10 gw_1  
  
!---- This command explicitly defines which GW handles  
!---- calls for 17.. numbers that could be done for the  
!---- local zones only.  
  
gw-type-prefix 1#* default-technology  
  
!---- This command defines the default technology prefix  
!---- that is necessary for routing decisions.  
  
no shutdown  
  
!--- This command turns the service up.  
  
!  
end
```

### GK\_2 ECV-2600-14

```
!  
hostname ECV-2610-14  
!  
interface Ethernet0/0  
 ip address 10.52.218.46 255.255.255.0  
!  
gatekeeper zone local gk-zone2.test.com test.com 10.52.218.46  
zone remote gk-zone1.test.com test.com 10.52.218.47 1719  
zone prefix gk-zone2.test.com 16.. gw-priority 10 gw_2  
zone prefix gk-zone1.test.com 17..  
gw-type-prefix 1#* default-technology  
no shutdown  
  
!  
end
```

# Verify

This section provides information you can use to confirm your configuration works properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

**Note:** Before you attempt any **debug** commands, refer to Important Information on Debug Commands. For more information on the commands below, see the Troubleshooting Commands section of this document.

- **show gateway** Displays the Gateway registration status.
- **show gatekeeper endpoints** Displays all Gateways registered to the Gatekeeper.
- **show gatekeeper zone prefix** Displays all zone prefixes configured on the Gatekeeper.
- **show gatekeeper call** Shows active calls processed by the Gatekeeper.
- **debug h225 asn1** Displays H225 (Registration, Admission, and Status [RAS] and Q931 call setup) messages.
- **debug cch323 h225** Displays H225 call setup messages.
- Troubleshooting and Debugging VoIP Call Basics
- VoIP Debug Commands
- Cisco IOS Voice, Video, and Fax Command Reference, Release 12.2

# Troubleshoot

This section provides information you can use to troubleshoot your configuration.

## Troubleshooting Commands

To troubleshoot, check several vital points:

- All Gateways are to be registered with the corresponding Gatekeepers.
- Gateways should have the correct dial plan (dial-peers configured).
- Gatekeepers should have the correct dial plan (zone prefixes configured).

The steps described in Troubleshooting and Debugging VoIP Call Basics complement the output of **debug** and **show** commands related to Gateway-to-Gatekeeper interaction and should be used to highlight the voice problems related to other Cisco IOS subsystems. The sample outputs from **show** commands highlight the steps above, and **debug** output shows the sequence of RAS and H225 messages at all four routers.

**Note:** The **debug h225 asn1** command generates a very large output, so it should be used with great care. Some unnecessary output has been deleted from the **debug** commands below.

**Note:** Before you issue **debug** commands, refer to Important Information on Debug Commands.

```
!--- Check the GW registration on the GW.
ECV-2610-17#show gateway
  Gateway gw_1 is registered to Gatekeeper gk-zone1.test.com
  ??
Alias list (CLI configured)
H323-ID gw_1
Alias list (last RCF)
H323-ID gw_1
  ??
H323 resource thresholding is DisabledECV-2610-17#
```

-----  
*!--- And on the corresponding GK.*

??

ECV-2610-15#**show gatek en**

GATEKEEPER ENDPOINT REGISTRATION

=====

CallSignalAddr Port RASSignalAddrPort Zone Name Type F

-----

10.52.218.49 1720 10.52.218.4951194 **gk-zone1.test.com** VOIP-GW

H323-ID: **gw\_1**

Total number of active registrations = 1

ECV-2610-15#

-----  
??

*!--- The same for the second GW.*

ECV-2610-16#**show gateway**

**Gateway gw\_2 is registered to Gatekeeper gk-zone2.test.com ??**

Alias list (CLI configured)

H323-ID gw\_2

Alias list (last RCF)

H323-ID gw\_2

?? H323 resource thresholding is Disabled

ECV-2610-16#

-----??

*!--- And the second corresponding GK.*

ECV-2610-14#**show gatek en**

GATEKEEPER ENDPOINT REGISTRATION

=====

CallSignalAddr Port RASSignalAddr Port Zone Name Type F

-----

10.52.218.48 1720 10.52.218.48 52080 **gk-zone2.test.com** VOIP-GW

H323-ID:

**gw\_2**

Total number of active registrations = 1 ??

ECV-2610-14#

-----  
*!--- To check the dial plan on the GKs:*

??

ECV-2610-15#**show gatek zone pr**

ZONE PREFIX TABLE

=====

GK-NAME E164-PREFIX

-----

**gk-zone2.test.com 16..**

**gk-zone1.test.com 17..??**

ECV-2610-15#

ECV-2610-15#

*!--- All configured prefixes should be seen in the zone list.*

-----??

*!--- To check the dial plan on the GKs:*

```
ECV-2610-14#  
ECV-2610-14#show gatek zone pr  
      ZONE PREFIX TABLE
```

```
=====
```

```
GK-NAME E164-PREFIX
```

```
-----  
gk-zone2.test.com 16..  
gk-zone1.test.com 17..??
```

```
ECV-2610-14#
```

```
-----??
```

```
ECV-2610-15#show gatekeeper call  
Total number of active calls = 1.
```

```
      GATEKEEPER CALL INFO
```

```
=====
```

```
LocalCallIDAge(secs) BW
```

```
5-0 1 64(Kbps)
```

```
Endpt(s): Alias E.164Addr CallSignalAddr Port RASignalAddr Port
```

```
src EP: gw_2 1611 10.52.218.48 1720 10.52.218.48 59067
```

```
dst EP: gw_1 1711 10.52.218.49 1720 10.52.218.49 58841??
```

```
ECV-2610-15#
```

```
-----??
```

*!--- The conversation between the GW and the GK consists of  
!--- exchange RAS messages. Here are two messages that show  
!--- successful registration of the GW to the GK.*

```
ECV-2610-17#
```

```
ECV-2610-17#debug h225 asn1
```

```
H.225 ASN1 Messages debugging is on
```

```
ECV-2610-17#
```

```
*Mar 2 07:45:53: RAS OUTGOING PDU ::=
```

*!--- The GW sends a RAS registration request message to the GK.*

```
value RasMessage ::= registrationRequest :
```

```
{  
  requestSeqNum 93  
  protocolIdentifier { 0 0 8 2250 0 2 }  
  discoveryComplete FALSE  
  callSignalAddress  
  {  
  }  
  rasAddress  
  {  
    ipAddress :  
    {  
      ip '0A34DA31'H  
      port 57733  
    }  
  }  
  terminalType  
  {  
    mc FALSE  
    undefinedNode FALSE  
  }  
  gatekeeperIdentifier {"gk-zone1.test.com"}  
  endpointVendor  
  {
```



```

vendor
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
}
timeToLive 60
keepAlive TRUE
endpointIdentifier {"8215266C0000000F"}
    willSupplyUUIEs FALSE
}

*Mar 2 07:45:53:
*Mar 2 07:45:53: RAS INCOMING PDU ::=

!--- The GK accepts the registration request and replies with
!--- a confirmation.

value RasMessage ::= registrationConfirm :
{
requestSeqNum 93
protocolIdentifier { 0 0 8 2250 0 2 }
callSignalAddress
{
}
gatekeeperIdentifier {"gk-zone1.test.com"}
    endpointIdentifier {"8215266C0000000F"}
timeToLive 60
    willRespondToIRR FALSE
}??

-----??

!--- The incoming H225 call setup message from the remote GW.
!--- The example is the debug cch323 h225 command.

ECV-2610-17# debug cch323 h225
*Mar 2 07:46:03: cch323_h225_receiver: received msg of type
SETUPIND_CHOSEN

*Mar 2 07:46:03: cch323_h225_setup_ind: callingNumber[]
calledNumber[1711]

*Mar 2 07:46:03: cch323_h225_setup_ind--calling IE NOT present
*Mar 2 07:46:03:==== PI in cch323_h225_setup_ind = 0??

*Mar 2 07:46:03: Receive: infoXCap 0??

*Mar 2 07:46:03: Receive infoXCap ccb 0??

*Mar 2 07:46:03: src address = 10.52.218.49 of SETUPIND_CHOSEN
*Mar 2 07:46:03: dest address = 10.52.218.47 of SETUPIND_CHOSEN??

*Mar 2 07:46:03: cch323_run_h225_sm: received event
H225_EVENT_FAST_SETUP_IND while

at state H225_IDLE??

*Mar 2 07:46:03: cch323_run_h225_sm: Setup ccb 0x821FCE98 callID
0xFFFFFFFF
*Mar 2 07:46:03: cch323_h225_act_fastStartSetupInd: codec match = 1
*Mar 2 07:46:03: cch323_rtp_set_non_rtp_call: Non-RTP call end
*Mar 2 07:46:03: H.225 SM: changing from H225_IDLE state to
H225_REQ_WAIT_FOR_ARQ

```

state for callID FFFFFFFF??

-----  
*!--- Now the example of the debug h225 asn1  
!--- command from all four routers.  
!--- The messages are sent from the originating GW.*

ECV-2610-16#debug h225 asn1  
**H.225 ASN1 Messages debugging is on**  
ECV-2610-16#

*!--- The GW\_2 initiates a call to 1711 phone located on GW\_1.  
!--- Here is the messages that show the process on GW\_2:??*

\*Mar 2 14:28:08.824: RAS OUTGOING PDU ::=

*!--- The GW\_2 asks gk-zone2 to resolve the e164 number 1711 to IP  
!--- address.*

```
value RasMessage ::= admissionRequest :
{
  requestSeqNum 3091
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8217FB5000000001"}
  destinationInfo
  {
    e164 : "1711"
  }
  srcInfo
  {
    e164 : "1611",
    h323-ID : {"gw_2"}
  }
  bandwidth 640
  callReferenceValue 8
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
      t35Extension 0
      manufacturerCode 18
    }
    data '80000008200A1046585320312F312F31'H
  }
  conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  activeMC FALSE
  answerCall FALSE
  canMapAlias TRUE
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  willSupplyUUIEs FALSE
}
??
```

\*Mar 2 14:28:08.960: RAS INCOMING PDU ::=

*!--- The gk-zone2 notifies GW\_2 that the request is in progress as it  
!--- is forwarded to the other gk-zone1 and is not processed locally.*

```

??
value RasMessage ::= requestInProgress :
{
!--- Note the sequence numbers in the request equal the number in
!--- the reply.

requestSeqNum 3091
delay 9000
}

??
*Mar 2 14:28:09.169: RAS INCOMING PDU ::=

!--- The gk-zone2 grants permission to start call and resolves the
!--- e164 number 1711 to IP address of GW_1.

value RasMessage ::= admissionConfirm :
{
!--- The sequence numbers in the request equal the number in the reply.

requestSeqNum 3091
bandWidth 640
callModel direct : NULL
destCallSignalAddress ipAddress :
{
ip '0A34DA31'H

!--- The IP address 10.52.218.49 of GW_1.

port 1720
}
irrFrequency 240
destinationInfo
{
e164 : "1711"
}
willRespondToIRR FALSE
uuiesRequested
{
setup FALSE
callProceeding FALSE
connect FALSE
alerting FALSE
information FALSE
releaseComplete FALSE
facility FALSE
progress FALSE
empty FALSE
}
}

*Mar 2 14:28:09.193: H225 NONSTD OUTGOING PDU ::=

value H323_UU_NonStdInfo ::=
{
version 0
progIndParam progIndIEinfo :
{
progIndIE '00000003'H
}
}
}

```

\*Mar 2 14:28:09.197: H225.0 OUTGOING PDU ::=

*!--- The GW\_2 now can place H323 (q931) call setup message directly  
!--- to GW\_1.*

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body setup :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      sourceAddress
      {
        h323-ID : {"gw_2"}
      }
      sourceInfo
      {
        gateway
        {
          protocol
          {
            voice :
            {
              supportedPrefixes
              {??

              prefix e164 : "1#"
            }
          }
        }
      }
      mc FALSE
      undefinedNode FALSE
    }
    destinationAddress
    {
      e164 : "1711"
    }
    activeMC FALSE
    conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
    conferenceGoal create : NULL
    callType pointToPoint : NULL
    sourceCallSignalAddress ipAddress :
    {
      ip '0A34DA30'H
      port 11001
    }
    callIdentifier
    {
      guid 'F748749F163011CC801DC5F8EEB46E69'H
    }
    fastStart
    {
      '0000000D4001800A040001000A34DA3043F3'H,
      '400000060401004D40018011140001000A34DA30...'H
    }
    mediaWaitForConnect FALSE
    canOverlapSend FALSE
  }
  h245Tunneling FALSE
}
```

```
nonStandardControl
{??

    {
nonStandardIdentifier h221NonStandard :
{
    t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data 'C00100028006000400000003'H
}
    }
}
}
```

\*Mar 2 14:28:09.573: **H225.0 INCOMING PDU** ::=

*!--- The GW\_1 replies with an H323 (q931) callProceeding message.*

```
value H323_UserInformation ::=
{
h323-uu-pdu
{
h323-message-body callProceeding :
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
{
mc FALSE
undefinedNode FALSE
}
callIdentifier
{
guid 'F748749F163011CC801DC5F8EEB46E69'H
}
fastStart
{
'0000000D40018011140001000A34DA314942000A...'H,
'400000060401004D40018011140001000A34DA30...'H
}
}
h245Tunneling FALSE
}
}
```

\*Mar 2 14:28:09.766: **H225.0 INCOMING PDU** ::=

*!--- The GW\_1 sends an H323 (q931) call Progress message.*

```
value H323_UserInformation ::=
{
h323-uu-pdu
{
h323-message-body progress :
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
{
mc FALSE
undefinedNode FALSE
}
callIdentifier
```



```
nonStandardIdentifier h221NonStandard :
{
    t35CountryCode 181
    t35Extension 0
    manufacturerCode 18
}
data 'C00100028006000400000002'H
}
}
}
```

\*Mar 2 14:28:11.909: **show call active voice**  
**Total call-legs: 2**

??GENERIC:  
SetupTime=13848499 ms  
Index=1  
**PeerAddress=1611**  
PeerSubAddress=  
PeerId=2  
PeerIfIndex=11  
LogicalIfIndex=8  
ConnectTime=13849192  
**CallDuration=00:00:19**  
**CallState=4**

*!--- This means the call is active.*

CallOrigin=2  
ChargedUnits=0  
InfoType=2  
**TransmitPackets=442**  
**TransmitBytes=8840**  
**ReceivePackets=1104**  
**ReceiveBytes=22080**

*!--- This shows that there is two-way voice for this call leg.*  
*!--- 0 values a problem.*

**TELE:**

*!--- The call is outgoing and started from the PSTN. That is why*  
*!--- TELE: is first in the output.*

ConnectionId=[0xF748749F 0x163011CC 0x801CC5F8 0xEEB46E69]  
IncomingConnectionId=[0xF748749F 0x163011CC 0x801CC5F8  
0xEEB46E69]  
TxDuration=22100 ms  
VoiceTxDuration=2209 ms  
FaxTxDuration=0 ms  
**CoderTypeRate=g729r8**  
NoiseLevel=-48  
ACOMLevel=2  
OutSignalLevel=-57  
InSignalLevel=-53  
InfoActivity=2  
ERLLevel=16  
SessionTarget=  
ImgPages=0  
GENERIC:  
SetupTime=13848887 ms  
Index=1  
**PeerAddress=1711**

```
PeerSubAddress=
PeerId=1PeerIf
Index=13
LogicalIfIndex=0
ConnectTime=13849185
CallDuration=00:00:20
CallState=4
CallOrigin=1
ChargedUnits=0
InfoType=2
TransmitPackets=1038
TransmitBytes=20760
ReceivePackets=488
ReceiveBytes=9760
VOIP:
ConnectionId[0xF748749F 0x163011CC 0x801CC5F8 0xEEB46E69]
IncomingConnectionId[0xF748749F 0x163011CC 0x801CC5F8
0xEEB46E69]
RemoteIPAddress=10.52.218.49RemoteUDPPort=18754
```

*!--- The signaling and RTP stream IP addresses.*

```
RemoteSignallingIPAddress=10.52.218.49
RemoteSignallingPort=1720
RemoteMediaIPAddress=10.52.218.49
RemoteMediaPort=18754
RoundTripDelay=5 ms
SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
FastConnect=TRUE
```

```
Separate H245 Connection=FALSE
```

```
H245 Tunneling=FALSE
```

```
SessionProtocol=cisco
SessionTarget=ras
OnTimeRvPayout=6630
GapFillWithSilence=0 ms
GapFillWithPrediction=0 ms
GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPayoutDelay=70 ms
LoWaterPayoutDelay=50 ms
ReceiveDelay=50 ms
LostPackets=0
EarlyPackets=0
LatePackets=0
```

*!--- The DSP statistics.*

```
VAD = enabled
CoderTypeRate=g729r8
CodecBytes=20Total call-legs: 2
```

```
ECV-2610-16#
ECV-2610-16#
```

```
u all
All possible debugging has been turned off
```

*!--- The following messages shows the call disconnect  
!--- process at the GW\_2.*



```
ECV-2610-16#deb h225 asn1
H.225 ASN1 Messages debugging is on:
*Mar 2 14:29:52.017: H225.0 INCOMING PDU ::=
```

```
!-- The GW_1 sends H323 (q931) Release complete message.
```

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body releaseComplete :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      callIdentifier
      {
        guid 'F748749F163011CC801DC5F8EEB46E69'H
      }
    }
    h245Tunneling FALSE
  }
}
```

```
*Mar 2 14:29:52.025: H225.0 OUTGOING PDU ::=
```

```
!-- The GW_2 replies with the H323 (q931) releaseComplete
!-- message.
```

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body releaseComplete :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      callIdentifier
      {
        guid 'F748749F163011CC801DC5F8EEB46E69'H
      }
    }
    h245Tunneling FALSE
  }
}
```

```
*Mar 2 14:29:52.041: RAS OUTGOING PDU ::=
```

```
!-- The GW_2 notifies GK-2 that the call is complete.
```

```
value RasMessage ::= disengageRequest :
{
  requestSeqNum 3095
  endpointIdentifier {"8217FB5000000001"}
  conferenceID 'F748749F163011CC801DC5F8EEB46E69'H
  callReferenceValue 8
  disengageReason normalDrop : NULL
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  answeredCall FALSE
}
```

\*Mar 2 14:29:52.090: RAS INCOMING PDU ::=

*!--- The GK-2 confirms the message.*

```
value RasMessage ::= disengageConfirm :
{
  requestSeqNum 3095
}
```

u all

All possible debugging has been turned off

-----  
*!--- The debug output from the GK-2.*

ECV-2610-14#debug h225 asnl

**H.225 ASN1 Messages debugging is on**

ECV-2610-14#

Mar 2 14:28:20.952:

Mar 2 14:28:20.952: RAS INCOMING PDU ::=

*!--- The GW\_2 asks permission to place the call.*

*!--- Now it is incoming RAS PDU as it is on the GK-2, but the*

*!--- same sequence number.*

```
value RasMessage ::= admissionRequest :
{
  requestSeqNum 3091
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8217FB5000000001"}
  destinationInfo
  {
    e164 : "1711"
  }
  srcInfo
  {
    e164 : "1611",
    h323-ID: {"gw_2"}
  }
  bandwidth 640
  callReferenceValue 8
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
      t35Extension 0
      manufacturerCode 18
    }
  }
  data '80000008200A1046585320312F312F31'H
  conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  activeMC FALSE
  answerCall FALSE
  canMapAlias TRUE
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  willSupplyUUIEs FALSE
  }??
```

Mar 2 14:28:20.992: RAS OUTGOING PDU ::=

*!--- The GK-2 asks GK-1 to resolve the Number for the remote*

*!--- zone.*

```
value RasMessage ::= locationRequest :
{
  requestSeqNum 1026
  destinationInfo
  {
    e164 : "1711"
  }
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
      t35Extension 0
      manufacturerCode 18
    }
    data '8284901100F748749F163011CC801DC5F8EEB46E...'H
  }
  replyAddress ipAddress :
  {
    ip '0A34DA2E'H
    port 1719
  }
  sourceInfo
  {
    h323-ID : {"gk-zone2.test.com"}
  }
  canMapAlias TRUE
}
```

Mar 2 14:28:21.024: RAS OUTGOING PDU ::=

*!--- The GK-2 notifies GW\_2 that the call is processing.*

```
value RasMessage ::= requestInProgress :
{
  requestSeqNum 3091
  delay 9000
}
```

Mar 2 14:28:21.157:

Mar 2 14:28:21.157: **RAS INCOMING** PDU ::=

*!--- The GK-1 replies to GK-2 with the permission.*

```
value RasMessage ::= locationConfirm :
{
  requestSeqNum 1026
  callSignalAddress ipAddress :
  {
    ip'0A34DA31'H
    port 1720
  }
  rasAddress ipAddress :
  {
    ip '0A34DA31'H
    port 55679
  }
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
    }
  }
}
```

```
t35Extension 0
manufacturerCode 18
}
data '0001400300670077005F0031200067006B002D00...'H
}
  destinationInfo
  {
    e164 : "1711"
  }
  destinationType
  {
    gateway
    {
    protocol
    {
      voice :
      {
        supportedPrefixes
        {
          }
        }
      }
    }
  }
  mc FALSE
  undefinedNode FALSE
}
}
```

Mar 2 14:28:21.209: **RAS OUTGOING** PDU::=

*!--- The GK-2 replies to GW\_2 with the permission to place  
!--- the call.*

```
value RasMessage ::= admissionConfirm :
{
  requestSeqNum 3091
  bandwidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
  {
    ip '0A34DA31'H
    port 1720
  }
  irrFrequency 240
  destinationInfo
  {
    e164 : "1711"
  }
  willRespondToIRR FALSE
  uuiesRequested
  {
    setup FALSE
    callProceeding FALSE
    connect FALSE
    alerting FALSE
    information FALSE
    releaseComplete FALSE
    facility FALSE
    progress FALSE
    empty FALSE
  }
}
```

ECV-2610-14#u all

All possible debugging has been turned off

ECV-2610-14#debug h225 asn1

**H.225 ASN1 Messages debugging is on**

Mar 2 14:30:04.145: RAS INCOMING PDU ::=

*!--- The GK-2 gets notification from GW\_2 that the call  
!--- has ended.*

```
value RasMessage ::= disengageRequest :
{
  requestSeqNum 3095
  endpointIdentifier {"8217FB5000000001"}
    conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  callReferenceValue 8
  disengageReason normalDrop : NULL
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  answeredCall FALSE
}
```

Mar 2 14:30:04.157: RAS OUTGOING PDU ::=

```
value RasMessage ::= disengageConfirm :
{
  requestSeqNum 3095
}
```

ECV-2610-14#u all

All possible debugging has been turned off

ECV-2610-14#

-----  
*!--- The debug output from the GK-2.*

ECV-2610-15#ECV-2610-15#debug h225 asn1

**H.225 ASN1 Messages debugging is on**

\*Mar 2 14:28:14.690:

\*Mar 2 14:28:14.694:

RAS INCOMING PDU ::=

*!--- The request from the GK-2.*

```
value RasMessage ::= locationRequest :
{
  requestSeqNum 1026
  destinationInfo
  {
    e164 : "1711"
  }
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard:
    {
      t35CountryCode 181
      t35Extension 0
        manufacturerCode 18
    }
  }
  data '8284901100F748749F163011CC801DC5F8EEB46E...'H
}
replyAddress ipAddress :
```

```
{
ip '0A34DA2E'H
port 1719
}
sourceInfo
{
h323-ID : {"gk-zone2.test.com"}
}
canMapAlias TRUE
}
```

\*Mar 2 14:28:14.754: RAS OUTGOING PDU ::=

*!--- The reply from the GK-1 to GK-2.*

```
value RasMessage ::= locationConfirm :
{
requestSeqNum 1026
callSignalAddress ipAddress :
{
ip '0A34DA31'H
port 1720
}
rasAddress ipAddress :
{
ip '0A34DA31'H
port 55679
}
nonStandardData
{
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data '0001400300670077005F0031200067006B002D00...'H
}
destinationInfo
{
e164 : "1711"
}
destinationType
{
gateway
{
protocol
{
voice :
{
supportedPrefixes
{
}
}
}
}
}
mc FALSE
undefinedNode FALSE
}
}
```

\*Mar 2 14:28:15.159: RAS INCOMING PDU ::=

*!--- The GW\_1 asks GK-1 for permission to accept the call.*

```
value RasMessage ::= admissionRequest :
{
  requestSeqNum 101
  callType pointToPoint : NULL
  callModel direct : NULL
  endpointIdentifier {"8261828000000003"}
    destinationInfo
  {
    e164 : "1711"
  }
    srcInfo
  {
    e164 : "1611",
    h323-ID: {"gw_2"}
  }
  srcCallSignalAddress ipAddress:
  {
    ip '0A34DA30'H
    port 1100
  }
  bandwidth 640
  callReferenceValue 7
  nonStandardData
  {
    nonStandardIdentifier h221NonStandard :
    {
      t35CountryCode 181
      t35Extension 0
      manufacturerCode 18
    }
  }
  data '80000008200A1046585320312F312F31'H
  }
  conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  activeMC FALSE
  answerCall TRUE
  canMapAlias TRUE
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  willSupplyUUIEs FALSE
  }
}
```

\*Mar 2 14:28:15.191: RAS OUTGOING PDU ::=

*!--- The permission is granted.*

```
value RasMessage ::= admissionConfirm :
{
  requestSeqNum 101
  bandwidth 640
  callModel direct : NULL
  destCallSignalAddress ipAddress :
  {
    ip '0A34DA31'H
    port 1720
  }
  irrFrequency 240
  willRespondToIRR FALSE
    uuiesRequested
  {
    setup FALSE
    callProceeding FALSE
    connect FALSE
  }
}
```

```
alerting FALSE
information FALSE
releaseComplete FALSE
facility FALSE
progress FALSE
empty FALSE
    }
}
```

ECV-2610-15#

ECV-2610-15#show gatek call

**Total number of active calls = 1.**

GATEKEEPER CALL INFO

=====

LocalCallID Age(secs) BW

7-63391

33 64(Kbps)

Endpt(s): Alias E.164Addr CallSignalAddr Port RASSignalAddr Port

**src** EP: gw\_2 1611 10.52.218.48 1720 10.52.218.48 59067

**dst** EP: gw\_1 1711 10.52.218.49 1720 10.52.218.49 58841

ECV-2610-15#ECV-2610-15#u all

**All possible debugging has been turned off**

ECV-2610-15#debug h225 asn1

**H.225 ASN1 Messages debugging is on**

\*Mar 2 14:29:57.767: RAS INCOMING PDU ::=

*!--- The GK-1 gets notification from GW\_1 that the call has ended.*

value RasMessage ::= **disengageRequest** :

```
{
  requestSeqNum 105
  endpointIdentifier {"8261828000000003"}
  conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  callReferenceValue 7
  disengageReason normalDrop : NULL
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  answeredCall TRUE
}
```

\*Mar 2 14:29:57.779: RAS OUTGOING PDU ::=

*!--- The GK-1 confirms the message.*

value RasMessage ::= **disengageConfirm** :

```
{
  requestSeqNum 105
}
```

ECV-2610-15#u all

**All possible debugging has been turned off**

*!--- The debugs must always be turned off when the collection  
!--- is completed.*

-----  
*!--- The debugs at the terminating gateway GW\_1.*

ECV-2610-17#

ECV-2610-17#debug h225 asn1

**H.225 ASN1 Messages debugging is on**



\*Mar 1 11:02:27:

\*Mar 1 11:02:27: **H225.0 INCOMING PDU** ::=

*!--- The first message is the H225 call setup from GW\_2.*

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body setup :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      sourceAddress
      {
        h323-ID : {"gw_2"}
      }
      sourceInfo
      {
        gateway
      }
      protocol
      {
        voice :
        {
          supportedPrefixes
          {??
            {
              prefix e164 : "1#"
            }
          }
        }
      }
      mc FALSE
      undefinedNode FALSE
    }
    destinationAddress
    {
      e164 : "1711"
    }
    activeMC FALSE
    conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
    conferenceGoal create : NULL
    callType pointToPoint : NULL
    sourceCallSignalAddress ipAddress :
    {
      ip '0A34DA30'H
      port 11001
    }
    callIdentifier
    {
      guid 'F748749F163011CC801DC5F8EEB46E69'H
    }
    fastStart
    {
      '0000000D4001800A040001000A34DA3043F3'H,
      '400000060401004D40018011140001000A34DA30...'H
    }
    mediaWaitForConnect FALSE
    canOverlapSend FALSE
  }
  h245Tunneling FALSE
  nonStandardControl
  {
    ??
  }
}
```

```
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data 'C00100028006000400000003'H
}
}
}
```

\*Mar 1 11:02:27: RAS OUTGOING PDU ::=

*!--- The GW\_1 asks GK-1 for permission to accept the call.*

```
value RasMessage ::= admissionRequest :
{
requestSeqNum 101
callType pointToPoint : NULL
callModel direct : NULL
endpointIdentifier {"8261828000000003"}
destinationInfo
{
e164: "1711"
}
srcInfo
{
e164 : "1611",
h323-ID : {"gw_2"}
}
srcCallSignalAddress ipAddress:
{
ip '0A34DA30'H
port 11001
}
bandwidth 640
callReferenceValue 7
nonStandardData
{
nonStandardIdentifier h221NonStandard :
{
t35CountryCode 181
t35Extension 0
manufacturerCode 18
}
data '80000008200A1046585320312F312F31'H
}
conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
activeMC FALSE
answerCall TRUE
canMapAlias TRUE
callIdentifier
{
guid 'F748749F163011CC801DC5F8EEB46E69'H
}
willSupplyUUIEs FALSE
}
```

\*Mar 1 11:02:27: \*Mar 1 11:02:27: RAS INCOMING PDU ::=

*!--- The permission is granted.*

```
value RasMessage ::= admissionConfirm:
{
requestSeqNum 101
```

```
bandwidth 640
callModel direct: NULL
destCallSignalAddress ipAddress :
{
ip '0A34DA31'H
    port 1720
}
irrFrequency 240
willRespondToIRR FALSE
    uuiesRequested
{
setup FALSE
callProceeding FALSE
    connect FALSE
alerting FALSE
information FALSE
releaseComplete FALSE
facility FALSE
progress FALSE
empty FALSE
    }
}
```

\*Mar 1 11:02:27: **H225.0 OUTGOING PDU ::=**

*!-- The GW\_1 replies to the GW-2 with the callProceeding message.*

```
value H323_UserInformation::=
{
h323-uu-pdu
{
h323-message-body callProceeding:
{
protocolIdentifier { 0 0 8 2250 0 2 }
    destinationInfo
{
mc FALSE
undefinedNode FALSE
}
callIdentifier
{
guid 'F748749F163011CC801DC5F8EEB46E69'H
}
fastStart
{
'0000000D40018011140001000A34DA314942000A...'H,
    '400000060401004D40018011140001000A34DA30...'H
}
}
    h245Tunneling FALSE
}
}
```

\*Mar 1 11:02:27: **H225.0 OUTGOING PDU ::=**

*!-- The call Progress follows.*

```
value H323_UserInformation::=
{
h323-uu-pdu
{
h323-message-body progress:
{
protocolIdentifier { 0 0 8 2250 0 2 }
destinationInfo
```



```
{
??{
    nonStandardIdentifier h221NonStandard :
    {
    t35CountryCode 181
    t35Extension 0
    manufacturerCode 18
    }
    data 'C00100028006000400000002'H
    }
    }
}
```

ECV-2610-17#u all

**All possible debugging has been turned off**

ECV-2610-17#

ECV-2610-17#debug h225 asn1

**H.225 ASN1 Messages debugging is on**

ECV-2610-17#

\*Mar 1 11:04:10: **H225.0 OUTGOING PDU** ::=

*!--- The GW\_1 drops the call.*

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body releaseComplete :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      callIdentifier
      {
        guid 'F748749F163011CC801DC5F8EEB46E69'H
      }
    }
    h245Tunneling FALSE
  }
}
```

??\*Mar 1 11:04:10: **RAS OUTGOING PDU** ::=

*!--- The GW\_1 notifies GK-1 that the call has ended.*

```
value RasMessage ::= disengageRequest :
{
  requestSeqNum 105
  endpointIdentifier {"8261828000000003"}
  conferenceID 'F748749F163011CC801CC5F8EEB46E69'H
  callReferenceValue 7
  disengageReason normalDrop : NULL
  callIdentifier
  {
    guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
  answeredCall TRUE
}
```

\*Mar 1 11:04:10: **H225.0 INCOMING PDU** ::=

*!--- The GW\_2 drops the call from its side.*

```
value H323_UserInformation ::=
{
  h323-uu-pdu
```

```
{
h323-message-body releaseComplete :
  {
protocolIdentifier { 0 0 8 2250 0 2 }
callIdentifier
  {
guid 'F748749F163011CC801DC5F8EEB46E69'H
  }
}
h245Tunneling FALSE
}
}

*Mar 1 11:04:10: RAS INCOMING PDU ::=


!--- The GK-1 confirms the message.

value RasMessage ::= disengageConfirm :
{
requestSeqNum 105
}

u all
All possible debugging has been turned off

!--- The debugs must always be turned off when the collection
!--- is completed.
```

## Related Information

- [Understanding H.323 Gatekeepers](#)
- [Troubleshooting Gatekeeper Registration Issues](#)
- [Understanding Cisco IOS H.323 Gatekeeper Call Routing](#)
- [Cisco High-Performance Gatekeeper](#)
- [Configuring H.323 Gateways](#)
- [Configuring H.323 Gatekeepers](#)
- [Troubleshooting and Understanding Cisco Gatekeeper Bandwidth Management](#)
- [Configuring H.323 Support for Virtual Interfaces](#)
- [Voice Technology Support](#)
- [Voice and Unified Communications Product Support](#)
- [Troubleshooting Cisco IP Telephony](#) 
- [Technical Support & Documentation – Cisco Systems](#)

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Feb 02, 2006

Document ID: 21063

---