H.323-to-H.323 Interworking on CUBE

This chapter describes how to configure and enable features for H.323-to-H.323 connections on CUBE.

> **Note** H.323 protocol is no longer supported from 17.6.1 onwards. Consider using SIP for multimedia applications.

Configuring H.323-to-H.323 connections on a CUBE open all ports by default. If CUBE has a public IP address and a PSTN connection, CUBE becomes vulnerable to malicious attackers who can execute toll fraud across the gateway. To eliminate this threat, you can bind an interface to a private IP address that is inaccessible to untrusted hosts. In addition, you can protect any public or untrusted interface by configuring a firewall or an access control list (ACL) to prevent unwanted traffic from traversing the router.

Feature Information for H.323-to-H.323 Interworking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 1: Feature Information for H.323-to-H.323 Interworking

Feature Name	Releases	Feature Information
H.323-to-H.323 Connections on a Cisco Unified Border Element	12.3(1)	H.323-to-H.323 Gateway configuration provides a network-to-network demarcation point between independent VoIP and video networks by for billing, security, call-admission control, QoS, and signaling interworking.

Feature Name	Releases	Feature Information
Managing H.323 IP Group Call Capacities	12.2(13)T	Creates a maximum capacity for the IP group providing extra control for load and resource balancing.
Overlap Signaling for H.323-to-H.323 Connections on a Cisco Unified Border Element	12.3(11)T	The terminating gateway is responsible for collecting all the called number digits. Overlap signaling is implemented by matching destination patterns on the dial peers.
Rotary Support	12.3(11)T 12.4(6)T	12.3(11)T—H.323-to-H.323 Call Failure Recovery (Rotary) on a Cisco Unified Border Element. Eliminates codec restrictions and enables the Cisco UBE to restart codec negotiation with the originating endpoint based on the codec capabilities of the next dial peer in the rotary group for H.323-to-H.323 interconnections. 12.4(6)T—Secure RTP with IPSEC for Signaling.
Signal Interworking	12.3(11)T	H.323-to-H.323 Interworking Between Fast Start and Slow Start. This feature enables the Cisco UBE to bridge calls between VoIP endpoints that support only H.323 FastStart procedures and endpoints that support only normal H.245 signaling (SlowStart).

Prerequisites

- Enable CUBE application on a device
- Perform basic H.323 gateway configuration. See Configuring H.323 Gateway
- Perform basic H.323 gatekeeper configuration. See Configuring H.323 Gatekeeper

Restrictions

- Voice class codec is not supported.
- LTI-based transcoding is not supported.
- Supplementary services with transcoding is not supported.
- DTMF Interworking rtp-nte to out of band is not supported when high density transcoder is enabled. Use normal transcoding for rtp-nte to out of band DTMF interworking.
- SCCP based codec transcoding is not supported. An exception to this restriction is slow start to slow start with a static codec.

Slow Start to Fast-Start Interworking

The slow-start to fast-start interworking feature allows two endpoints configured for slow start and fast start respectively to connect with each other through CUBE without dropping the call.

Restrictions for Slow-Start and Fast-Start Interworking

- Slow-start to fast-start interworking is supported only for H.323-to-H.323 calls.
- Transcoding in slow-start to fast-start interworking is not supported.

Enabling Interworking between Slow Start and Fast Start

Configure interworking between slow start and fast start on both inbound and outbound call legs.

✎

| **Note** | This task should not be used in situations where fast-start to fast-start or slow-start to slow-start calls are possible. |

Before you begin

Ensure that a codec is configured on incoming and outgoing call legs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use one of the following commands to configure interworking between slow start and fast start.
   - **call start interwork** in global VoIP configuration mode
   - **call start interwork** in voice class configuration and applied to inbound and outbound dial peers.
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Use one of the following commands to configure interworking between slow start and fast start. - **call start interwork** in global VoIP configuration mode	Enables interworking between slow start and fast start.

Command or Action	Purpose

• **call start interwork** in voice class configuration and applied to inbound and outbound dial peers.
 	
Example:
 	
In global VoIP configuration mode
 	
`Device(config)# voice service voip`
`Device(conf-voi-serv)# h323`
`Device(conf-serv-h323)# call start interwork`
 	
Example:
 	
In voice class configuration mode
 	
`!Configuring a Voice class with Fast Start and Slow`
` Start Interworking`
`Device(config)# voice class h323 10`
`Device(config-class)# call start interwork`
 	
`!Applying the voice class to a dial peer.`
`Device(config)# dial-peer voice 20 voip`
`Device(config-dial-peer)# voice-class h323 10`	

Step 4 **end**	Exits to privileged EXEC mode.

Call Failure Recovery (Rotary)

Call failure recovery (Rotary) is a feature that provides the flexibility to route a call to a destination with multiple paths based on the policy of a service provider. If one path disconnects the call for any reason (like unreachableDestination, destinationReject, noPermission etc), the call can be routed by choosing another dial peer to the same destination based on configured preference.

Rotary is implemented using the dial peer hunt feature (see Configuring Hunt Groups), and the search for a successful dial peer continues until a **huntstop** command is encountered.

The feature described in this chapter is an enhancement that removes a restriction on codec configuration, that requires for identical codec capabilities configured on all dial peers in a rotary group. This is done by supporting an Empty Capability set (TCS=0) when rotary is configured.

The feature allows the CUBE to restart the codec negotiation process with the originating endpoint based on the codec capabilities of the next dial peer in the rotary group.

Enabling Call Failure Recovery (Rotary) without Identical Codec Configuration

Before you begin

Configure Call Failure Recovery (Rotary) using dial-peer hunt groups. See Configuring Dial-Peer Hunt Groups.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **voice service voip**
4. **h323**
5. **emptycapability**
6. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters VoIP voice-service configuration mode.
Step 4	**h323** **Example:** `Device(conf-voi-serv)# h323`	Enters H.323 voice-service configuration mode.
Step 5	**emptycapability** **Example:** `Device(conf-serv-h323)# emptycapability`	Enables call failure recovery (TCS=0) without the need for identical codec configuration.
Step 6	**exit** **Example:** `Router(conf-serv-h323)# exit`	Exits the current mode.

Managing H.323 IP Group Call Capacities

Managing maximum capacity for an IP group is done with carrier IDs created on an IP trunk group. If you do not configure specific carrier IDs, you can use the **ip circuit default only** command to create a single

carrier. However, if you want to use carrier ID-based routing, or if you need extra control for load and resource balancing, you must configure carrier IDs in conjunction with the **voice source-group** command.

CUBE works with the **voice source-group** command to provide matching criteria for incoming calls. The **voice source-group** command assigns a name to a set of source IP group characteristics. The terminating gateway uses these characteristics to identify and translate the incoming VoIP call. If there is no voice source group match, the default carrier ID is used, any source carrier ID on the incoming message is transmitted without change, and no destination carrier is available. Call-capacity information is reported to the gatekeeper, but carrier routing information is not.

If the voice source group matches, the matched source carrier ID is used and the target carrier ID defined in the voice source group is used for the destination carrier ID.

Note   You can use this task only when there are no active calls are active.

\>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **h323**
5. ip circuit max-calls *maximum-calls*
6. ip circuit carrier-id *carrier-name* [reserved-calls *reserved*]
7. **ip circuit default only**
8. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters VoIP voice-service configuration mode.

	Command or Action	**Purpose**
Step 4	**h323** **Example:** `Router(conf-voi-serv)# h323`	Enters H.323 voice-service configuration mode.
Step 5	ip circuit max-calls *maximum-calls* **Example:** `Router(config-serv-h323)# ip circuit max-calls 1500`	(Required only if reserved calls are to exceed 1000) Sets the maximum number of aggregate H.323 IP circuit carrier call legs. If you do not configure this value, the default maximum value is 1000 reserved call legs. You may need to configure a lower value to obtain overload behavior. You can also configure a higher value. **Note** After you set a maximum number of call legs for defined circuits, any aggregate capacity left over is available for default circuits. For example, if you specify 1000 as the maximum number of call legs and then reserve 200 call legs for defined circuits, 800 call legs are available for use by default circuits. **Note** CUBE prevents you from allocating all of the capacity to specified carriers; at least one available circuit is required, which can be the default.
Step 6	ip circuit carrier-id *carrier-name* [reserved-calls *reserved*] **Example:** `Router(config-serv-h323)# ip circuit carrier-id AA reserved-calls 500`	(Optional) Defines an IP circuit using the specified name as the circuit ID. **Note** The **reserved** keyword for this command is optional. Using this keyword creates a specified maximum number of calls for that circuit ID. The default value is 200 call legs.
Step 7	**ip circuit default only** **Example:** `Router(config-serv-h323)# ip circuit default only`	(Optional) Creates a single carrier to use all of the call capacity available to CUBE. **Note** If you use the **ip circuit default only** command, you cannot use the **ip** circuit carrier-id command to configure more circuits. Using the ip circuit default only command creates a single carrier using the default carrier name.
Step 8	**exit** **Example:** `Router(conf-serv-h323)# exit`	Exits the current mode.

Configuration Examples for Managing H.323 IP Group Call Capacities

The following examples show a default carrier with no voice source group configured:

Example: Default Carrier with No Voice Source Group

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit default only
```

If there is no incoming source carrier ID:

- Capacity only is reported to the gatekeeper using the default circuit (two call legs).
- No source or destination carrier information is reported.

If there is an incoming source carrier ID:

- Two call legs are counted against the default circuit and reported to the GK.
- The source carrier ID is passed through the gateway to the terminating leg.

The following examples show a configuration with more reserved calls than the default value for the **max-calls** argument (1000):

Example: Configuration with Default Calls in Excess of 1000

This example assigns 1100 calls to other carriers, leaving 400 calls available to the default carrier:

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit carrier-id AA reserved-calls 500
  ip circuit carrier-id bb reserved-calls 500
  ip circuit carrier-id cc reserved-calls 100
```

The following examples show the default carrier configured with an incoming source carrier but no voice source group configured.

> **Note** In this example, 800 call legs are implicitly reserved for the default circuit.

Example: Default Carrier and Incoming Source Carrier with No Voice Source Group

> **Note** A gatekeeper is required with carrier-id routing.

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit carrier-id AA reserved-calls 200
```

If there is no incoming source carrier ID:

- Capacity only is reported to the GK using the default circuit (two call legs).
- No source or destination carrier information is reported.

If there is an incoming source carrier ID called "AA":

- One call leg is counted against circuit "AA".
- One call leg (outbound) is counted against the default circuit.
- The source carrier ID is passed through the gateway to the terminating leg.

If there is an incoming source carrier ID called "BB" (for example) or anything other than "AA":

- Two call legs are counted against the default circuit.
- The source carrier ID "BB" is passed through the gateway to the terminating leg.

The following examples show the first voice source-group match case:

Example: Voice Source-Group Match Case 1

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit carrier-id AA reserved-calls 200
!
voice source-group 1
 carrier-id source AA
 carrier-id target AA
```

If there is no incoming source carrier ID, the default circuit is used because there is no match in the voice source group.

If there is an incoming source carrier ID called "AA," the following are in effect:

- The voice source group matches.
- Both call legs are counted against circuit "AA".
- The source carrier ID is passed through the gateway to the terminating leg.
- The destination carrier ID is "AA".

The following examples show the second voice source group match case:

Example: Voice Source-Group Match Case 2

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit carrier-id AA reserved-calls 200
  ip circuit carrier-id BB reserved-calls 200
!
voice source-group 1
 carrier-id source AA
 carrier-id target BB
```

If there is no incoming source carrier ID, the default circuit is used because there is no match in the voice source group.

If there is an incoming source carrier ID called "AA":

- The voice source-group matches.
- One leg is counted against circuit "AA".
- One leg is counted against circuit "BB".

- The source carrier ID is passed through the gateway to the terminating leg.
- The destination carrier ID is "BB".

The following examples show the third voice source-group match case:

Example: Voice Source-Group Match Case 3

```
voice service voip
 allow-connections h323 to h323
 h323
  ip circuit max-calls 1000
  ip circuit carrier-id AA reserved-calls 200
  ip circuit carrier-id BB reserved-calls 200
!
voice source-group 1
 access-list 1
 carrier-id source BB
```

If the access-list matches, the following apply:

- One leg is counted against circuit "BB".
- One leg is counted against the default circuit (for the destination circuit).
- The source carrier ID is synthesized to "BB" and used to report to the gatekeeper. It is also used on the outgoing setup.

If a source carrier ID is received on the incoming setup, it is overridden with the synthesized carrier ID

Overlap Signaling

Overlap signaling requires that called digits be sent one-by-one as they are received from the calling device. The first digit is sent in a call setup message and subsequent digits are sent in information messages. This technique is used when a receiving gateway is able to recognize variable-length phone numbers, and requires that the originating gateway signal the end of the call setup process.

Overlap signaling is implemented by matching destination patterns on the dial peers. When H.225 signal overlap is configured on the originating gateway, it sends the SETUP to the terminating gateway once a dial-peer match is found. The originating gateway sends all further digits received from the user to the terminating gateway using INFO messages until it receives a sending complete message from the user. The terminating gateway receives the digits in SETUP and subsequent INFO messages and does a dial-peer match. If a match is found, it sends a SETUP with the collected digits to the PSTN. All subsequent digits are sent to the PSTN using INFO messages to complete the call.

Configuring Overlap Signaling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **h323**
5. **h225 signal overlap**
6. **h225 timeout t302** *seconds*

7.   exit

DETAILED STEPS

Procedure

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters VoIP voice-service configuration mode.
Step 4	**h323** **Example:** `Router(conf-voi-serv)# h323`	Enters H.323 voice-service configuration mode.
Step 5	**h225 signal overlap** **Example:** `Router(conf-serv-h323)# h225 signal overlap`	Activates overlap signaling to the destination gateway.
Step 6	**h225 timeout t302** *seconds* **Example:** `Router(conf-serv-h323)# h225 timeout t302 15`	Sets the t302 timer timeout value. The argument is as follows: • *seconds*— Number of seconds for timeouts. Range: 1 to 30.
Step 7	**exit** **Example:** `Router(conf-serv-h323)# exit`	Exits the current mode.

Verifying H.323-to-H.323 Interworking

To verify Cisco Unified Border Element feature configuration and operation, perform the following steps (listed alphabetically) as appropriate.

✎

Note   The word "calls" refers to call legs in some commands and output.

SUMMARY STEPS

1. **show call active video**
2. **show call active voice**
3. **show call active fax**
4. **show call history video**
5. **show call history voice**
6. **show call history fax**
7. **show crm**
8. **show dial-peer voice**
9. **show running-config**
10. **show voip rtp connections**

DETAILED STEPS

Procedure

Step 1   **show call active video**

Use this command to display the active video H.323 call legs.

Step 2   **show call active voice**

Use this command to display call information for voice calls that are in progress.

Step 3   **show call active fax**

Use this command to display the fax transmissions that are in progress.

Step 4   **show call history video**

Use this command to display the history of video H.323 call legs.

Step 5   **show call history voice**

Use this command to display the history of voice call legs.

Step 6   **show call history fax**

Use this command to display the call history table for fax transmissions that are in progress.

Step 7   **show crm**

Use this command to display the carrier ID list or IP circuit utilization.

Step 8   **show dial-peer voice**

Use this command to display information about voice dial peers.

Step 9   **show running-config**

Use this command to verify which H.323-to-H.323, H.323-to-SIP, or SIP-to-SIP connection types are supported.

Step 10   **show voip rtp connections**

Use this command to display active Real-Time Transport Protocol (RTP) connections.

Troubleshooting H.323-to-H.323 Interworking

⚠

Caution   Under moderate traffic loads, these **debug** commands produce a high volume of output.

- **debug cch323 all**
- **debug h225 asn1**
- **debug h225 events**
- **debug h225 q931**
- **debug h245 asn1**
- **debug h245 events**
- **debug voip ipipgw**
- **debug voip ccapi inout**

Troubleshooting H.323-to-H.323 Interworking