Nano Cisco Unified Border Element Configuration Guide, Cisco IOS Release 15M&T

First Published: July 23, 2013

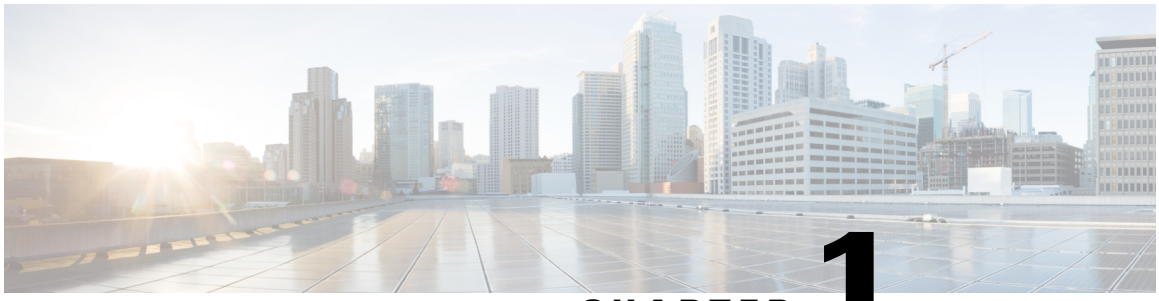
Last Modified: March 31, 2016

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

PART I

SIP Trunking Topology Deployment



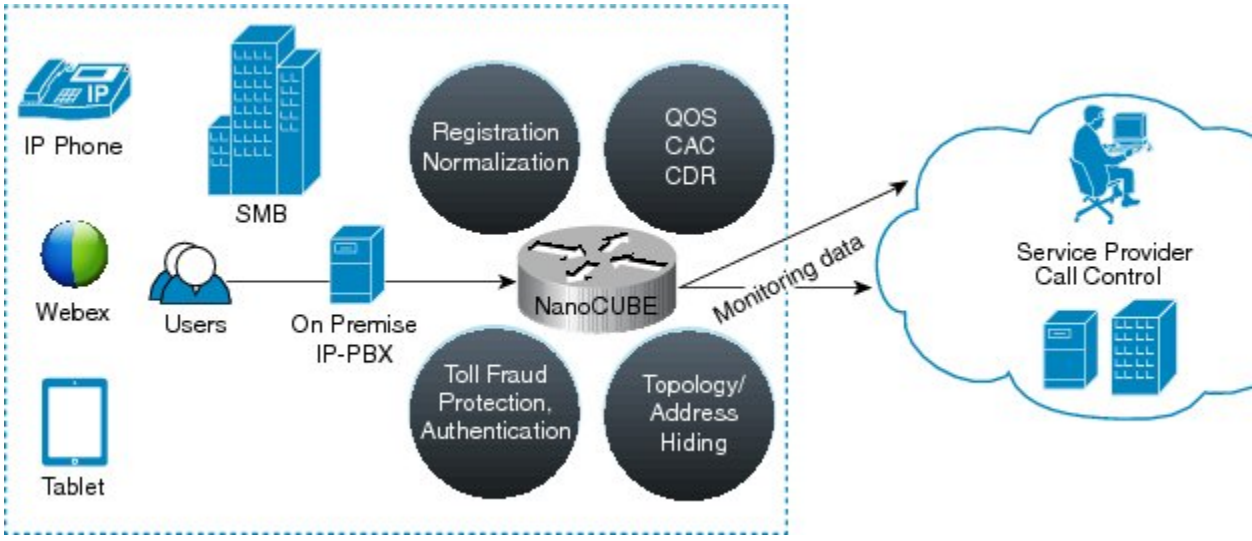
CHAPTER 1

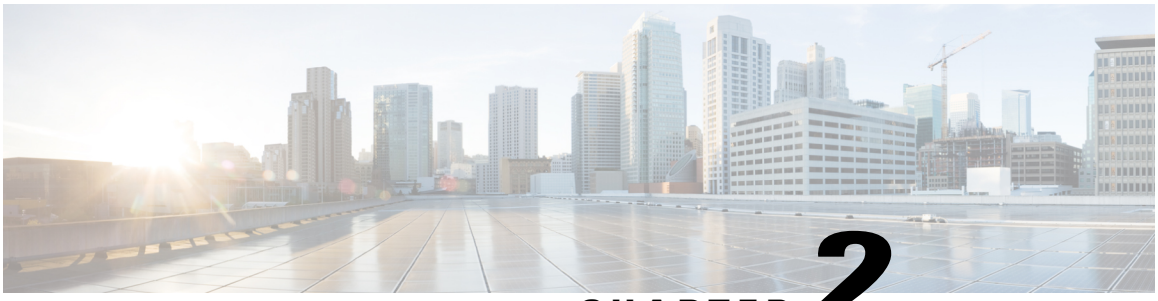
SIP Trunking Topology

Using NanoCUBE in SIP Trunking Topology

NanoCUBE can be used in SIP trunking deployments in the same way as Cisco UBE is being used. In these deployments, the Call Control Agent (CUCM, CME, and so on) are locally available. NanoCUBE has two SIP trunks -- one towards the local Call Control agent and another towards the Service Provider. All end-user services are handled by the local Call Control agent, and NanoCUBE connects these enterprises to the Service Provider or the Packet-Switched Telephone Network over a SIP trunk.

Figure 1: NanoCUBE in SIP Trunking Scenario





CHAPTER 2

Configuration of SIP Trunking for PSTN Access SIP-to-SIP

This Cisco Unified Border Element (Enterprise) is a special Cisco IOS XE software image that runs on Cisco ASR1000. It provides a network-to-network interface point for billing, security, call admission control, quality of service, and signaling interworking. This chapter describes basic gateway functionality, software images, topology, and summarizes supported features.

Note Cisco Product Authorization Key (PAK)--A Product Authorization Key (PAK) is required to configure some of the features described in this guide. Before you start the configuration process, please register your products and activate your PAK at the following URL http://www.cisco.com/go/license .

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Configuration of SIP Trunking for PSTN Access SIP-to-SIP Features

This chapter contains the following configuration topics:

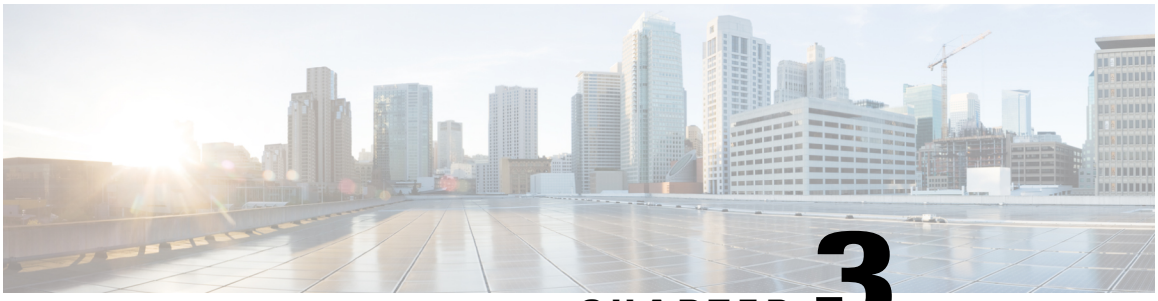
Cisco UBE (Enterprise) Prerequisites and Restrictions

- Prerequisites for Cisco Unified Border Element (Enterprise)

- Restrictions for Cisco Unified Border Element (Enterprise)

SIP trunk Monitoring

- Configuring Cisco UBE Out-of-dialog OPTIONS Ping for Specified SIP Servers or Endpoints



CHAPTER 3

Configuring SIP Registration Proxy on Cisco UBE

The Support for SIP Registration Proxy on Cisco UBE feature provides support for sending outbound registrations from Cisco Unified Border Element (UBE) based on incoming registrations. This feature enables direct registration of Session Initiation Protocol (SIP) endpoints with the SIP registrar in hosted unified communication (UC) deployments. This feature also provides various benefits for handling Cisco UBE deployments with no IP private branch exchange (PBX) support.

In certain Cisco UBE deployments, managed services are offered without an IPPBX installed locally at the branch office. A PBX located at the service provider (SP) offers managed services to IP phones. A Cisco UBE device located at the branch office provides address translation services. However, the registration back-to-back functionality is required to get the phone registered, so that calls can be routed to the branch or the phones.

In such deployment scenarios, enabling the Support for SIP Registration Proxy on Cisco UBE feature provides the following benefits:

- Support for back-to-back user agent (B2BUA) functionality.
- Options to configure rate-limiting values such as expiry time, fail-count value, and a list of registrars to be used for the registration.
- Registration overload protection facility.
- Option to route calls to the registering endpoint (user or phone).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Registration Pass-Through Modes

Cisco UBE uses the following two modes for registration pass-through:

End-to-End Mode

In the end-to-end mode, Cisco UBE collects the registrar details from the Uniform Resource Identifier (URI) and passes the registration messages to the registrar. The registration information contains the expiry time for rate-limiting, the challenge information from the registrar, and the challenge response from the user.

Cisco UBE also passes the challenge to the user if the register request is challenged by the registrar. The registrar sends the 401 or 407 message to the user requesting for user credentials. This process is known as challenge.

Cisco UBE ignores the local registrar and authentication configuration in the end-to-end mode. It passes the authorization headers to the registrar without the header configuration.

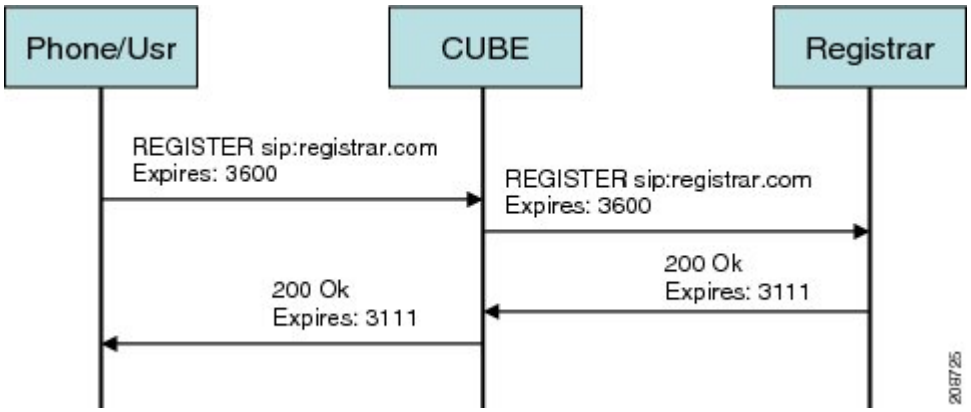
End-to-End Mode--Call Flows

This section explains the following end-to-end pass-through mode call flows:

Register Success Scenario

The figure below shows an end-to-end registration pass-through scenario where the registration request is successful.

Figure 2: End-to-End Registration Pass-through Mode--Register Success Scenario



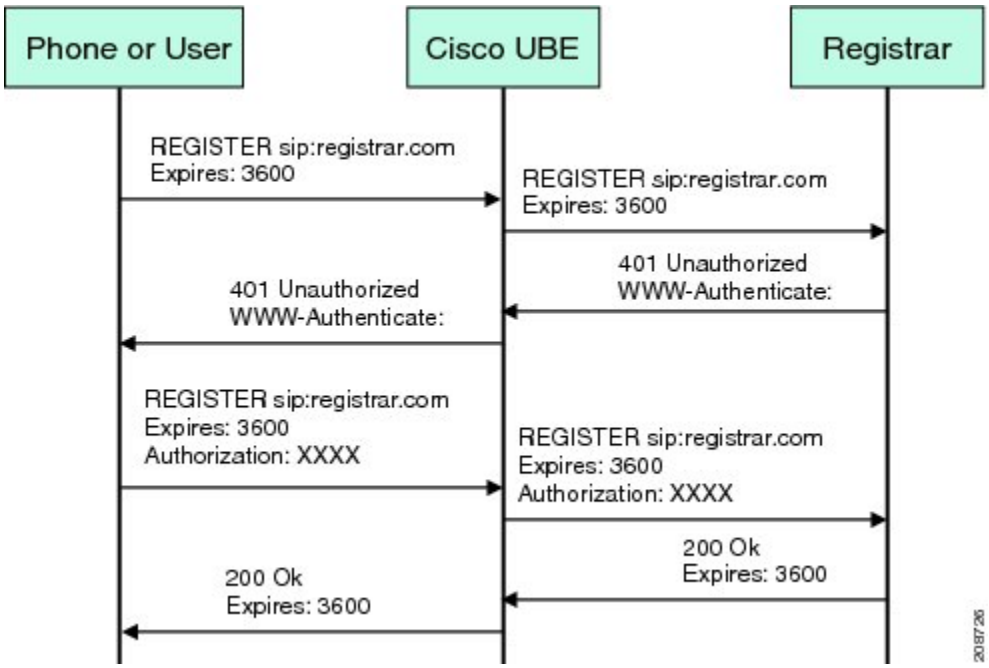
The register success scenario for the end-to end registration pass-through mode is as follows:

1. The user sends the register request to Cisco UBE.
2. Cisco UBE matches the request with a dial peer and forwards the request to the registrar.
3. Cisco UBE receives a success response message (200 OK message) from the registrar and forwards the message to the endpoint (user).
4. The registrar details and expiry value are passed to the user.

Registrar Challenging the Register Request Scenario

The figure below shows an end-to end registration pass-through scenario where the registrar challenges the register request.

Figure 3: End-to-End Registration Pass-through Mode--Registrar Challenging the Register Request Scenario



The following scenario explains how the registrar challenges the register request:

1. The user sends the register request to Cisco UBE.
2. Cisco UBE matches the register request with a dial peer and forwards it to the registrar.
3. The registrar challenges the register request.
4. Cisco UBE passes the registrar response and the challenge request, only if the registrar challenges the request to the user.
5. The user sends the register request and the challenge response to the Cisco UBE.
6. Cisco UBE forwards the response to the registrar.
7. Cisco UBE receives success message (200 OK message) from the registrar and forwards it to the user.

Peer-to-Peer Mode

In the peer-to-peer registration pass-through mode, the outgoing register request uses the registrar details from the local Cisco UBE configuration. Cisco UBE answers the challenges received from the registrar using the configurable authentication information. Cisco UBE can also challenge the incoming register requests and authenticate the requests before forwarding them to the network.

In this mode, Cisco UBE sends a register request to the registrar and also handles register request challenges. That is, if the registration request is challenged by the registrar (registrar sends 401 or 407 message), Cisco UBE forwards the challenge to the user and then passes the challenge response sent by the user to the registrar. In the peer-to-peer mode, Cisco UBE can use the **authentication** command to calculate the authorization header and then challenge the user depending on the configuration.

Note The **registrar** command must be configured in peer-to-peer mode. Otherwise, the register request is rejected with the 503 response message.

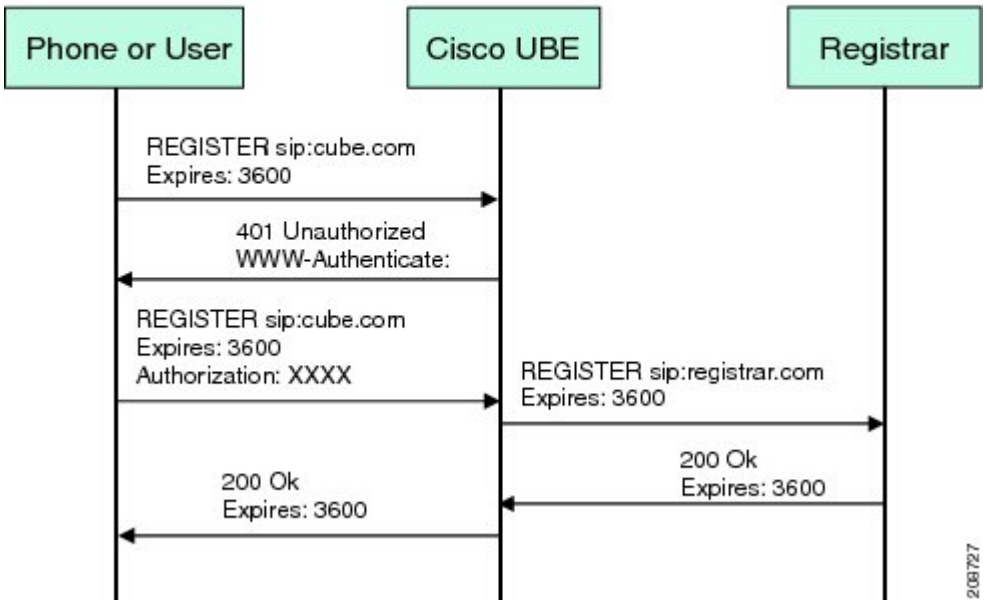
Peer-to-Peer Mode--Call Flows

This section explains the following peer-to-peer pass-through mode call flows:

Register Success Scenario

The figure below shows a peer-to-peer registration pass-through scenario where the registration request is successful.

Figure 4: Peer-to-Peer Registration Pass-through Mode--Register Success Scenario



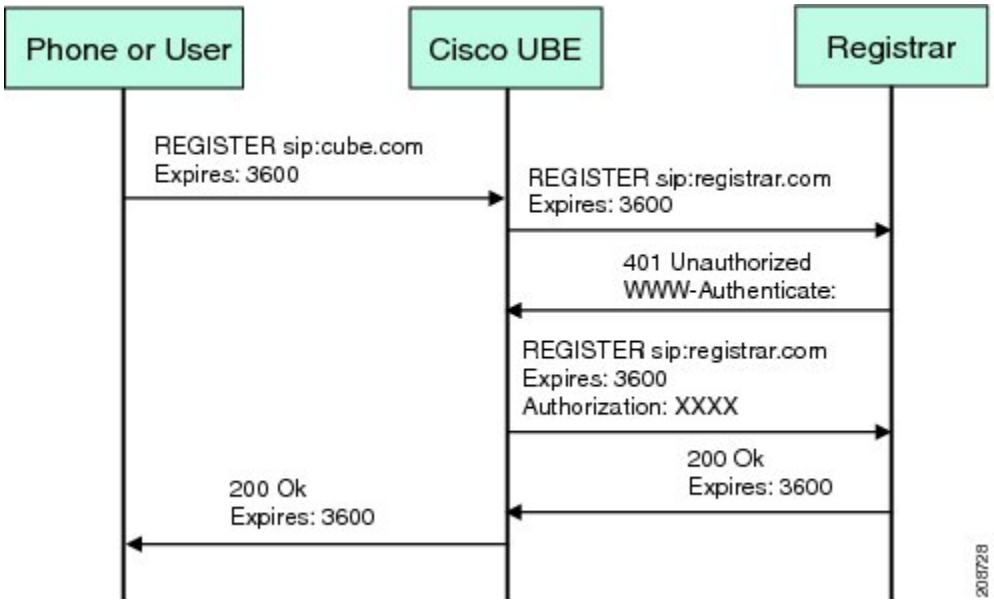
The register success scenario for a peer-to-peer registration pass-through mode is as follows:

1. The user sends the register request to Cisco UBE.
2. Cisco UBE matches the register request with a dial peer and forwards the register request to the registrar.
3. Cisco UBE receives a success message (200 OK message) from the registrar and forwards it to the endpoint (user). The following functions are performed:
   - Cisco UBE picks up the details about the registrar from the configuration.
   - Cisco UBE passes the registrar details and expiry value to the user.

Registrar Challenging the Register Request Scenario

The figure below shows a peer-to-peer registration pass-through scenario where the registration request is challenged by the registrar.

Figure 5: Peer-to-Peer Registration Pass-through Mode--Registrar Challenging the Register Request Scenario



The following scenario explains how the registrar challenges the register request:

1. The user sends the register request to Cisco UBE.
2. Cisco UBE matches the register request with a dial peer and forwards the register request to the registrar.
3. The user responds to the challenge request.
4. Cisco UBE validates the challenge response and forwards the register request to the registrar.
5. Cisco UBE receives a success message from the registrar and forwards it to the endpoint (user).

Registration in Different Registrar Modes

This section explains SIP registration pass-through in the following registrar modes:

Primary-Secondary Mode

In the primary-secondary mode the register message is sent to both the primary and the secondary registrar servers simultaneously.

The register message is processed as follows:

- The first successful response is passed to the phone as a SUCCESS message.
- All challenges to the request are handled by Cisco UBE.
- If the final response received from the primary and the secondary servers is an error response, the error response that arrives later from the primary or the secondary server is passed to the phone.
- If only one registrar is configured, a direct mapping is performed between the primary and the secondary server.
- If no registrar is configured, or if there is a Domain Name System (DNS) failure, the "503 service not available" message is sent to the phone.

DHCP Mode

In the DHCP mode the register message is sent to the registrar server using DHCP.

Multiple Register Mode

In the multiple register mode, you can configure a dial peer to select and enable the indexed registrars. Register messages must be sent only to the specified index registrars.

The response from the registrar is mapped the same way as in the primary-secondary mode. See the Registration in Different Registrar Modes, on page 12.

Registration Overload Protection

The registration overload protection functionality enables Cisco UBE to reject the registration requests that exceed the configured threshold value.

To support the registration overload protection functionality, Cisco UBE maintains a global counter to count all the pending outgoing registrations and prevents the overload of the registration requests as follows:

- The registration count is decremented if the registration transaction is terminated.
- The outgoing registrations are rejected if the count goes beyond a configured threshold.
- The incoming register request is rejected with the 503 response if the outgoing registration is activated by the incoming register request.
- A retry timer set for a random value is used for attempting the registration again if the registrations are originated from Cisco UBE or a gateway.

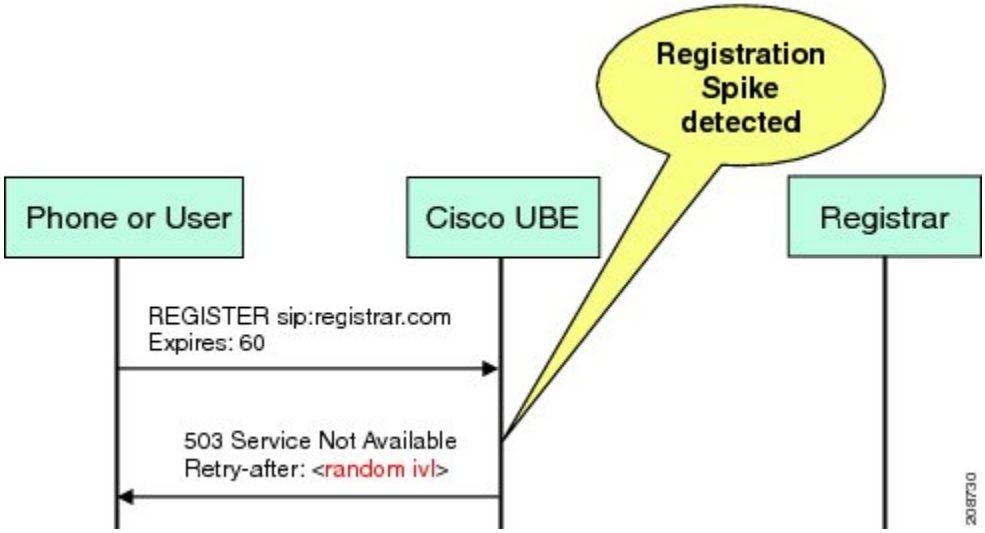
The registration overload protection functionality protects the network from the following:

- Avalanche Restart--All the devices in the network restart at the same time.
- Component Failures--Sudden burst of load is routed through the device due to a device failure.

Registration Overload Protection--Call Flow

The figure below shows the call flow when the register overload protection functionality is configured on Cisco UBE:

Figure 6: Register Overload Protection



The following steps explain the register overload protection scenario:

1. The user sends a register request to Cisco UBE.
2. Cisco UBE matches the request with a dial peer and forwards the register request to the registrar.
3. The registration is rejected with a random retry value when the registration threshold value is reached.

Note The call flow for the DNS query on the Out Leg is the same for the end-to-end and peer-to-peer mode.

Registration Rate-limiting

The registration rate-limiting functionality enables you to configure different SIP registration pass-through rate-limiting options. The rate-limiting options include setting the expiry time and the fail count value for a Cisco UBE. You can configure the expiry time to reduce the load on the registrar and the network. Cisco UBE limits the reregistration rate by maintaining two different timers--in-registration timer and out-registration timer.

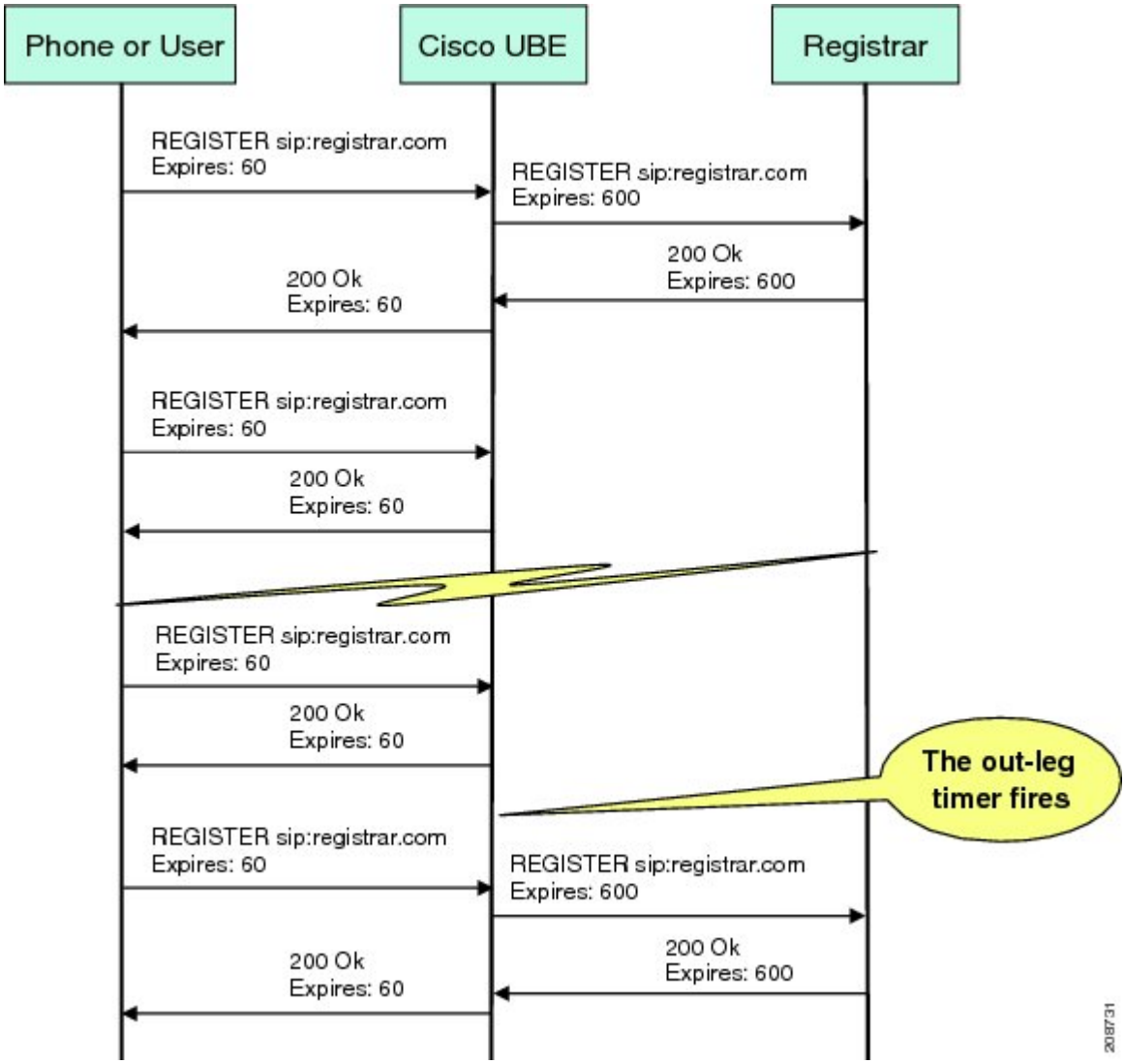
The initial registration is triggered based on the incoming register request. The expiry value for the outgoing register is selected based on the Cisco UBE configuration. On receiving the 200 OK message (response to

the BYE message) from the registrar, a timer is started using the expiry value available in the 200 OK message. The timer value in the 200 OK message is called the out-registration timer. The success response is forwarded to the user. The expiry value is taken from the register request and the timer is started accordingly. This timer is called the in-registration timer. There must be a significant difference between the in-registration timer and the out-registration timer values for effective rate-limiting.

Registration Rate-limiting Success--Call Flow

The figure below shows the call flow when the rate-limiting functionality is successful:

Figure 7: Rate-limiting Success Scenario



The following steps explain a scenario where the rate-limiting functionality is successful:

1 The user sends the register request to Cisco UBE.

2 Cisco UBE matches the registration request with a dial peer and forwards it to the registrar. The outgoing register request contains the maximum expiry value if the rate-limiting functionality is configured.

3 The registrar accepts the registration.

4 Cisco UBE forwards the success response with the proposed expiry timer value.

5 The user sends the reregistration requests based on the negotiated value. Cisco UBE resends the register requests until the out-leg expiry timer value is sent.

6 Cisco UBE forwards the subsequent register request to the registrar, if the reregister request is received after the out-leg timer is reached.

Prerequisites for SIP Registration Proxy on Cisco UBE

- You must enable the local SIP registrar. See Enabling Local SIP Registrar, on page 15.
- You must configure dial peers manually for call routing and pattern matching

Cisco Unified Border Element

- Cisco IOS Release 15.1(3)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.7S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions

- IPv6 support is not provided.

Configuring Support for SIP Registration Proxy on Cisco UBE

Enabling Local SIP Registrar

Perform this task to enable the local SIP registrar.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **registrar server** [**expires** [**max** *value*] [**min** *value*]]
6. **end**

DETAILED STEPS

	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 voice-service configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters service SIP configuration mode.
Step 5	**registrar server** [**expires** [**max** *value*] [**min** *value*]] **Example:** `Device(conf-serv-sip)# registrar server`	Enables the local SIP registrar. • Optionally you can configure the expiry time of the registrar using the following keywords: • **expires**--Configures the registration expiry time. • **max**--Configures the maximum registration expiry time. • **min**--Configures the minimum registration expiry time. **Note** The **registrar** command must be configured in peer-to-peer mode. Otherwise, the register request is rejected with the 503 response message.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Exits service SIP configuration mode and returns to privileged EXEC mode.

Configuring SIP Registration at the Global Level

Perform this task to configure the support for the SIP registration proxy on the Cisco UBE at the global level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **registration passthrough** [**static**] [**rate-limit** [**expires** *value*] [**fail-count** *value*]] [**registrar-index** [*index*]]
6. **end**

DETAILED STEPS

	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 voice-service configuration mode.

	Command or Action	Purpose
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters service SIP configuration mode.
Step 5	**registration passthrough** [**static**] [**rate-limit** [**expires** *value*] [**fail-count** *value*]] [**registrar-index** [*index*]] **Example:** `Device(conf-serv-sip)# registration passthrough`	Configures the SIP registration pass-through options. • You can specify different SIP registration pass-through options using the following keywords: • **rate-limit**--Enables rate-limiting. • **expires**--Configures expiry value for rate-limiting. • **fail-count**--Configures fail count during rate-limiting. • **registrar-index**--Configures a list of registrars to be used for registration.
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Exits service SIP configuration mode and returns to privileged EXEC mode.

Configuring SIP Registration at the Dial Peer Level

Perform this task to configure SIP registration at the dial peer level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **voatm** | **vofr** | **voip**}
4. **voice-class sip registration passthrough static** [**rate-limit** [**expires** *value*] [**fail-count** *value*] [**registrar-index** [*index*]] | **registrar-index** [*index*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* {**pots** | **voatm** | **vofr** | **voip**} **Example:** `Device(config)# dial-peer voice 444 voip`	Enters dial peer voice configuration mode.
Step 4	**voice-class sip registration passthrough static** [**rate-limit** [**expires** *value*] [**fail-count** *value*] [**registrar-index** [*index*]] | **registrar-index** [*index*]] **Example:** `Device(config-dial-peer)# voice-class sip registration passthrough static`	Configure SIP registration pass-through options on a dial peer on a dial peer. • You can specify different SIP registration pass-through options using the following keywords: • **rate-limit**--Enables rate-limiting. • **expires**--Configures expiry value for rate-limiting. • **fail-count**--Configures fail count during rate-limiting. • **registrar-index**--Configures a list of registrars to be used for registration.
Step 5	**exit** **Example:** `Device(config-dial-peer)# exit`	Exits dial peer voice configuration mode and returns to global configuration mode.

Configuring Registration Overload Protection Functionality

Perform this task to configure registration overload protection functionality on Cisco UBE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registration spike** *max-number*
5. **end**

DETAILED STEPS

	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	**sip-ua** **Example:** `Device(config)# sip-ua`	Enters SIP user-agent configuration mode.
Step 4	**registration spike** *max-number* **Example:** `Device(config-sip-ua)# registration spike 100`	Configures registration overload protection functionality on Cisco UBE.
Step 5	**end** **Example:** `Device(config-sip-ua)# end`	Exits SIP user-agent configuration mode and returns to privileged EXEC mode.

Configuring Cisco UBE to Route a Call to the Registrar Endpoint

Perform this task to configure Cisco UBE to route a call to the registrar endpoint.

Note You must perform this configuration on a dial peer that is pointing towards the endpoint.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **voatm** | **vofr** | **voip**}
4. **session target registrar**
5. **exit**

DETAILED STEPS

	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	**dial-peer voice** *tag* {**pots**	**voatm**	**vofr**	**voip**} **Example:** `Device(config)# dial-peer voice 444 voip`	Enters dial peer voice configuration mode.
Step 4	**session target registrar** **Example:** `Device(config-dial-peer)# session target registrar`	Configures Cisco UBE to route the call to the registrar endpoint.
Step 5	**exit** **Example:** `Device(config-dial-peer)# exit`	Exits dial peer voice configuration mode and returns to global configuration mode.

Verifying the SIP Registration on Cisco UBE

Perform this task to verify the configuration for SIP registration on Cisco UBE. The **show** commands need not be entered in any specific order.

SUMMARY STEPS

1. **enable**
2. **show sip-ua registration passthrough status**
3. **show sip-ua registration passthrough status detail**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 **show sip-ua registration passthrough status**
Displays the SIP user agent (UA) registration pass-through status information.

Example:

```
Device# show sip-ua registration passthrough status

CallId       Line         peer         mode In-Exp       reg-I Out-Exp
============ ============ ============ ==== ============ ===== ============
771          5500550055   1             p2p  64           1     64
================================================================================
```

Step 3 **show sip-ua registration passthrough status detail**
Displays the SIP UA registration pass-through status information in detail.

Example:

```
Device# show sip-ua registration passthrough status detail
============================================================
Configured Reg Spike Value: 0
Number of Pending Registrations: 0
============================================================
Call-Id: 763
Registering Number: 5500550055
Dial-peer tag: 601
Pass-through Mode: p2p
Negotiated In-Expires: 64 Seconds
Next In-Register Due in: 59 Seconds
In-Register Contact: 9.45.36.5
----------------------------------------
 Registrar Index: 1
 Registrar URL: ipv4:9.45.36.4
 Negotiated Out-Expires: 64 Seconds
 Next Out-Register After: 0 Seconds
============================================================
```

The following section will be added to the "Examples" section of the SIP to SIP chapter.

Example Configuring Support for SIP Registration Proxy on Cisco UBE

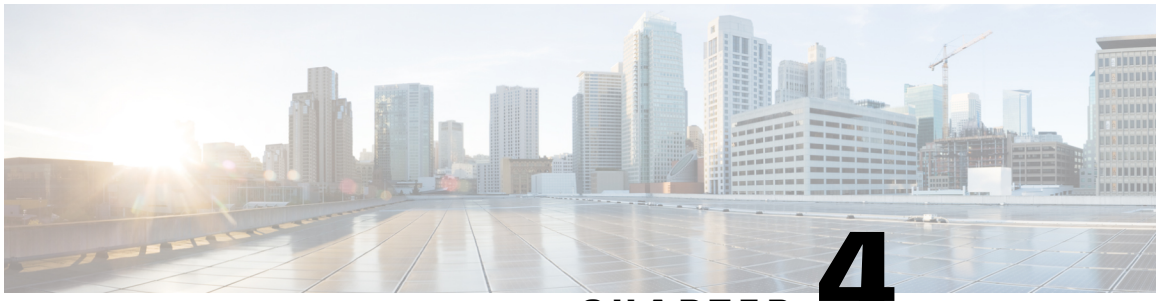
The following example shows how to configure support for the SIP registration proxy on the Cisco UBE.

```
!
!
voice service voip
sip
  registrar server expires max 121 min 61
  registration passthrough static rate-limit expires 9000 fail-count 5 registrar-index 1 3
 5
!
dial-peer voice 1111 voip
 destination-pattern 1234
 voice-class sip pass-thru content unsupp
 session protocol sipv2
 session target registrar
!
dial-peer voice 1111 voip
 destination-pattern 1234
 voice-class sip pass-thru content unsupp
 voice-class sip registration passthrough static rate-limit expires 9000 fail-count 5
registrar-index 1 3 5
 authentication username 1234 password 7 075E731F1A realm cisco.com
 session protocol sipv2
 session target registrar
!
sip-ua
 registration spike 1000
!
!
```

Feature Information for Support for SIP Registration Proxy on Cisco UBE

Table 1: Feature Information for Support for SIP Registration Proxy on Cisco UBE

Feature Name	Releases	Feature Information
Support for SIP Registration Proxy on Cisco UBE	15.1(3)T	The Support for SIP Registration Proxy on Cisco UBE feature provides support for sending outbound registrations from Cisco UBE based on incoming registrations. This feature enables direct registration of SIP endpoints with the SIP registrar in hosted UC deployments. This feature also provides various benefits for handling Cisco UBE deployments with no IPPBX support. The following commands were introduced or modified: **authentication** (dial peer), **registrar server**, **registration passthrough**, **registration spike**, **show sip-ua registration passthrough status**, **voice-class sip registration passthrough static rate-limit**.
Support for SIP Registration Proxy on Cisco UBE	Cisco IOS XE Release 3.7S	The Support for SIP Registration Proxy on Cisco UBE feature provides support for sending outbound registrations from Cisco UBE based on incoming registrations. This feature enables direct registration of SIP endpoints with the SIP registrar in hosted UC deployments. This feature also provides various benefits for handling Cisco UBE deployments with no IPPBX support. The following commands were introduced or modified: **authentication** (dial peer), **registrar server**, **registration passthrough**, **registration spike**, **show sip-ua registration passthrough status**, **voice-class sip registration passthrough static rate-limit**.



CHAPTER 4

Cisco UBE Out-of-dialog OPTIONS Ping

The Cisco Unified Border Element Out-of-dialog (OOD) Options Ping feature provides a keepalive mechanism at the SIP level between any number of destinations.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Out-of-dialog SIP OPTIONS Ping

The following are required for OOD Options ping to function. If any are missing, the Out-of-dialog (OOD) Options ping will not be sent and the dial peer is reset to the default active state.

- Dial-peer should be in active state
- Session protocol must be configured for SIP

- Configure Session target or outbound proxy must be configured. If both are configured, outbound proxy has preference over session target.

Cisco Unified Border Element

- Cisco IOS Release 15.0(1)M or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.1S or a later release must be installed and running on your Cisco ASR 1000 Series Router

Restrictions for Cisco Out-of-dialog SIP OPTIONS Ping for Specified SIP Servers or Endpoints

- The Cisco Unified Border Element OOD Options ping feature can only be configured at the VoIP Dial-peer level.
- All dial peers start in an active (not busied out) state on a router boot or reboot.
- If a dial-peer has both an outbound proxy and a session target configured, the OOD options ping is sent to the outbound proxy address first.
- Though multiple dial-peers may point to the same SIP server IP address, an independent OOD options ping is sent for each dial-peer.
- If a SIP server is configured as a DNS hostname, OOD Options pings are sent to all the returned addresses until a response is received.
- Configuration for Cisco Unified Border Element OOD and TDM Gateway OOD are different, but can co-exist.

Information about Cisco UBE Out-of-dialog OPTIONS Ping

The Out-of-dialog (OOD) Options Ping feature provides a keepalive mechanism at the SIP level between any number of destinations. A generic heartbeat mechanism allows Cisco Unified Border Element to monitor the status of SIP servers or endpoints and provide the option of busying-out a dial-peer upon total heartbeat failure. When a monitored endpoint heartbeat fails, the dial-peer is busied out. If an alternate dial-peer is configured for the same destination pattern, the call is failed over to the next preferred dial peer, or else the on call is rejected with an error cause code.

The table below describes error codes option ping responses considered unsuccessful and the dial-peer is busied out for following scenarios:

Table 2: Error Codes that busyout the endpoint

Error Code	Description
503	service unavailable

Error Code	Description
505	sip version not supported
no response	i.e. request timeout

All other error codes, including 400 are considered a valid response and the dial peer is not busied out.

Note The purpose of this feature is to determine if the SIP session protocol on the endpoint is UP and available to handle calls. It may not handle OPTIONS message but as long as the SIP protocol is available, it should be able to handle calls.

When a dial-peer is busied out, Cisco Unified Border Element continues the heartbeat mechanism and the dial-peer is set to active upon receipt of a response.

Configuring Cisco UBE Out-of-dialog OPTIONS Ping for Specified SIP Servers or Endpoints

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip options-keepalive** {**up-interval** *seconds* | **down-interval** *seconds* | **retry** *retries*}
5. **exit**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 200 voip`	Enters dial-peer configuration mode for the VoIP peer designated by tag.
Step 4	**voice-class sip options-keepalive** {**up-interval** *seconds* | **down-interval** *seconds* | **retry** *retries*} **Example:** `Device(config-dial-peer)# voice-class sip options-keepalive up-interval 12 down-interval 65 retry 3`	Monitors connectivity between endpoints. • **up-interval seconds** -- Number of up-interval seconds allowed to pass before marking the UA as unavailable.The range is 5-1200. The default is 60. • **down-interval seconds** -- Number of down-interval seconds allowed to pass before marking the UA as unavailable.The range is 5-1200. The default is 30. • **retry retries** -- Number of retry attempts before marking the UA as unavailable. The range is 1 to 10. The default is 5 attempts.
Step 5	**exit** **Example:** `Device(config-dial-peer)# exit`	Exits the current mode.

Troubleshooting Tips

The following commands can help troubleshoot the OOD Options Ping feature:

- **debug ccsip all** --shows all Session Initiation Protocol (SIP)-related debugging.
- **show dial-peer voice x** --shows configuration of keepalive information.

```
Device# show dial-peer voice | in options
voice class sip options-keepalive up-interval 60 down-interval 30 retry 5
voice class sip options-keepalive dial-peer action  = active
```

- **show dial-peer voice summary** --shows Active or Busyout dial-peer status.

```
Device# show dial-peer voice summary
          AD                  PRE PASS
TAG TYPE  MIN  OPER PREFIX    DEST-PATTERN KEEPALIVE
111 voip  up     up              0 syst   active
9  voip   up    down             0 syst   busy-out
```

Feature Information for Cisco UBE Out-of-dialog OPTIONS Ping for Specified SIP Servers or Endpoints

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.

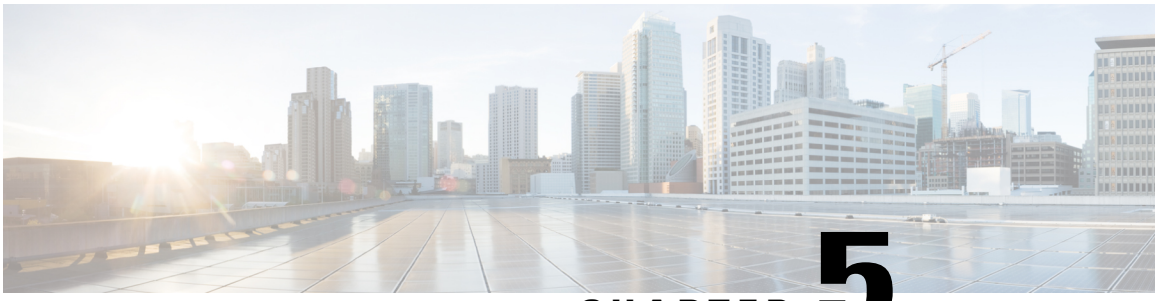
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for Cisco UBE Out-of-dialog OPTIONS Ping for Specified SIP Servers or Endpoints

Feature Name	Releases	Feature Information
Out-of-dialog OPTIONS Ping to Monitor Dial-peers to Specified SIP Servers and Endpoints	15.0(1)M 12.4(22)YB	This feature provides a keepalive mechanism at the SIP level between any number of destinations. The generic heartbeat mechanism allows Cisco UBE to monitor the status of SIP servers or endpoints and provide the option of busying-out associated dial-peer upon total heartbeat failure. In Cisco IOS Release 15.0(1)M, this feature was implemented on the Cisco Unified Border Element. The following command was introduced: **voice-class sip options-keepalive**
Out-of-dialog OPTIONS Ping to Monitor Dial-peers to Specified SIP Servers and Endpoints	Cisco IOS XE Release 3.1S	This feature provides a keepalive mechanism at the SIP level between any number of destinations. The generic heartbeat mechanism allows Cisco UBE to monitor the status of SIP servers or endpoints and provide the option of busying-out associated dial-peer upon total heartbeat failure. In Cisco IOS XE Release 3.1S, this feature was implemented on the Cisco Unified Border Element (Enterprise). The following command was introduced: **voice-class sip options-keepalive**

PART II

Hosted Scenarios Deployment



CHAPTER 5

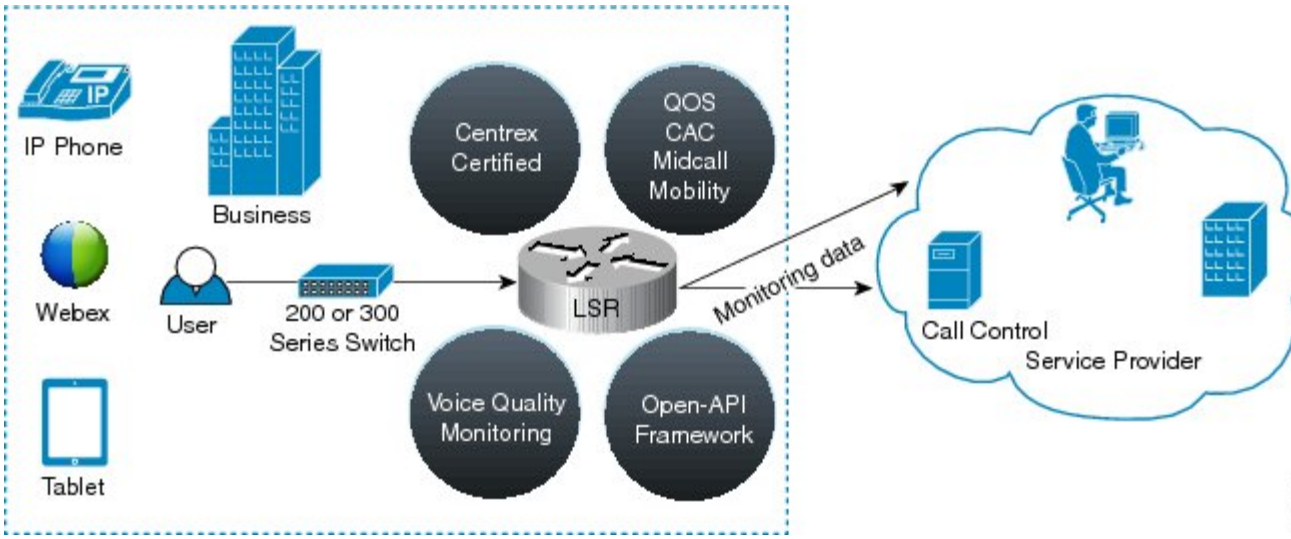
Hosted Scenarios Topology

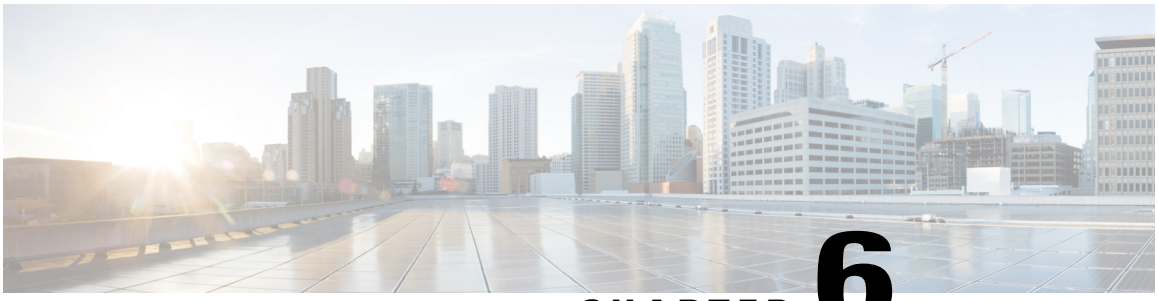
Using NanoCUBE in Hosted Scenarios (lineside)

NanoCUBE supports Hosted or Centrex-based deployments, typically found in Small and Medium Enterprises and Businesses. In these deployments, the Call Control Agents are located in Service Provider network or the Cloud. FXS and SIP end-points are connected to NanoCUBE without any Call Control agent in between them. Hence, NanoCUBE performs lineside termination for endpoints in the enterprise and connects to the Service Provider network or cloud over a SIP trunk. This is achieved by supporting Centrex-certified call flows and passing through Registrations, 401/407, and SUBSCRIBE/NOTIFY type of messages.

Note Some of the hosted deployments would need COR-related (Class of Restrictions) configurations for routing the calls from endpoint to service provider and vice-versa. Please refer to the following link: http://www.cisco.com/en/US/tech/tk652/tk90/technologies_configuration_example09186a008019d649.shtml.

Figure 8: NanoCUBE in Hosted Call Control Scenario





CHAPTER 6

Platform Support for NanoCUBE

Platform Support

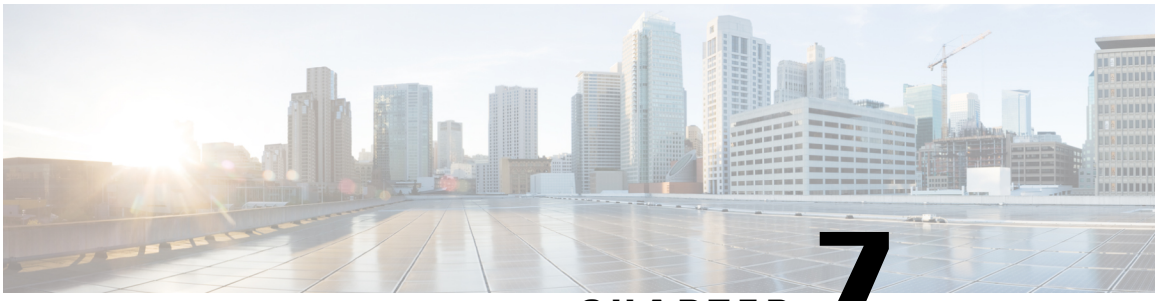
NanoCUBE is supported on various platforms running on Cisco IOS Software Releases. The following table provides details of supported platforms from Cisco IOS Release 15.6(2)T.

Integrated Service Routers		
Cisco Router	**Product Description**	**Default and Minimum Flash Memory**
C881-K9	Cisco 881 Ethernet Security Router	512 MB DRAM
C881-V-K9	Cisco 881 Voice Gateway Router	512 MB DRAM
C886VA-K9	Cisco 886 VDSL/ADSL over ISDN Multi-mode Router	512 MB DRAM
C886VAJ-K9	Cisco 886 VDSL/ADSL Annex J over ISDN Multi-mode Router	512 MB DRAM
C887VA-K9	Cisco 887 VDSL/ADSL over POTS Multi-mode Router	512 MB DRAM
C887VAM-K9	Cisco 887 VDSL/ADSL Annex M over POTS Multi-mode Router	512 MB DRAM
C888EA-K9	Cisco Multimode 888EA G.SHDSL (EFM/ATM) Router	512 MB DRAM
C892FSP-K9	Cisco 892FSP Gigabit Ethernet Security Router	512 MB DRAM
C897VA-M-K9	Cisco 897VA Gigabit Ethernet Security Router	512 MB DRAM

Note While upgrading some of these platforms from data to voice, the system may display the following warning message:

```
%Warning: File not a valid executable for this system
Abort Copy? [confirm]
```

Enter N to proceed with the loading of the image.



CHAPTER 7

Call Admission Control (CAC) Enhancement

The call admission control (CAC) Enhancement feature enhances the CAC mechanism to update the maximum number of incoming or outgoing connections for a dial peer and the bandwidth based on the media flow.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About CAC Enhancement

The CAC enhancement accounts for local call and bandwidth based on the media flowing locally or to the WAN side. In NanoCUBE call flows, if the media flows locally in NanoCUBE then the call is accounted as a local call and the maximum connections and bandwidth values are not updated. The local calls on NanoCUBE do not increment the count of the maximum connections counter.

How to Configure CAC Enhancement

Configuring CAC Enhancement

Perform the following task to exempt local media calls and bandwidth from being updated on a dial peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **max-conn** *number* **exempt-local-media**
5. **session protocol sipv2**
6. **max-bandwidth** *value* **exempt-local-media**
7. **end**

DETAILED STEPS

	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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 2 voip`	Enters dial peer configuration mode.
Step 4	**max-conn** *number* **exempt-local-media** **Example:** `Device(config-dial-peer)# max-conn 10 exempt-local-media`	Sets the maximum connections value for the dial peer to be not updated for local media calls.

	Command or Action	Purpose
Step 5	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Specifies the SIP Version 2 protocol for calls between local and remote routers using the packet network.
Step 6	**max-bandwidth** *value* **exempt-local-media** **Example:** `Device(config-dial-peer)# max-bandwidth 20 exempt-local-media`	Sets the bandwidth threshold value to be not accounted for local media calls.
Step 7	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Verifying CAC Enhancement

Perform this task to verify the configuration for the CAC enhancement. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice** *tag*
3. **show dial-peer voice** *tag*

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
`Device>` **enable**

Step 2 **show dial-peer voice** *tag*
Displays the information for voice dial peers.

Example:

```
Device# show dial-peer voice 302 include | bandwidth/maximum

         bandwidth/maximum = 0/200 Kilo bits per second
```

Step 3 **show dial-peer voice** *tag*

Displays the information for voice dial peers.

Example:

```
Device# show dial-peer voice 302 include | connections/maximum

         connections/maximum = connections/maximum = 0/2
```

Configuration Examples for CAC Enhancement

Example: Configuring the CAC Enhancement

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 2 voip
Device(config-dial-peer)# max-conn 10 exempt-local-media
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# max-bandwidth 20 exempt-local-media
Device(config-dial-peer)# end
```

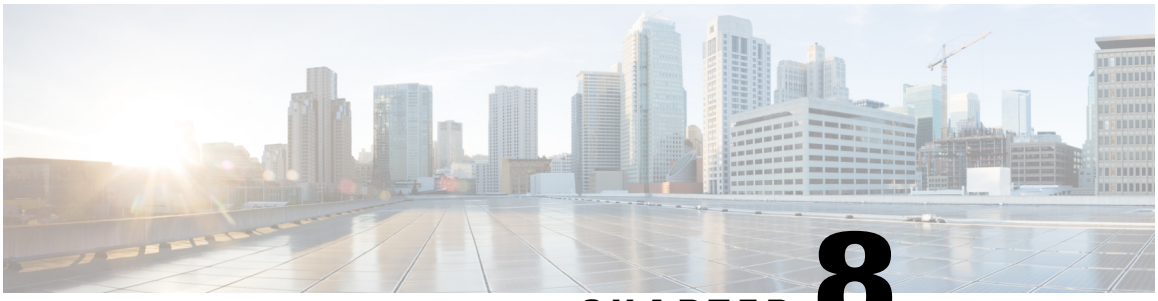
Feature Information for CAC Enhancement

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for CAC Enhancement

Feature Name	Releases	Feature Information
Nano CUBE -- CAC Enhancement	15.3(3)M	The call admission control (CAC) Enhancement feature enhances the CAC mechanism to update the maximum number of incoming or outgoing connections for a dial peer and the bandwidth based on the media flow.



CHAPTER 8

NanoCUBE -- Emergency Number Preemption

The emergency number preemption (also called 911-preemption) feature enables you to configure a list of emergency numbers. When the maximum number of incoming or outgoing connections on a dial-peer is reached, the other non-emergency calls are preempted from the session initiation protocol (SIP) dial-peer, allowing the emergency call to go through.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Emergency Number Preemption

The Emergency Number Preemption feature works only for maximum-connections-based call admission control (CAC) and not bandwidth-based CAC.

Information About Emergency Number Preemption

The emergency number is 911 in the U.S.A., but it may not be the same in other parts of the world; so this emergency number can configured by the user. This feature is triggered by the following two conditions:

- When the SIP dial peer connection reaches the maximum connections configured.
- When the current call's called number matches with the emergency number.

When the emergency preemption feature is triggered, the non-emergency call is preempted and the current emergency call is allowed to go through. If it cannot find another non-emergency call to preempt, the current emergency call will fail.

Note Using this feature, the existing multi-level precedence preemption (MLPP) feature is leveraged, as MLPP can preempt calls.

How to Configure Emergency Number Preemption

Configuring the Emergency Number

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **emergency** *number*
5. **end**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**emergency** *number* **Example:** `Device(conf-voi-serv)# emergency 912 913`	Configures the list of emergency numbers. You can set multiple emergency numbers, which will be set to highest priority if MLPP is configured.
Step 5	**end** **Example:** `Device(conf-voi-serv)# end`	Returns to privileged EXEC mode.

Configuring Preemption and the Maximum Connections on SIP Dial Peer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice mlpp**
4. **preemption voip-trunk sip**
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **max-conn** *number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice mlpp** **Example:** `Device(config)# voice mlpp`	Enters voice MLPP configuration mode that is used for MLPP commands.
Step 4	**preemption voip-trunk sip** **Example:** `Device(config-voice-mlpp)# preemption voip-trunk sip`	Enables preemption for VoIP trunk SIP dial peer.
Step 5	**exit** **Example:** `Device(config-voice-mlpp)# exit`	Exits voice MLPP configuration mode and returns to privileged EXEC mode.
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 30 voip`	Enters dial peer voice configuration mode.
Step 7	**max-conn** *number* **Example:** `Device(config-dial-peer)# max-conn 2`	Configures the maximum number of connections on SIP dial peers.
Step 8	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Verifying Emergency Number Preemption

Perform this task to verify the configuration for the emergency number preemption. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice** *tag*
3. **show voice mlpp voip-trunk sip**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
```
Device> enable
```

Step 2 **show dial-peer voice** *tag*
Displays the information for voice dial peers. The following sample output shows the MLPP SIP queues of calls (last line of the output) for a particular dial peer.

Example:
```
Device# show dial-peer voice 1

VoiceOverIpPeer1
        peer type = voice, system default peer = FALSE, information type = voice,
        description = `',
        tag = 1, destination-pattern = `',
        voice reg type = 0, corresponding tag = 0,
        allow watch = FALSE
        answer-address = `', preference=0,
        CLID Restriction = None
        CLID Network Number = `'
        CLID Second Number sent
        CLID Override RDNIS = disabled,
        rtp-ssrc mux = system
        source carrier-id = `', target carrier-id = `',
        source trunk-group-label = `',  target trunk-group-label = `',
        numbering Type = `unknown'
        group = 1, Admin state is up, Operation state is down,
        incoming called-number = `', connections/maximum = 0/2,
        bandwidth/maximum = 0/unlimited,
        DTMF Relay = disabled,
        modem transport = system,
        URI classes:
            Incoming (Request) =
            Incoming (Via) =
            Incoming (To) =
            Incoming (From) =
            Destination =
        Destination route-string = None
        huntstop = disabled,
        in bound application associated: 'DEFAULT'
        out bound application associated: ''
        dnis-map =
        permission :both
        incoming COR list:maximum capability
        outgoing COR list:minimum requirement
        outgoing LPCOR:
        Translation profile (Incoming):
        Translation profile (Outgoing):
        incoming call blocking:
```

```
translation-profile = `'
disconnect-cause = `no-service'
advertise 0x40 capacity_update_timer 25 addrFamily 4 oldAddrFamily 4
mailbox selection policy: none
type = voip, session-target = `',
technology prefix:
settle-call = disabled
ip media DSCP = ef, ip media rsvp-pass DSCP = ef
ip media rsvp-fail DSCP = ef, ip signaling DSCP = af31,
ip video rsvp-none DSCP = af41,ip video rsvp-pass DSCP = af41
ip video rsvp-fail DSCP = af41,
ip defending Priority = 0, ip preemption priority = 0
ip policy locator voice:
ip policy locator video:
UDP checksum = disabled,
IPv6 UDP checksum = disabled
session-protocol = sipv2, session-transport = system,
req-qos = best-effort, acc-qos = best-effort,
req-qos video = best-effort, acc-qos video = best-effort,
req-qos audio def bandwidth = 64, req-qos audio max bandwidth = 0,
req-qos video def bandwidth = 384, req-qos video max bandwidth = 0,
RTP dynamic payload type values: NTE = 101
Cisco: NSE=100, fax=96, fax-ack=97, dtmf=121, fax-relay=122
       CAS=123, TTY=119, ClearChan=125, PCM switch over u-law=0,
       A-law=8, GSMAMR-NB=117 iLBC=116, AAC-ld=114, iSAC=124
       lmr_tone=0, nte_tone=0
       h263+=118, h264=119
       G726r16 using static payload
       G726r24 using static payload
RTP comfort noise payload type = 19
fax rate = voice,   payload size =  20 bytes
fax protocol = system
fax-relay ecm enable
Fax Relay ans treatment disabled
Fax Relay ans enabled
Fax Relay SG3-to-G3 Enabled (by system configuration)
fax NSF = 0xAD0051 (default)
codec = g729r8,   payload size =  20 bytes,
video codec = None
voice class codec = `'
voice class sip session refresh system
voice class sip rsvp-fail-policy voice post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy voice post-alert optional keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert optional keep-alive interval 30
text relay = disabled
Media Setting = forking (disabled) flow-through (global)stats-disconnect (disabled)
Expect factor = 10, Icpif = 20,
Playout Mode is set to adaptive,
Initial 60 ms, Max 1000 ms
Playout-delay Minimum mode is set to default, value 40 ms
Fax nominal 300 ms
Max Redirects = 1, signaling-type = cas,
VAD = enabled, Poor QOV Trap = disabled,
Source Interface = NONE
voice class sip url = system,
voice class sip tel-config = system,
voice class sip rel1xx = system,
tvoice class sip outbound-proxy = system,
voice class sip asserted-id = system,
voice class sip privacy = system,
voice class sip e911 = system,
voice class sip history-info = system,
voice class sip pass-thru headers = system,
voice class sip pass-thru subscribe-notify-events = system,
voice class sip pass-thru content unsupp = system,
voice class sip pass-thru content sdp = system,
voice class sip copy-list = system,
voice class sip anat = system,
voice class sip g729 annexb-all = system,
voice class sip early-offer forced = system,
```

```
         voice class sip negotiate cisco = system,
         voice class sip reset timer expires 183 = system,
         voice class sip block 180 = system,
         voice class sip block 181 = system,
         voice class sip block 183 = system,
         voice class sip preloaded-route = system,
         voice class sip random-contact = system,
         voice class sip random-request-uri validate = system,
         voice class sip call-route p-called-party-id = system,
         voice class sip call-route history-info = system,
         voice class sip call-route url = system,
         voice class sip privacy-policy send-always = system,
         voice class sip privacy-policy passthru = system,
         voice class sip privacy-policy strip history-info = system,
         voice class sip privacy-policy strip diversion = system,
         voice class sip bandwidth audio  = system,
         voice class sip bandwidth video  = system,
         voice class sip error-code-override options-keepalive failure = system,
         voice class sip error-code-override call spike failure = system,
         voice class sip error-code-override cac-bandwidth failure = system,
         voice class sip encap clear-channel  = system,
         voice class sip send 180 sdp = system,
         voice class sip map resp-code 181 = system,
         voice class sip bind control = system,
         voice class sip bind media = system,
         voice class sip registration passthrough = System
         voice class sip nat mode = System
         voice class sip conn reuse = System
         voice class sip authenticate redirecting-number = system,
         voice class sip referto-passing = system,
         voice class sip extension = system,
         voice class phone proxy name: None
         voice class phone proxy config: N/A
         redirect ip2ip = disabled
         local peer = false
         probe disabled,
         Secure RTP: system (use the global setting)
         mobility=0, snr=, snr_noan=, snr_delay=0, snr_timeout=0
         snr calling-number local=disabled, snr ring-stop=disabled, snr answer-too-soon timer=0
rtcp_keepalive = system

         voice class perm tag = `'
         Time elapsed since last clearing of voice call statistics never
         Connect Time = 0, Charged Units = 0,
         Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
         Accepted Calls = 0, Refused Calls = 0,
         Bandwidth CAC Accepted Calls = 0, Bandwidth CAC Refused Calls = 0,
         Last Disconnect Cause is "",
         Last Disconnect Text is "",
         Last Setup Time = 0.
         Last Disconnect Time = 0.
       MLPP SIP Queues:  0-1 1-0 2-0 3-0 4-0 5-0 6-0 7-0 8-1 9-0
```

Step 3 **show voice mlpp voip-trunk sip**

Displays information about the SIP MLPP call queues. In the sample output, (A) means ACTIVE when a call ID is non-local call and (I) means INACTIVE when callID is a local call. During preemption, INACTIVE calls are skipped. LEVEL 0 means internal precedence level is 0.

Example:

```
Device# show voice mlpp voip-trunk sip

 dial-peer voice 3 voip
 MLPP Preempted Count = 0.
 LEVEL 0: 23(A) 22(A) 24(I)
 LEVEL 8: 28(A)
```

Troubleshooting Tips

Use the **debug voip mlpp** command to troubleshoot the Emergency Number Preemption feature.

Configuration Examples for Emergency Number Preemption

Example: Configuring Emergency Number

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# emergency 912 913
Device(conf-voi-serv)# end
```

Example: Configuring Preemption and the Maximum Connections on SIP Dial Peer

```
Device> enable
Device# configure terminal
Device(config)# voice mlpp
Device(config-voice-mlpp)# preemption voip-trunk sip
Device(config-voice-mlpp)# exit
Device(config)# dial-peer voice 30 voip
Device(config-dial-peer)# max-conn 2
Device(config-dial-peer)# end
```

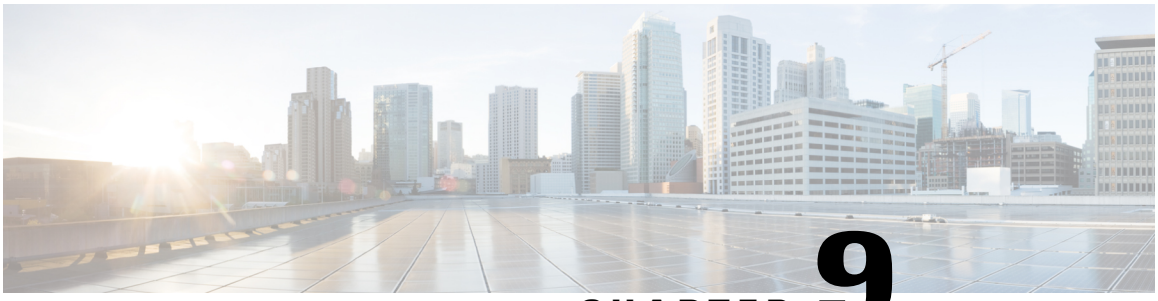
Feature Information for Emergency Number Preemption

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Emergency Number Preemption

Feature Name	Releases	Feature Information
Nano CUBE (Emergency Number Preemption)	15.3(3)M	The emergency number preemption (also called 911-preemption) feature enables you to configure a list of emergency numbers. When the maximum number of incoming or outgoing connections on a dial-peer is reached, the other non-emergency calls are preempted from the session initiation protocol (SIP) dial-peer, allowing the emergency call to go through.



CHAPTER 9

Nano CUBE SUBSCRIBE-NOTIFY Passthrough

The SUBSCRIBE-NOTIFY mechanism is used for implementation of features such as Message Waiting Indication (MWI), Shared Call Appearance, Multiple Caller Appearance, Busy Lamp Field, and so on.

In Nano CUBE, the SUBSCRIBE-NOTIFY framework on Unified Communications (UC) products supports the following:

- Configurable and Selective Passthrough of SUBSCRIBE and NOTIFY transactions from phones with the normalization required for address or topology hiding and dialog content updates for “dialog” event subscription.
- Survivability mode handling of incoming SUBSCRIBE request for critical events.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for SUBSCRIBE-NOTIFY Passthrough

- The SUBSCRIBE-NOTIFY passthrough framework can only pass through events when there is a one-to-one association between the incoming request and the outgoing location to which the request has be sent out. This means that either:
  - There should be an outbound dial-peer identified for the request received.

    or
  - The outbound target for the request could be only a single registrar.
- The following use cases are not supported:
  - SUBSCRIBE-NOTIFY passthrough with hunting of outbound dial-peers to which the subscribe or notify requests need to be sent.
  - SUBSCRIBE-NOTIFY passthrough where an inbound dial-peer has peer-to-peer mode of registration passthrough enabled with more than one registrar (there will be no forking of Subscribe-Notify Requests)
  - Local subscribe handling of unsupported events when the remote registrar is unavailable. Local subscribe handling is only applicable to cases where the inbound dial-peer matching the subscribe has registration passthrough enabled with “local-fallback.”

Information About SUBSCRIBE-NOTIFY Passthrough

The key attributes of the SUBSCRIBE-NOTIFY Passthrough feature are as follows:

- Message Passthrough Application (MPA)—The SIP MPA handles SUBSCRIBE-NOTIFY passthrough. This application maintains subscribe dialogs, and references the dial-peer database and registration passthrough configurations to route the initial SUBSCRIBE and unsolicited NOTIFY requests.
- Header Passthrough—All the non-mandatory headers in SUBSCRIBE-NOTIFY requests and responses are passed through from one endpoint to the other.
- Content Passthrough—The content bodies in SUBSCRIBE-NOTIFY requests are passed through transparently from one endpoint to the other.
- “Dialog” Event Content Manipulation—The content in the NOTIFY body for a dialog event is updated before passthrough when the dialog is maintained by the Nano CUBE.
- Passthrough Configuration and Filtering—SUBSCRIBE-NOTIFY passthrough is configurable globally as well as under dial-peer, and can be configured for selected events using the configuration of an event list.
- Error Passthrough for SUBSCRIBE-NOTIFY Requests—When an error is received for a SUBSCRIBE-NOTIFY request, the error is passed through to the peer with the relevant headers.
- Backward Compatibility—The SIP MPA has the highest priority when SUBSCRIBE-NOTIFY passthrough is enabled. If passthrough is not enabled (either for all events or for a specific event), the current applications will control the incoming requests and responses.

- 401/407 Error Message Passthrough—SIP message 401/407 is sent by the user-agent server (UAS) or end device to challenge messages like INVITE/REFER/SUBSCRIBE and request for endpoint credentials information. Nano CUBE does not store endpoint credential information to act on behalf of phone or endpoint. To enable the passthrough of 401/407, you can enable the **error passthru** command at the global level. The messages 401/407 are in passthru mode for INVITE/REFER/SUBSCRIBE.

SUBSCRIBE-NOTIFY Passthrough Request Routing

The first step of request or response routing is for Nano CUBE to determine whether or not the request has to be passed through. When a new SUBSCRIBE or unsolicited NOTIFY request arrives, its headers are used to match an incoming dial-peer. If the incoming dial-peer has SUBSCRIBE-NOTIFY Passthrough (SNPT) enabled or if there is no incoming dial-peer and global SNPT is enabled for that event, then the request is handed off to be passed through. For solicited subscriptions, the passthrough check is applicable only to the initial SUBSCRIBE request; subsequent requests or responses are not checked and will be routed based on updated dialog parameters.

The second step is to determine the outbound destination of the SUBSCRIBE or unsolicited NOTIFY request.

- First Pass: Outbound dial-peer match—An outbound VoIP dial-peer is first matched based on the request headers (From, To, and Via), the Subscriber Number (userid in the To header), and the incoming dial-peer Class of Restrictions (CoR) if any. If there is a match, the request is routed to the session target.
- Second Pass: Configured registrar for registration passthrough in peer-to-peer mode—If no outbound dial-peer is found and the incoming dial-peer has registration passthrough enabled in static (peer-to-peer) mode with a single registrar configured, then the request is routed to the registrar address.
- Third Pass: Configured registrar for registration passthrough in end-to-end mode—If no outbound dial-peer is found and the incoming dial-peer has registration passthrough enabled in dynamic (end-to-end) mode:
  - If the request Uniform Resource Identifier (URI) has the CUBE IP address, the request is routed to the configured registrar if only a single registrar is configured.
  - If the request URI has a non-CUBE IP address, then the request is routed to that IP address.
- Fourth Pass: Request URI-based routing—If no outbound dial-peer is found and no registration passthrough is configured, the request URI is used to route the request if it does not point to the CUBE's IP address.

SUBSCRIBE-NOTIFY Passthrough Survivability Mode

In survivability mode, the CUBE could encounter the following scenarios:

- When the CUBE receives a line-seize (event) subscribe in survivability mode, it checks the line-seize queue to see if another phone has already seized the same line; if not, CUBE accepts the subscription, sends a NOTIFY response with State = Active, and starts the timer for expiration. In survivability mode, SUBSCRIBE received for any event other than line-seize is rejected.
- If another phone has already subscribed for the line, CUBE sends a 200 OK (request successful) response for the new subscribe, but a final NOTIFY to indicate that the subscription has been terminated.
- If the subscription timer expires without re-subscription from the phone, CUBE sends a final NOTIFY to remove the subscription.

- If a subscription is created in active mode, but re-subscriptions or unsubscriptions are received in survivability mode, then CUBE returns an error for this subscription.

How to Configure SUBSCRIBE-NOTIFY Passthrough

Configuring an Event List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-event** *number*
4. **event** *name*
5. **end**

DETAILED STEPS

	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 class sip-event** *number* **Example:** `Device(config)# voice class sip-event 1`	Enters voice class configuration mode and configures the list of events to be passed through.
Step 4	**event** *name* **Example:** `Device(config-class)# event message-summary`	Adds the name of the event to be added to the event list.
Step 5	**end** **Example:** `Device(config-class)# end`	Returns to privileged EXEC mode.

Configuring SUBSCRIBE-NOTIFY Event Passthrough Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **pass-thru subscribe-notify-events** *tag*
6. **end**

DETAILED STEPS

	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 class voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters voice service SIP configuration mode.
Step 5	**pass-thru subscribe-notify-events** *tag* **Example:** `Device(conf-serv-sip)# pass-thru subscribe-notify-events 1`	Configures SUBSCRIBE-NOTIFY passthrough event with the SIP event list tag number to be linked globally. • You can use the **pass-thru subscribe-notify-events all** command to configure passthrough for all SUBSCRIBE-NOTIFY events.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Configuring SUBSCRIBE-NOTIFY Event Passthrough at the Dial-Peer Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip pass-thru subscribe-notify-events** *tag*
5. **end**

DETAILED STEPS

	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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 123 voip`	Enters dial peer voice configuration mode.
Step 4	**voice-class sip pass-thru subscribe-notify-events** *tag* **Example:** `Device(config-dial-peer)# voice-class sip pass-thru subscribe-notify-events 1`	Configures SUBSCRIBE-NOTIFY passthrough event with the SIP event list tag number to be linked globally. • You can use the **voice-class sip pass-thru subscribe-notify-events all** command to configure passthrough for all SUBSCRIBE-NOTIFY events.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Verifying SUBSCRIBE-NOTIFY Passthrough

Perform this task to verify the configuration for SUBSCRIBE-NOTIFY Passthrough and to verify the subscriptions created. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice** *number* **| inc pass**
3. **show subscription asnl session active**
4. **show subscription sip**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
`Device>` **enable**

Step 2 **show dial-peer voice** *number* **| inc pass**
Displays the information for voice dial peers. The following sample output shows the configured SUBSCRIBE-NOTIFY passthrough event for a particular dial peer.

Example:
`Device#` **show dial-peer voice 123 | inc pass**

```
ip media DSCP = ef, ip media rsvp-pass DSCP = ef
ip video rsvp-none DSCP = af41,ip video rsvp-pass DSCP = af41
voice class sip pass-thru headers = system,
voice class sip pass-thru subscribe-notify-events = system,
voice class sip pass-thru content unsupp = system,
voice class sip pass-thru content sdp = system,
voice class sip privacy-policy passthru = system,
voice class sip registration passthrough = System
voice class sip referto-passing = system
```

Step 3 **show subscription asnl session active**

Displays information about Application Subscribe/Notify Layer (ASNL)-based and non-ASNL-based SIP subscriptions.

Example:

```
Device# show subscription asnl session active

ASNL Active Subscription Records Details:
=========================================
Number of active subscriptions: 1
URL: sip:user@10.7.104.88
  Event Name : stress
  Session ID : 8
  Expiration Time : 50 seconds
  Subscription Duration : 5 seconds
  Protocol : ASNL_PROTO_SIP
  Remote IP address : 10.7.104.88
  Port : 5060
  Call ID : 5
  Total Subscriptions Sent : 1
  Total Subscriptions Received: 0
  Total Notifications Sent : 0
  Total Notifications Received  : 2
  Last response code : ASNL_NOTIFY_RCVD
  Last error code : ASNL_NONE
  First Subscription Time : 10:55:12 UTC Apr 9 2000
  Last Subscription Time : 10:55:12 UTC Apr 9 2000
  First Notify Time : 10:55:12 UTC Apr 9 2000
  Last Notify Time : 10:55:17 UTC Apr 9 2000
  Application that subscribed : stress
  Application receiving notification: stress
```

Step 4 **show subscription sip**

Displays information about ASNL-based and non-ASNL-based SIP subscriptions.

Example:

```
Device# show subscription sip

ASNL Active Subscription Records Summary:
=========================================

Number of active subscriptions: 2

SubId      CallId     Proto            URL                                Event
-----      ------     -----            ---                                -----
1          N/A        ASNL_PROTO_SIP  "Plutus" <sip:1111003@primary>   all
2          N/A        ASNL_PROTO_SIP  sip:1111003@primaryappserver1    as-feature

Client    EXPIRES(sec)    EVENT
======    ============    ======
1111003         0          as-feature-event
Client    EXPIRES(sec)    EVENT
======    ============    ======
1234            0          message-summary
```

Troubleshooting Tips

Use the following commands to troubleshoot SUBSCRIBE-NOTIFY Passthrough:

- **debug mpa events**

- **debug mpa error**
- **debug ccsip messages**
- **debug asnl events**
- **debug asnl error**
- **debug ccsip all**

Configuration Examples for SUBSCRIBE-NOTIFY Passthrough

Example: Configuring an Event List

The following example shows how to configure an event list and add an event to the list of events that have to be passed through.

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-event-list 1
Device(config-class)# event 1 message-summary
Device(config-class)# end
```

Example: Configuring SUBSCRIBE-NOTIFY Event Passthrough Globally

The following example shows how to configure the SUBSCRIBE-NOTIFY passthrough event and link the SIP event list tag number globally.

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# pass-thru subscribe-notify-events 1
Device(conf-serv-sip)# end
```

Example: Configuring SUBSCRIBE-NOTIFY Event Passthrough under a Dial Peer

The following example shows how to configure the SUBSCRIBE-NOTIFY passthrough event and link the SIP event list tag number under a dial peer.

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 123 voip
Device(config-dial-peer)# voice-class sip pass-thru subscribe-notify-events 1
Device(config-dial-peer)# end
```

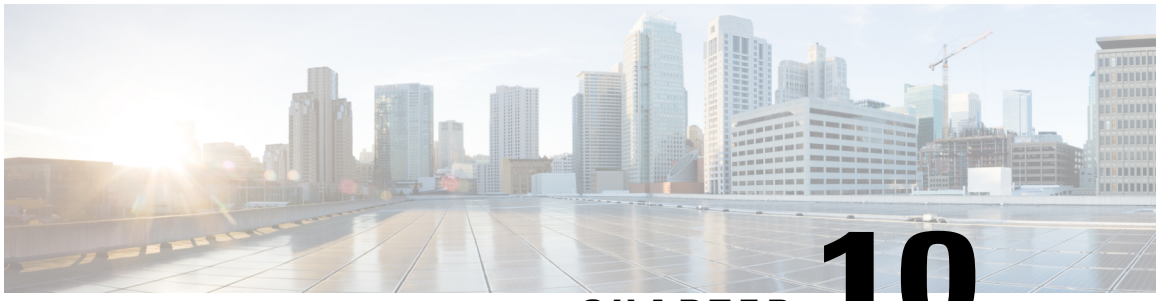
Feature Information for SUBSCRIBE-NOTIFY Passthrough

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for SUBSCRIBE-NOTIFY Passthrough

Feature Name	Releases	Feature Information
Nano CUBE (SUBSCRIBE-NOTIFY Passthrough)	15.3(3)M	The SUBSCRIBE-NOTIFY mechanism is used for implementation of features such as Message Waiting Indication (MWI), Shared Call Appearance, Multiple Caller Appearance, Busy Lamp Field, and so on. In Nano CUBE, the SUBSCRIBE-NOTIFY framework on Unified Communications (UC) products supports the following: • Configurable and Selective Passthrough of SUBSCRIBE and NOTIFY transactions from phones with the normalization required for address or topology hiding and dialog content updates for “dialog” event subscription. • Survivability mode handling of incoming SUBSCRIBE request for critical events.



CHAPTER 10

Nano CUBE - INFO DTMF Relay

The INFO dual-tone multi-frequency (DTMF) relay feature in Nano CUBE provides support for INFO-INFO DTMF relay. This support will enable passing through INFO requests with single DTMF digit information in the Content Body (0- 9, *, #, a, b, c, d) to the peer.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for INFO DTMF Relay

- Only INFO-INFO is supported. Other interworking modes such as INFO-KPML and INFO-NOTIFY are not supported.
- Only one digit can be sent per INFO message. If there are more then one digits in the “Signal=” line from the incoming INFO, only the first digit is accepted.

Information About INFO DTMF Relay

The INFO DTMF relay feature is enabled on the NanoCUBE when both the inbound and outbound SIP dial peers are configured using DTMF-relay SIP-INFO configuration and negotiated end-to-end using the Allow and Accept headers.

When the NanoCUBE provisioned with SIP-INFO DTMF Relay option receives an INVITE/200 OK (for INVITE) with the Allow header containing INFO and Accept header containing “application/dtmf-relay”, it advertises its support for INFO DTMF relay using the Allow header and notifies the incoming call leg that INFO DTMF Relay has been negotiated by forwarding the Accept header. If the remote endpoint does not advertise its support for INFO DTMF Relay, the DTMF negotiation proceeds with the consideration that INFO is not one of the supported methods, and another common method will be negotiated. If nothing else matches between the two legs, DTMF negotiation will fallback to in-band mode.

How to Configure INFO DTMF Relay

Configuring INFO-INFO DTMF Relay Passthrough

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **dtmf-relay sip-info**
5. **end**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 300 voip`	Enters dial peer voice configuration mode.
Step 4	**dtmf-relay sip-info** **Example:** `Device(config-dial-peer)# dtmf-relay sip-info`	Configures the INFO-INFO DTMF relay passthrough (via SIP-INFO messages) on the inbound and outbound dial peer.
Step 5	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Verifying INFO DTMF Relay

Perform this task to verify the configuration for the INFO-INFO DTMF Relay Passthrough. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show sip-ua statistics**
3. **show sip-ua calls dtmf-relay sip-info**
4. **show sip-ua history dtmf-relay sip-info**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
`Device>` **enable**

Step 2 **show sip-ua statistics**
Displays response, traffic, and retry Session Initiation Protocol (SIP) statistics.

Example:

```
Device# show sip-ua statistics

SIP Response Statistics (Inbound/Outbound)
    Informational:
     Trying 1/1, Ringing 0/0,
     Forwarded 0/0, Queued 0/0,
     SessionProgress 0/1
    Success:
     OkInvite 0/1, OkBye 1/0,
     OkCancel 0/0, OkOptions 0/0,
     OkSubscibe 0/0, OkNotify 0/0,
     OkInfo 0/1, 202Accepted 0/0
     OkRegister 12/49
    Redirection (Inbound only except for MovedTemp(Inbound/Outbound)) :
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0/0, UseProxy 0,
      AlternateService 0
      Client Error:
      BadRequest 0/0, Unauthorized 0/0,
      PaymentRequired 0/0, Forbidden 0/0,
      NotFound 0/0, MethodNotAllowed 0/0,
      NotAcceptable 0/0, ProxyAuthReqd 0/0,
      ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
      ReqEntityTooLarge 0/0, ReqURITooLarge 0/0,
      UnsupportedMediaType 0/0, BadExtension 0/0,
      TempNotAvailable 0/0, CallLegNonExistent 0/0,
      LoopDetected 0/0, TooManyHops 0/0,
      AddrIncomplete 0/0, Ambiguous 0/0,
      BusyHere 0/0, RequestCancel 0/0,
      NotAcceptableMedia 0/0, BadEvent 0/0,
      SETooSmall 0/0
     Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0,
      PreCondFailure 0/0
     Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NotExistAnywhere 0/0, NotAcceptable 0/0
      Miscellaneous counters:
      RedirectRspMappedToClientErr 0
SIP Total Traffic Statistics (Inbound/Outbound)
      Invite 0/0, Ack 0/0, Bye 0/0,
      Cancel 0/0, Options 0/0,
      Prack 0/0, Comet 0/0,
      Subscribe 0/0, NOTIFY 0/0,
      Refer 0/0, Info 0/0
      Register 49/16
Retry Statistics
      Invite 0, Bye 0, Cancel 0, Response 0,
      Prack 0, Comet 0, Reliable1xx 0, Notify 0
      Register 4, Subscribe 0
SDP application statistics:
Parses: 0, Builds 0
Invalid token order: 0, Invalid param: 0
Not SDP desc: 0, No resource: 0
Last time SIP Statistics were cleared: <never>
```

Step 3 **show sip-ua calls dtmf-relay sip-info**

Displays active SIP calls with INFO DTMF Relay mode.

Example:

```
Device# show sip-ua calls dtmf-relay sip-info

Total SIP call legs:2, User Agent Client:1, User Agent Server:1
```

```
SIP UAC CALL INFO
Call 1
SIP Call ID                : 9598A547-5C1311E2-8008F709-2470C996@172.27.161.122
   State of the call       : STATE_ACTIVE (7)
   Calling Number          : sipp
   Called Number           : 3269011111
   CC Call ID              : 2
No.         Timestamp         Digit           Duration
=========================================================
0    01/12/2013 17:23:25.615  2                   250
1    01/12/2013 17:23:25.967  5                   300
2    01/12/2013 17:23:26.367  6                   300

Call 2
SIP Call ID                : 1-29452@172.25.208.177
   State of the call       : STATE_ACTIVE (7)
   Calling Number          : sipp
   Called Number           : 3269011111
   CC Call ID              : 1
No.         Timestamp         Digit           Duration
=========================================================
0    01/12/2013 17:23:25.615  2                   250
1    01/12/2013 17:23:25.967  5                   300
2    01/12/2013 17:23:26.367  6                   300

   Number of SIP User Agent Client(UAC) calls: 2

SIP UAS CALL INFO
Call 1
SIP Call ID                : 1-29452@172.25.208.177
   State of the call       : STATE_ACTIVE (7)
   Calling Number          : sipp
   Called Number           : 3269011111
   CC Call ID              : 1
No.         Timestamp         Digit           Duration
=========================================================
0    01/12/2013 17:23:25.615  2                   250
1    01/12/2013 17:23:25.967  5                   300
2    01/12/2013 17:23:26.367  6                   300

Call 2
SIP Call ID                : 9598A547-5C1311E2-8008F709-2470C996@172.27.161.122
   State of the call       : STATE_ACTIVE (7)
   Calling Number          : sipp
   Called Number           : 3269011111
   CC Call ID              : 2
No.         Timestamp         Digit           Duration
=========================================================
0    01/12/2013 17:23:25.615  2                   250
1    01/12/2013 17:23:25.967  5                   300
2    01/12/2013 17:23:26.367  6                   300

   Number of SIP User Agent Server(UAS) calls: 2
```

Step 4 **show sip-ua history dtmf-relay sip-info**

Displays SIP call history with specific DTMF Relay mode.

Example:

```
Device# show sip-ua history dtmf-relay sip-info

Call 1
SIP Call ID          : 1-29452@172.25.208.177
Calling Number       : sipp
Called Number        : 3269011111
CC Call ID           : 1
DTMF Relay Mode      : sip-info
```

```
No.      Timestamp              Digit         Duration
=======================================================
0    01/12/2013 17:23:25.615  2    250
1    01/12/2013 17:23:25.967  5    300
2    01/12/2013 17:23:26.367  6    300
Call 2
SIP Call ID          : 9598A547-5C1311E2-8008F709-2470C996@172.27.161.122
Calling Number       : sipp
Called Number        : 3269011111
CC Call ID           : 2
DTMF Relay Mode      : sip-info

No.      Timestamp              Digit         Duration
=======================================================
0    01/12/2013 17:23:25.615  2    250
1    01/12/2013 17:23:25.967  5    300
2    01/12/2013 17:23:26.367  6    300
```

Configuration Examples for INFO DTMF Relay

Example: Configuring INFO DTMF Relay Passthrough

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 300 voip
Device(config-dial-peer)# dtmf-relay sip-info
Device(config-dial-peer)# end
```

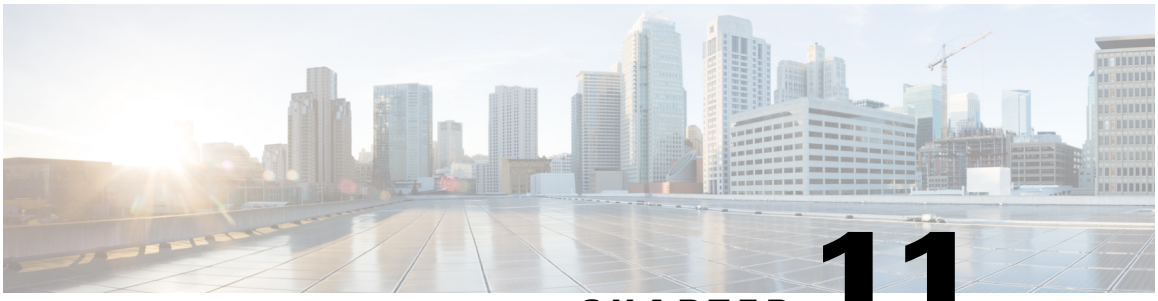
Feature Information for INFO DTMF Relay

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for INFO DTMF Relay

Feature Name	Releases	Feature Information
Nano CUBE (INFO DTMF Relay)	15.3(3)M	The INFO dual-tone multi-frequency (DTMF) relay feature in Nano CUBE provides support for INFO-INFO DTMF relay. This support will enable passing through INFO requests with single DTMF digit information in the Content Body (0-9, *, #, a, b, c, d) to the peer.



CHAPTER 11

Survivability Enhancements

The Survivability Enhancements feature on the Nano CUBE is used to:

- Monitor the WAN status periodically from the Nano CUBE.
- Route calls and handle line-seize subscriptions locally when the WAN link is down.
- Synchronize the registrations with the server when the WAN link is up.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

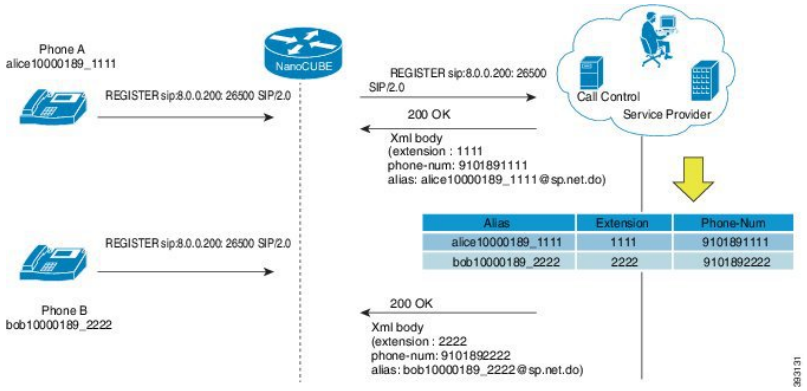
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Survivability Enhancements

Registration through Alias Mapping

The following illustration shows how a phone (with alias mapping) registers to the service provider via Nano CUBE

Figure 9: SIP Phone Registration



The AOR sent in the REGISTER is an alias which is mapped to an extension and/or phone number by the service provider. The service provider returns the mapping details in the 200 OK response sent to the REGISTER. Nano CUBE provides the ability to cache the alias mapping details in its call routing database. When a call is made from the phone, the Request-URI of the INVITE contains the dialed number (short extension or phone number).

If WAN is up, Nano CUBE will always route the INVITE sent from the phone to the service provider without looking up at the alias mapping cache.

If WAN or the service provider is down, that is, in survivability mode, Nano CUBE will route the INVITE locally by looking up at the alias mapping cache.

Alias Mapping—Supported Methods

1 When the service provider returns the mapping details in the 200 OK message of the REGISTER in the following predefined format:

Alias	Extension	Phone
alice10000189_1111	1111	10000189

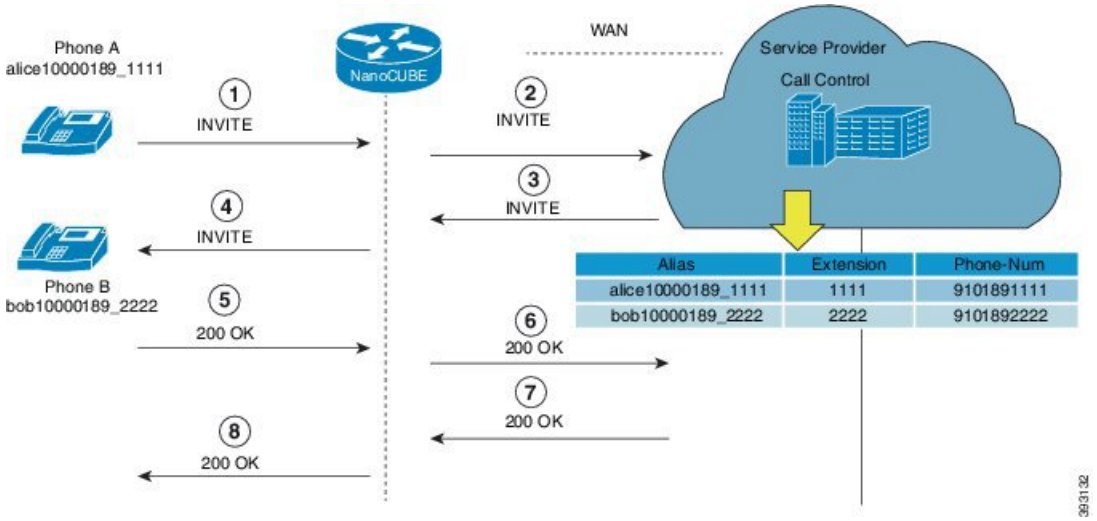
2 The short extension or phone number is embedded in the AOR of the REGISTER. For example, AOR is alice10000189_1111 and the short extension is 1111.

An inbound sip profile can be applied to the REGISTER which extracts the extension part from the AOR and adds a X-CISCO-EXTENSION header.

Nano CUBE when WAN is UP

The following illustration provides an example as to how a typical phone makes a call to another local phone registered in the same server when WAN or the registrar server is up in a typical hosted deployment. The circled numbers in the image indicate the numerical order in which the sequence occurs.

Figure 10: WAN Link is UP - Nano CUBE Deployment



The call flow scenario is as follows: Phone A initiates a call to the Phone B registered to the same server.

1. Phone A sends an initial INVITE request to Phone B to participate in a call session via Nano CUBE.
2. Nano CUBE sends this INVITE to the service provider.
3. The service provider in turn sends the INVITE to Nano CUBE. Since the WAN link is up, the service provider maps details of the user from the register server and provides details of the user, for example, alias of the user, short extension number, and phone number.
4. Nano CUBE sends INVITE with all the above mentioned information to Phone B.
5. Phone B sends a 200 OK response to Nano CUBE for the received INVITE.
6. Nano CUBE sends a 200 OK answer to the service provider.
7. The service provider responds to Nano CUBE with a 200 OK answer.
8. A final 200 OK response is sent to Phone A by Nano CUBE and the call is established between Phone A and Phone B.

Example: Normal Mode (WAN is Up in P2P Mode)

```
CUBE# show sip-ua registration passthrough status

CallId    DirectoryNum   peer     mode   In-Exp      reg-I   Out-Exp   survival
=======   ==========     =====    ====   ========    ====    ====      =========
21        NCPhone1006    1        p2p    135 /144     1       144        normal
================================================================================
```

Example: Normal Mode (WAN is Up in E2E Mode)

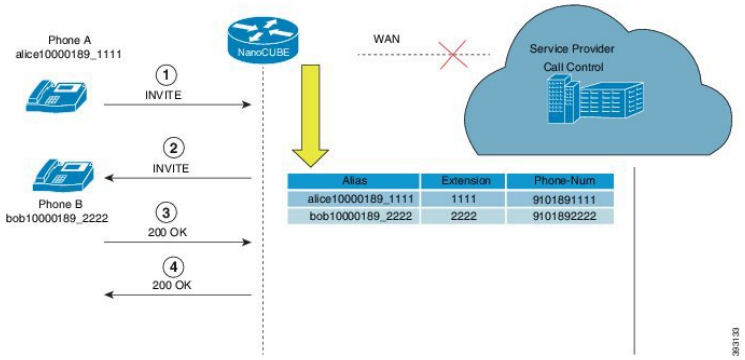
```
CUBE# show sip-ua registration passthrough status

CallId      DirectoryNum    peer    mode      In-Exp       reg-I     Out-Exp   survival
=======     ============    ====    =====     =====        =====     =======   ========
14574       NCPhone1006     301     e2e       117 /120      --       120       normal
==============================================================================
```

Nano CUBE Survivability when WAN is Down

In survivability mode, Nano CUBE provides end-to end telephony services when access to the centralized servers is interrupted because of a WAN outage or other factors, like the server being down.

The following illustration shows how a call is established between two end points when WAN link is down during survivability by directly dialing into an extension.

Figure 11: Nano CUBE Survivability when WAN is Down



Earlier, when WAN was down, User A could only contact User B using either the alias or the user-id of User B, and not using their extensions or phone numbers.

Now, in the event the WAN link or registration server is down, when a local call is made, INVITE is sent to Nano CUBE. Nano CUBE maps the details of the user like extension number and phone-number stored during registration. Local phones can now be reached on their short extensions or phone numbers by similar phones subscribed to the server through the same Nano CUBE.

It is possible to register multiple contacts for a single AOR; however, if multiple contacts are registered for a single subscriber, the Nano CUBE uses only the topmost registered contact to deliver the call to that subscriber. For this reason, multiple contacts are not supported.

Example: Survivability Mode in P2P (regsync mode) when WAN is Down

```
CUBE# show sip-ua registration passthrough status

CallId   DirectoryNum    peer    mode    In-Exp     reg-I  Out-Exp   survival
=======  ========= ============  =====   =========  =====  =======   ========
38       NCPhone1008     1       p2p     3595 /3600   1      3600      regsync
==============================================================================
```

Example: Survivability Mode in E2E (local fallback mode) when WAN is Down

```
CUBE# show sip-ua registration passthrough status

CallId   DirectoryNum    peer    mode   In-Exp   reg-I  Out-Exp  survival
=======  ============    ======  ====   =======  =====  =======  ========
70       NCPhone1006     1       e2e    35 /70    --     0       locfall
==============================================================================
```

```
CallId   DirectoryNum    peer    mode   In-Exp   reg-I  Out-Exp  survival
=======  ============    ======  ====   =======  =====  =======  ========
513      NCPhone1008     1       e2e    40 /70    --     0       locfall
```

Different Modes of Survivability Enhancements

The survivability feature addresses the following issues:

1. When a WAN link or registrar server comes up, it needs to wait till each SIP phone sends the REGISTER message to the server, so that outside phones can reach that phone.
2. If the phone register timer setting is too large, the outside phone needs to wait that much time to reach that phone, after a link flap.
3. If the phone register timer setting is too small, it will flood the WAN link.
4. When the WAN link or registrar server is down, local calls cannot be made.

There are two ways to address these issues:

- Local fallback
- Registration synchronization

Local Fallback

- Nano CUBE does not need to configure credentials, as the phones will trigger registration. Although Nano CUBE receives REGISTER messages for each phone every 5 minutes; for example, it will throttle and send REGISTER messages every 1 hour to the registrar server, avoiding high WAN bandwidth usage. This will address the issues 1, 2, and 3.
- In normal operation when the WAN link or registrar server is up, the phone's primary server URL is the registrar server (E2E) registration.
- "OPTIONS ping" is used to monitor the registrar server link status. When the detected link is down, Nano CUBE will reply with a 500 message and when the phone receives this message, it will send the REGISTER message to Nano CUBE, which is the secondary server (P2P registration). Nano CUBE will reply with a 200 OK message to P2P registration when the link is down. The dial-peer will keep dynamic registrar session target and the local call will not fail. This will address issue 4.

Registration Synchronization

- If you configure the phones to send REGISTER messages every 1 hour (to help alleviate the WAN link), the NanoCUBE uses the credentials configured to respond to registrar server authentication challenge. This addresses issue 3.
- When the WAN link or registration server is down (detected by OPTIONS ping), the NanoCUBE keeps the registration database of the SIP phones previously registered successfully, and it does not send REGISTER messages out; NanoCUBE replies with a 200 OK message and dial-peer will keep the dynamic registrar session target. The local call will not fail, addressing issue 4.
- When the registrar link is up after link flap, the NanoCUBE sends REGISTER message for each phone that was earlier successfully registered to the registrar server. This is throttled to avoid bulk REGISTER messages flooding WAN link as well as the registrar. This addresses issues 1 and 2.

How to Configure Survivability Enhancements

Configuring Local Fallback or Registration Synchronization Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **registration passthrough local-fallback** *tag*
6. **end**

DETAILED STEPS

	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 voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters voice service SIP configuration mode.
Step 5	**registration passthrough local-fallback** *tag* **Example:** `Device(conf-serv-sip)# registration passthrough local-fallback 10`	Configures SIP registration passthrough for local fallback mode; this will locally respond to REGISTER in p2p mode when WAN is down. The *tag* is the WAN link or registrar server dial-peer tag.

	Command or Action	Purpose
		• To configure the registration sync mode, you can use the **registration passthrough reg-sync** *tag* command. Use the **static** keyword to set the phone URL to p2p registration.
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Configuring Local Fallback or Registration Synchronization on a Dial Peer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip registration passthrough local-fallback** *tag*
5. **end**

DETAILED STEPS

	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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 4 voip`	Enters dial peer VoIP configuration mode.
Step 4	**voice-class sip registration passthrough local-fallback** *tag*	Configures SIP registration passthrough for local fallback mode; this will locally respond to REGISTER in p2p mode

	Command or Action	Purpose
	Example: `Device(config-dial-peer)# voice-class sip registration passthrough local-fallback 10`	when WAN is down. The *tag* is the WAN link or registrar server dial-peer tag. • To configure the registration sync mode, you can use the **voice-class sip registration passthrough reg-sync** *tag* command.
Step 5	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Configuring OPTIONS Ping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip options-keepalive up-interval** *value* **down-interval** *value*
5. **end**

DETAILED STEPS

	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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 3 voip`	Enters dial peer configuration mode.

	Command or Action	Purpose
Step 4	**voice-class sip options-keepalive up-interval** *value* **down-interval** *value* **Example:** `Device(config-dial-peer)# voice-class sip options-keepalive up-interval 120 down-interval 120`	Configures OPTIONS keepalive timer interval for DOWN and UP endpoints.
Step 5	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Configuring Registration Timer

Perform the following task to configure the registration timer in the NanoCUBE rather than on all SIP phones.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **registrar server expires max** *value* **min** *value*
6. **end**

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters voice service SIP configuration mode.
Step 5	**registrar server expires max** *value* **min** *value* **Example:** `Device(conf-serv-sip)# registrar server expires max 300 min 200`	Configures the maximum and minimum time (in seconds) for the registration expiry in NanoCUBE. • If the phone sends expiry time as 600 seconds, then the NanoCUBE will reply with 200 OK message and expiry time 300 seconds, and the phone will resend with expiry 300.
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Configuring the REGISTER Message Throttling in Nano CUBE

Perform the following task to throttle the REGISTER message in Nano CUBE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **registration passthrough rate-limit expires** *value* **local-fallback** *tag*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• 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 voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters voice service SIP configuration mode.
Step 5	**registration passthrough rate-limit expires** *value* **local-fallback** *tag* **Example:** `Device(conf-serv-sip)# registration passthrough rate-limit expires 3600 local-fallback 3`	Configures the SIP registration passthrough rate-limit expiry value for local-fallback (e2e). Although Nano CUBE receives the REGISTER message every 5 minutes (300 seconds), it will send only one register message every one hour. • Under dial peer configuration mode, you can use the **voice-class sip registration passthrough rate-limit expires** *value* **reg-sync** *dial-peer-tag* command.
Step 6	**end** **Example:** `Device(conf-serv-sip)# end`	Returns to privileged EXEC mode.

Configuring the Class of Restrictions (COR) List

Perform the following task to configure the COR list to allow the local call to go through the registrar.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **corlist incoming** *dial-peer*
5. **corlist outgoing** *dial-peer*
6. **description** *string*
7. **destination-pattern** *number*
8. **session protocol sipv2**
9. **session target registrar**
10. **voice-class sip registration passthrough local-fallback** *tag*
11. **end**

DETAILED STEPS

	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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 3 voip`	Enters dial peer configuration mode.
Step 4	**corlist incoming** *dial-peer* **Example:** `Device(config-dial-peer)# corlist incoming FromPhone`	Specifes the COR to be applied on an incoming dial peer (for incoming calls).
Step 5	**corlist outgoing** *dial-peer* **Example:** `Device(config-dial-peer)# corlist outgoing FromSP`	Specifes the COR to be applied for outgoing dial peer (for outgoing calls).

	Command or Action	Purpose
Step 6	**description** *string* **Example:** `Device(config-dial-peer)# description registration`	Adds a description to a dial peer.
Step 7	**destination-pattern** *number* **Example:** `Device(config-dial-peer)# destination-pattern 1111`	Specifies either the prefix or the full E.164 telephone number to be used for the dial peer.
Step 8	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Specifies the session protocol for SIP calls between local and remote devices using the packet network.
Step 9	**session target registrar** **Example:** `Device(config-dial-peer)# session target registrar`	Specifies to route the call to the registrar end point for SIP dial peers.
Step 10	**voice-class sip registration passthrough local-fallback** *tag* **Example:** `Device(config-dial-peer)# voice-class sip registration passthrough local-fallback 5`	Configures SIP registration passthrough for local fallback mode.
Step 11	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Verifying Survivability Enhancements

Perform this task to verify the configurations for the survivability enhancements. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice summary**
3. **show sip-ua registration passthrough status**
4. **show sip-ua register status**
5. **show voip rtp connections**
6. **show call active voice compact**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
```
Device> enable
```

Step 2 **show dial-peer voice summary**
Displays the summary information for each voice dial peer.

Example:
```
Device# show dial-peer voice summary

dial-peer hunt 0
             AD                                      PRE PASS                 OUT
TAG    TYPE  MIN  OPER PREFIX    DEST-PATTERN      FER THRU SESS-TARGET    STAT PORT    KEEPALIVE
1      voip  up   up              1111...            0  syst registrar
2      voip  up   down            1......           0  syst ipv4:10.104.45.253           busyout
1000   voip  down down            9900...           0  syst ipv4:9.0.0.174:30601
101    voip  down down            1......           0  syst ipv4:10.104.45.31
102    voip  down down            11.....           0  syst ipv4:10.104.45.253
300    voip  down down            .T                0  syst
400    voip  down down            11110...          0  syst registrar
```

Step 3 **show sip-ua registration passthrough status**
Displays information about the SIP user agent registration passthrough status. In the sample output shown below, the parameter In-Exp shows the remaining expiry time and the survival field parameters can be regsync, locfall, or normal.

Example:
```
Device# show sip-ua registration passthrough status

CallId       Line         peer         mode In-Exp        reg-I Out-Exp survival
============ ============ ============ ==== ============= ===== ======= ========
5300         1111008      1            e2e  1041  /1200   ----- 1200    normal *
5305         1111002      1            e2e  2847  /3000   ----- 3000    normal *
5311         1111020      1            e2e  1070  /1200   ----- 1200    normal *
================================================================================
```

Step 4 **show sip-ua register status**
Displays information about the SIP user agent register status.

Example:

```
Device# show sip-ua register status

Line          peer  expires(sec) reg survival P-Associ-URI
========      ===== ============ === ========
11123         23     59          yes regsync
```

Step 5 **show voip rtp connections**

Displays Real-Time Transport Protocol (RTP) named event packets.

Example:

```
Device# show voip rtp connections

VoIP RTP Port Usage Information:
Max Ports Available: 8091, Ports Reserved: 101, Ports in Use: 2
Port range not configured, Min: 16384, Max: 32767

                                                                                                  Ports
Ports       Ports
Media-Address Range                                                                             Available
Reserved    In-use

Default Address-Range                                                                             8091
101         2

VoIP RTP active connections :
No. CallId     dstCallId  LocalRTP RmtRTP LocalIP                                          RemoteIP
1     5324       5325       16410    16464  9.40.1.168                                       9.40.1.173
2     5325       5324       16412    16528  9.40.1.168                                       9.40.1.174
Found 2 active RTP connections
```

Step 6 **show call active voice compact**

Displays the compact version of the call information for voice calls in progress.

Example:

```
Device# show call active voice compact

<callID>  A/O FAX T<sec> Codec       type        Peer Address       IP R<ip>:<udp>
Total call-legs: 2
      5324 ANS     T9     g711ulaw    VOIP        P1111008       9.40.1.173:16464
      5325 ORG     T9     g711ulaw    VOIP        P1111020       9.40.1.174:16528
```

Configuration Examples for Survivability Enhancements

Example: Configuring Local Fallback Globally

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
```

```
Device(conf-serv-sip)# registration passthrough local-fallback 10
Device(config-serv-sip)# end
```

Example: Configuring Local Fallback on a Dial Peer

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 2 voip
Device(config-dial-peer)# voice-class sip registration passthrough local-fallback 10
Device(config-dial-peer)# end
```

Example: Configuring OPTIONS Ping

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 3 voip
Device(config-dial-peer)# voice-class sip options-keepalive up-interval 120 down-interval
120
Device(config-dial-peer)# end
```

Example: Configuring the Registration Timer

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# registrar server expires max 300 min 200
Device(conf-serv-sip)# end
```

Example: Configuring REGISTER Message Throttling

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# registration passthrough rate-limit expires 3600 local-fallback 3
Device(conf-serv-sip)# end
```

Example: Configuring the COR List

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 2 voip
Device(config-dial-peer)# corlist incoming FromPhone
Device(config-dial-peer)# corlist outgoing FromSP
Device(config-dial-peer)# description registration
Device(config-dial-peer)# destination-pattern 1111
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# session target registrar
Device(config-dial-peer)# voice-class sip registration passthrough local-fallback 5
Device(config-dial-peer)# end
```

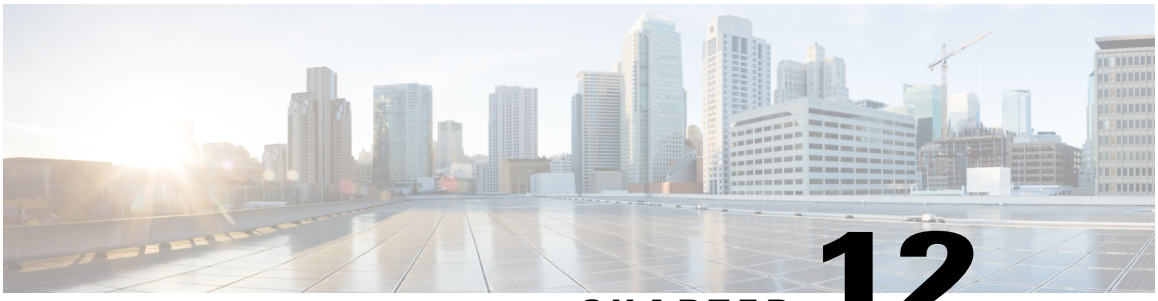
Feature Information for Survivability Enhancements

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for Survivability Enhancements

Feature Name	Releases	Feature Information
Survivability Enhancements	15.3(3)M	When a WAN link goes down temporarily or the registrar server is down, local calls cannot be made and no calls can be routed to and from the phones. The Survivability Enhancements feature on the NanoCUBE is used to: • Monitor the WAN status periodically from the Nano CUBE. • Route calls and handle line-seize subscriptions locally when the WAN link is down. • Synchronize the registrations with the server when the WAN link is up.
Survivability Enhancements—Support for Extensions and Phone Numbers	Cisco IOS 15.6(2)T	From Cisco IOS 15.6(2)T onwards, when the WAN link or registration server is down, local phones can be reached on their short extensions or phone numbers by similar phones subscribed to the server through the same NANOCUBE.



CHAPTER 12

Voice Quality Monitoring

The Voice Quality Monitoring (VQM) feature uses Flexible NetFlow to export voice quality metrics related to media (voice) quality, such as conversational mean opinion score (MOS), packet loss rate, and so on. VQM enables you to monitor the quality of calls traversing your VoIP network, and you can diagnose the cause of voice quality issues and troubleshoot them.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Voice Quality Monitoring

The **aqm-register-fnf** command must be configured before you use the **media monitoring** command to configure voice quality metrics.

Information About Voice Quality Monitoring

The VQM (Voice Quality Monitor) uses Flexible NetFlow to export voice quality metrics measured by the **media monitoring** command. To help the NetFlow collector to process the flow record, VQM also reports call-related information such as calling number, called number, call setup time, and so on. The Voice Quality Metrics enables statistics gathering on packet arrival (late/lost/early). From these statistics, a voice quality measurement is developed to show the quality of the call. The output is in a simple format, using a system of good, poor, and bad types of ratings.

The following are the five metrics added to Call Detail Record (CDR) and Management Information Base (MIB) in NanoCUBE, indicating voice quality:

1. MOSQe (conversational quality MOS)
2. Round-trip-delay.
3. Receive-delay (current jitter buffer size).
4. Packet-Loss-Rate.
5. Out-of-Order-Rate.

The CDR is sent at the end of a call if AAA accounting is configured.

A CDR example is as follows:

<MOS-Con>4.4072</MOS-Con>

<round-trip-delay>1 ms</round-trip-delay>

<receive-delay>64 ms</receive-delay>

<voice-quality-total-packet-loss>0.0000 %</ voice-quality-total-packet-loss>

<voice-quality-out-of-order>0.0000 %</ voice-quality-out-of-order>

VQM Metrics

The following are the metrics exported by VQM:

NanoCube IOS VQM, Voice/Audio Quality Metric	Description
GwReceivedCalledNumber	The directory number portion of To URI from the Session Initiation Protocol (SIP) signaling or the extension receiving the call.
GwReceivedCallingNumber	The directory number portion of From URI from the SIP signaling or the extension originating the call.
SetupTime	The time at which monitoring began on this RTP stream.
CallDuration	The time (in milliseconds) from when monitoring began on this RTP stream until the reception of the last RTP packet on this stream.

DspRXBadPkt	The total number of packets determined by the simulated jitter buffer to be having bad protocols and that need to be dropped.
DspRXOutSeq	The total number of packets that arrive at the jitter buffer out of sequence.
DspConf CodecID	The last voice coder-decoder (CODEC) detected in this RTP stream. Note that an endpoint may change the voice CODEC mid-stream.
DspPlayDelay Cur	The current jitter buffer delay in milliseconds. In the case of an RTP stream in the call history, the last jitter buffer delay (does not apply to a fixed jitter buffer configuration).
DspPlayDelayMin	The minimum jitter buffer delay in milliseconds (does not apply to a fixed jitter buffer configuration)
DspPlayDelayMax	The maximum jitter buffer delay in milliseconds (does not apply to a fixed jitter buffer configuration).
rfc3550JitterMeanMilliseconds	The packet-to-packet delay variation (jitter) in milliseconds, as defined in RFC 3550.
ProtocolCallId	The SIP call ID read-only by the SIP proxy. This value may be unknown if using the B2BUA or if call signaling is not being monitored.
GlobalCallId	Internally generated ID identifying this call.
DspDely RT	The instantaneous round-trip delay. This may be obtained from RTCP XR or SR reports; or if no reports are available, from an average of ICMP echo or timestamp requests sent to both endpoints. If no report information is available and round-trip delay cannot be determined from ICMP (example, a firewall in the path did not allow the traffic), this statistic will be reported as unavailable.
DspDelyED	The instantaneous one-way delay, including any delay that can be introduced by the jitter buffer and codec processing.
DspRFactrR1	The listening quality R factor. Listening quality indicates the perceived quality of the transmission for a user not actively involved in the conversation, but passively listening. Listening quality does not consider delay or recency. Some users may prefer R-factor measurements to MOS scores, because MOS scales may differ based on the CODEC type and region of deployment, whereas R factor measurements are consistent across CODECs and regions.

DspRFactrR2	The conversational quality R factor. Conversational quality indicates the impact of the quality of the transmission on the dynamics of conversational exchanges between two parties; such metrics take into account delay, echo, and recency. For example, for a link with a large delay, participants in a conversation might frequently find themselves interrupting each other and talking over each other, since one party will be unable to perceive when the other party has started talking. Some users may prefer R factor measurements to MOS, since MOS scales may differ based on the CODEC type and region of deployment, whereas R factor measurements are consistent across CODECs and regions.
DspRFactrMosConv	The conversational quality MOS. Conversational quality indicates the impact of the quality of the transmission on the dynamics of conversational exchanges between two parties; such metrics take into account delay, echo, and recency. For example, for a link with a large delay, participants in a conversation might frequently find themselves interrupting each other and talking over each other, since one party will be unable to perceive when the other party has started talking.
DspRFactrMosLisn	The listening quality MOS. Listening quality indicates the perceived quality of the transmission for a user not actively involved in the conversation, but passively listening. Listening quality does not consider delay or recency.
DspCealRatioAV	Average of Concealment Ratio reports since the start of a call.
DspConfJtrTyp	The configured jitter buffer type for this RTP stream, either adaptive or fixed. An adaptive jitter buffer dynamically varies the delay from packet reception to packet playback; a fixed jitter buffer uses the same delay for each packet. This is a jitter buffer; no packets are actually being discarded.
DspConfJtrMin	The minimum delay that will be applied to packets received when using an adaptive jitter buffer.
DspConfJtrInit	The value that represents the initial delay that will be applied to received packets when using an adaptive jitter buffer. When using a fixed jitter buffer, this represents the delay that will be applied to each packet when it is received.
DspConfJtrMax	The value that represents an upper bound on the delay that will be applied to received packets when using an adaptive jitter buffer. When using a fixed jitter buffer, this metric represents the maximum number of packets that can be inserted into the buffer. (Subsequently, inserted packets will be discarded.)
DspRXEarlPkt	The total number of packets for this RTP stream arriving early (prior to the anticipated packet arrival). Each packet is classified as either late or early with the exception of the first packet that is treated as a reference packet.

DspRXLatPkt	The total number of packets for this RTP stream arriving late (after the anticipated packet arrival). Each packet is classified as either late or early with the exception of the first packet that is treated as a reference packet.
DspPktBfrOvr	The total number of packets discarded by the jitter buffer due to jitter buffer overrun.
DspPktCealCount	The total number of packets discarded by the jitter buffer due to jitter buffer underrun.

How to Configure Voice Quality Monitoring

Configuring Voice Quality Metrics

Before You Begin

The **aqm-register-fnf** command must be configured at the global configuration mode to export the audio and video call quality metrics to flow record using Flexible NetFlow collector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **media monitoring** *max-calls*
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **media monitoring** *max-calls*

DETAILED STEPS

	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.

	Command or Action	Purpose
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters voice service configuration mode and specified Voice over IP as the voice-encapsulation type.
Step 4	**media monitoring** *max-calls* **Example:** `Device(conf-voi-serv)# media monitoring 300`	Enables media monitoring and specifies the maximum number of calls to be monitored. **Note** You can monitor only up to 302 channels for NANOCUBE, that is, about 151 calls.
Step 5	**exit** **Example:** `Device(conf-voi-serv)# exit`	Exits voice service configuration mode and returns to global configuration mode.
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Device(conf-voi-serv)# dial-peer voice 5 voip`	Enters dial-peer configuration mode, defines a particular dial peer, and specifies the method of voice encapsulation as VoIP.
Step 7	**media monitoring** *max-calls* **Example:** `Device(config-dial-peer)# media monitoring 300`	Enables media monitoring for calls landing on the dial peer specified in Step 6.

Enabling Media Statistics Globally

Perform this task to globally enable media statistics in voice-service configuration mode to estimate the values for packet loss, jitter, and round-trip time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **media statistics**
5. **end**

DETAILED STEPS

	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 voice service VoIP configuration mode.
Step 4	**media statistics** **Example:** `Device(conf-voi-serv)# media statistics`	Enables media statistics to estimate the values of packet loss, jitter, and Round Trip Time (RTT) statistics. • The statistics are displayed using the **show voice history** and **show call active voice** commands. • If the **media statistics** command is disabled, the values will be zero.
Step 5	**end** **Example:** `Device(conf-voi-serv)# end`	Returns to privileged EXEC mode.

Verifying Voice Quality Monitoring

Perform this task to verify the configuration for voice quality monitoring. The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show voip rtp connections**
3. **show sccp connections**
4. **show voice monitoring-channels**
5. **show call active voice**
6. **show call active voice stats**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:
```
Device> enable
```

Step 2 **show voip rtp connections**
Displays Real-Time Transport Protocol (RTP) named event packets.

Example:
```
Device# show voip rtp connections

No. CallId   dstCallId    LocalRTP   RmtRTP    LocalIP    RemoteIP   MPSS
1      37      38           16582     18236    10.1.1.2   10.1.1.7   NO
2      38      37           16524     19542    10.1.1.2   10.1.1.1   NO
3      39      40           17644     2000     10.1.1.2   10.1.1.2   NO
4      41      40           16622     2000     10.1.1.2   10.1.1.2   NO
```

Step 3 **show sccp connections**
Displays information about the connections controlled by the Skinny Client Control Protocol (SCCP) transcoding and conferencing applications.

Example:
```
Device# show sccp connections

sess_id      conn_id    stype    mode       codec     ripaddr     rport    sport
3            4          xcode    sendrecv   g711u     100.1.1.2   2000     16622
3            3          xcode    sendrecv   g711u     100.1.1.2   2000     17644
Total number of active session(s) 1, and connection(s) 2
```

Step 4 **show voice monitoring-channels**
Displays voice monitoring statistics.

Example:
```
Device# show voice monitoring-channels

max vq mon channels = 10 vq mon channels in use = 2 vq mon channels left =8
```

Step 5 **show call active voice**
Displays statistics on the CUBE if the Voice Quality Metrics feature is configured.

Example:

```
Device# show call active voice

RxPakNumber=5496
RxSignalPak=0
RxComfortNoisePak=0
RxDuration=109900
RxVoiceDuration=109920
RxOutOfSeq=0
RxLatePak=0
RxEarlyPak=0
RxBadProtocol=0
LevelRxPowerMean=0
ErrRxDrop=0
ErrRxControl=0
```

Step 6 **show call active voice stats**

Displays Concealment Statistics and R-Factor Statistics (G.107 MOS) on the Cisco UBE if the Voice Quality Metrics feature is configured. A sample output is provided below for a voice call using G.711ulaw, VAD on, and at 5 percent packet loss rate.

Example:

```
Device# show call active voice statslsec MC

DSP/CS: CR=0.0527, AV=0.0502, MX=0.0527, CT=1220, TT=24270, OK=50, CS=44, SC=0, TS=50, DC=0
SP/RF: ML=3.9855, MC=0.0000, R1=79, R2=0, IF=15, ID=0, IE=0, BL=25, R0=94, VR=1.1
```

In the sample output, the following can be noted:

- The average conceal ratio (AV) is about 5 percent.
- The ratio of total conceal time and total speech time is about 5 percent (1220/24270).
- BL for codec G.711 is 25 (based on G.113).
- IE for codec G.711 is 0 (G.113).
- IE for codec G.711 is 0.
- R0 is 94 (G.107).

The following table defines the abbreviations used in the sample output.

Table 9: Router Output Definitions for the show call active voice stats command

Type	Abbreviation	Definition

DSP/CS: Concealment Statistics	CR	concealRatioCurrent
	AV	ConcealRatioAverage
	MX	ConcealRatioMaximum
	CT	ConcealDuration
	TT	SpeechDuration
	OK	OkSeconds
	CS	ConcealSeconds
	SC	SevereConcealSeconds
	TS	SevereConcealThreshold
DSP/RF: R-Factor Statistics (G.107 MOS)	ML	MOSLQE
	R1	RFactorProfile1
	IF	IeEff
	BL	CodecBaselineBPL
	R0	R0Default
	VR	R-Factor version

Troubleshooting Tips

Use the following debug commands to troubleshoot the Voice Quality Monitoring feature.

- **debug sccp messages**
- **debug voip rtp packets**
- **debug performance monitor**
- **debug radius accounting**
- **debug aaa accounting**

Configuration Examples for Voice Quality Monitoring

Example: Configuring Voice Quality Metrics

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# media monitoring 300
Device(conf-serv-sip)# exit
Device(config)# dial-peer voice 5 voip
Device(config-dial-peer)# media monitoring 300
```

Example: Configuring Media Statistics Globally

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# media statistics
Device(conf-voi-serv)# end
```

Feature Information for Voice Quality Monitoring

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for Voice Quality Monitoring

Feature Name	Releases	Feature Information
Voice Quality Monitoring	15.3(3)M	The Voice Quality Monitoring (VQM) feature uses Flexible NetFlow to export voice quality metrics related to media (voice) quality, such as conversational mean opinion score (MOS), packet loss rate, and so on. VQM enables you to monitor the quality of calls traversing your VoIP network, and you can diagnose the cause of voice quality issues and troubleshoot them.