# Configuration of SIP Trunking for PSTN Access (SIP-to-SIP) Configuration Guide, Cisco IOS Release 15M&T

**Last Modified:** November 25, 2015

**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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.

# CONTENTS

CHAPTER 1

# 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

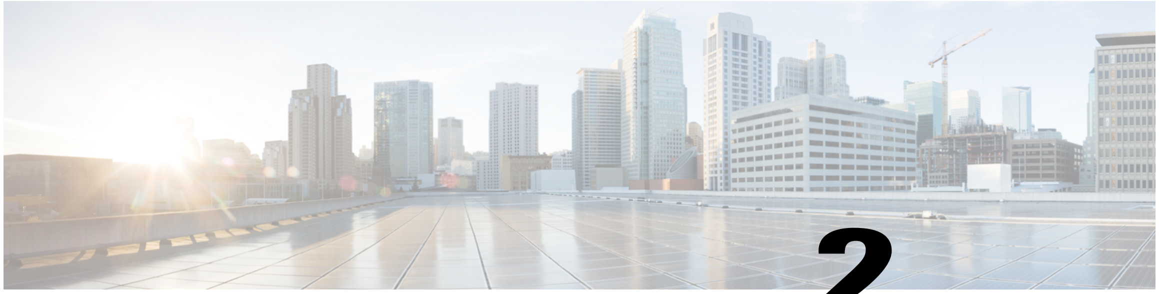
This chapter contains the following configuration topics:

### 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 2

# 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.

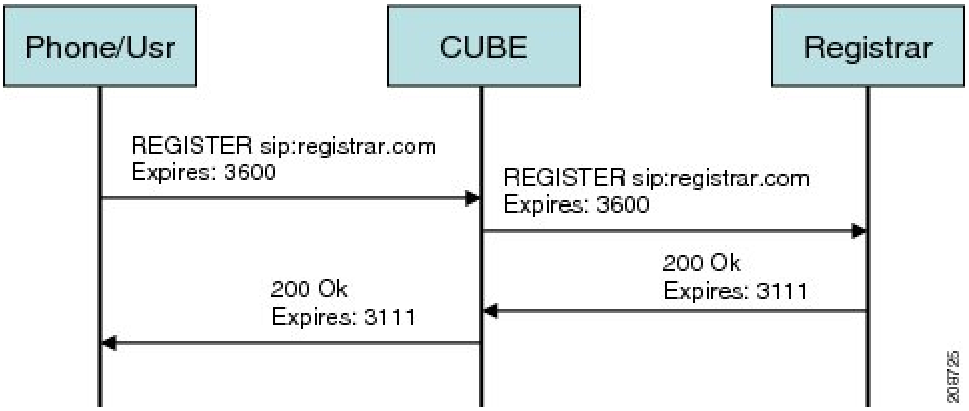
### End-to-End Mode--Call Flows

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 1: End-to-End Registration Pass-through Mode--Register Success Scenario*



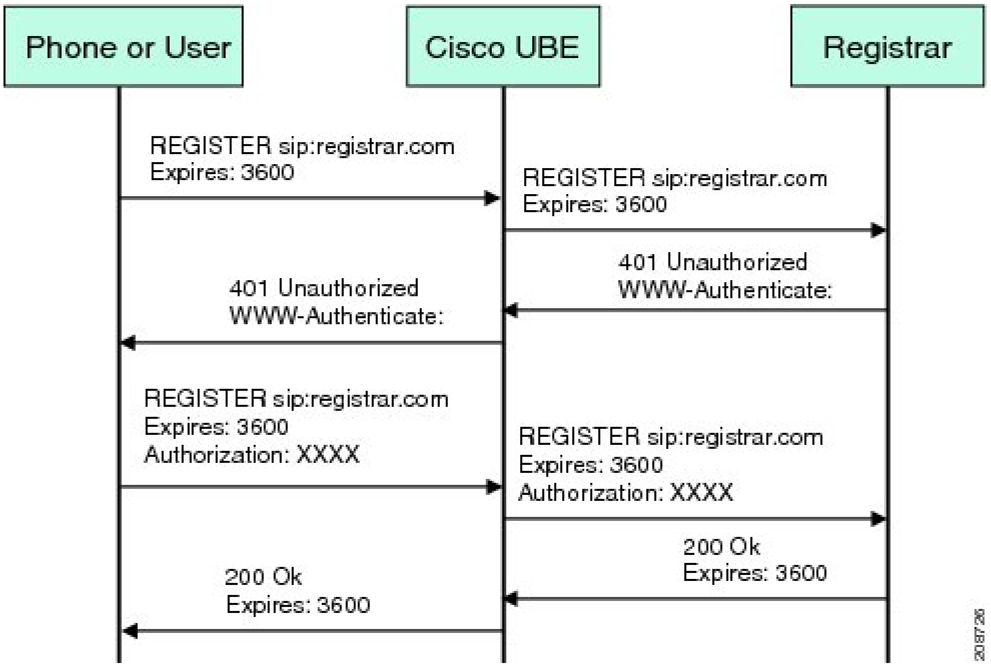
The register success scenario for the end-to end registration pass-through mode is as follows:

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 2: 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.

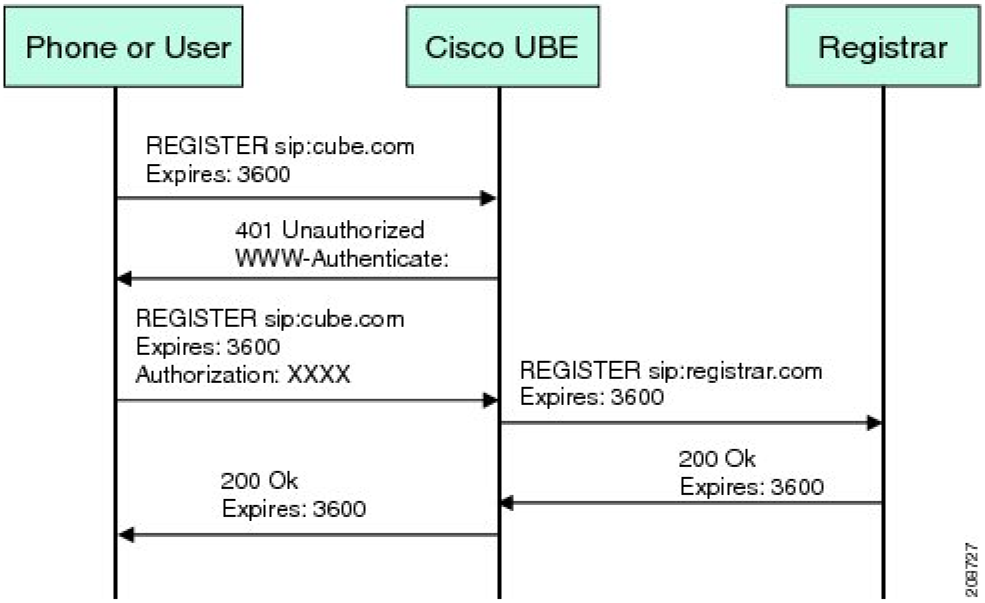
Peer-to-Peer Mode--Call Flows

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 3: 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:

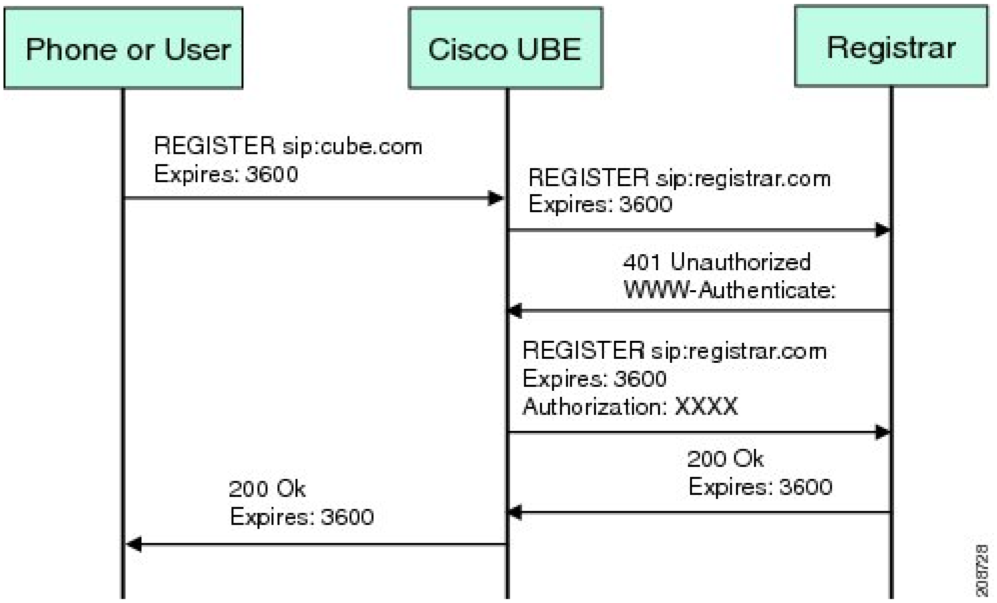
1. The user sends the register request to Cisco UBE.
2. Cisco UBE matches the register request with a dial peer and forwards the register request to the registrar.
3. Cisco UBE receives a success message (200 OK message) from the registrar and forwards it to the endpoint (user). The following functions 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 4: 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 8.

# 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.

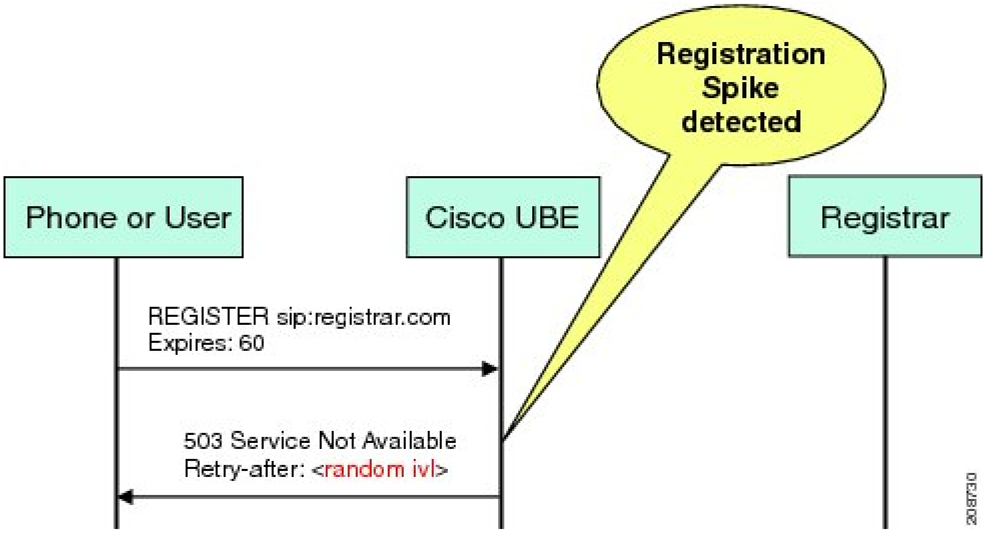
The registration overload protection functionality protects the network from the following:

- 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 5: Register Overload Protection*


Registration Spike detected
Phone or User
Cisco UBE
Registrar
REGISTER sip:registrar.com
Expires: 60
503 Service Not Available
Retry-after: <random ivl>
208730

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

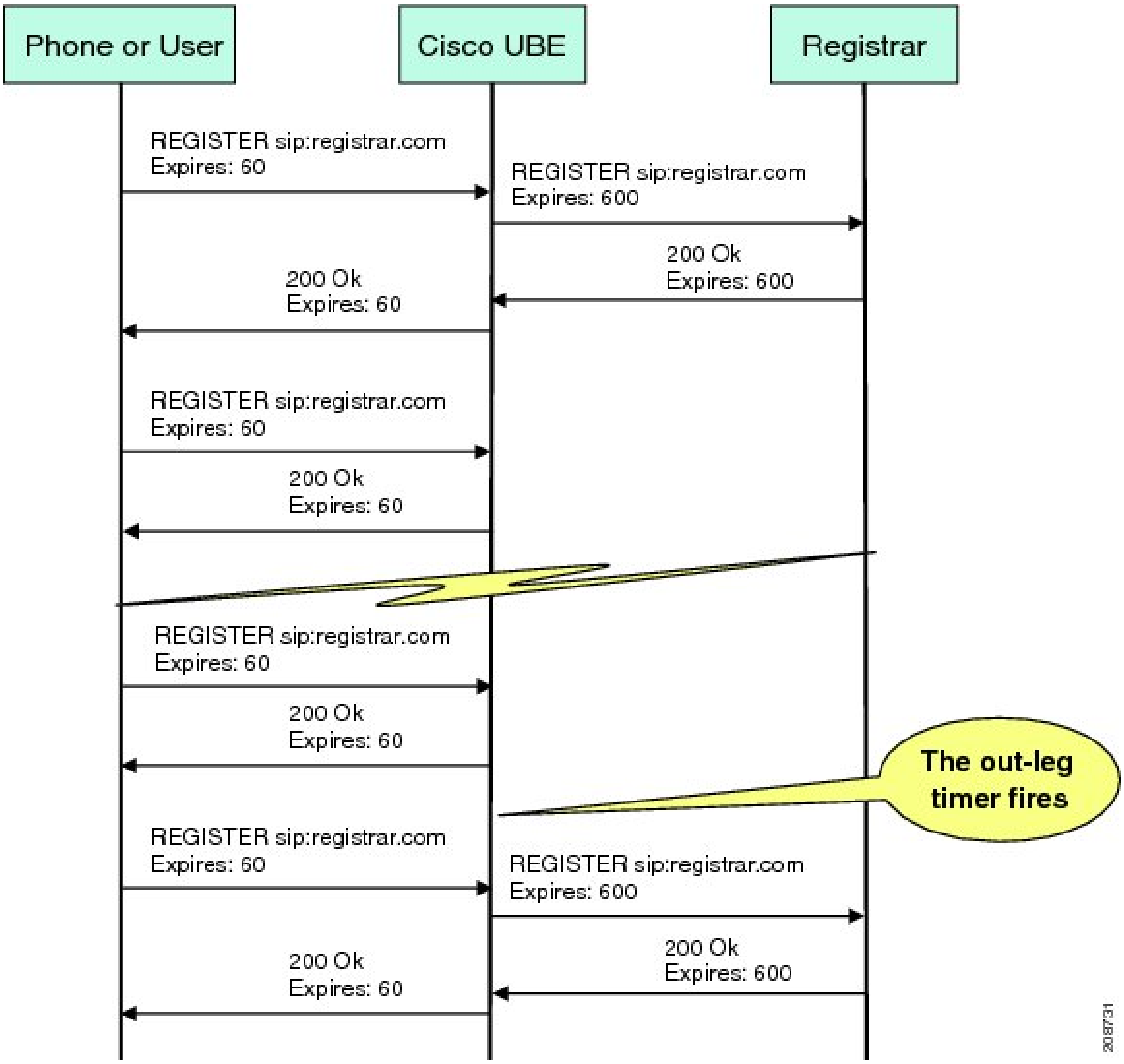
The registration rate-limiting functionality enables you to configure different SIP registration pass-through rate-limiting options. The rate-limiting options include setting the expiry time and the fail count value for a Cisco UBE. You can configure the expiry time to reduce the load on the registrar 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 6: 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 12.
- 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **voice service voip**<br>**Example:**<br>`Device(config)# voice service voip` | Enters voice-service configuration mode. |
| **Step 4** | **sip**<br>**Example:**<br>`Device(conf-voi-serv)# sip` | Enters service SIP configuration mode. |
| **Step 5** | **registrar server** [**expires** [**max** *value*] [**min** *value*]]<br>**Example:**<br>`Device(conf-serv-sip)# registrar server` | Enables the local SIP registrar.<br>• Optionally you can configure the expiry time of the registrar using the following keywords:<br>• **expires**--Configures the registration expiry time.<br>• **max**--Configures the maximum registration expiry time.<br>• **min**--Configures the minimum registration expiry time.<br>**Note** The **registrar** command must be configured in peer-to-peer mode. Otherwise, the register request is rejected with the 503 response message. |
| **Step 6** | **end**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **voice service voip**<br>**Example:**<br>`Device(config)# voice service voip` | Enters voice-service configuration mode. |
| **Step 4** | **sip**<br>**Example:**<br>`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*]]<br>**Example:**<br>`Device(conf-serv-sip)# registration passthrough` | Configures the SIP registration pass-through options.<br>• You can specify different SIP registration pass-through options using the following keywords:<br>• **rate-limit**--Enables rate-limiting.<br>• **expires**--Configures expiry value for rate-limiting.<br>• **fail-count**--Configures fail count during rate-limiting. |

| | Command or Action | Purpose |
|---|---|---|
| | | - **registrar-index**--Configures a list of registrars to be used for registration. |
| **Step 6** | **end**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>- Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* {**pots** \| **voatm** \| **vofr** \| **voip**}<br>**Example:**<br>`Device(config)# dial-peer voice 444 voip` | Enters dial peer voice configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **voice-class sip registration passthrough static** [**rate-limit** [**expires** *value*] [**fail-count** *value*] [**registrar-index** [*index*]] \| **registrar-index** [*index*]]<br>**Example:**<br>`Device(config-dial-peer)# voice-class sip registration passthrough static` | Configure SIP registration pass-through options on a dial peer on a dial peer.<br>• You can specify different SIP registration pass-through options using the following keywords:<br>• **rate-limit**--Enables rate-limiting.<br>• **expires**--Configures expiry value for rate-limiting.<br>• **fail-count**--Configures fail count during rate-limiting.<br>• **registrar-index**--Configures a list of registrars to be used for registration. |
| **Step 5** | **exit**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **sip-ua**<br>**Example:**<br>`Device(config)# sip-ua` | Enters SIP user-agent configuration mode. |
| **Step 4** | **registration spike** *max-number*<br>**Example:**<br>`Device(config-sip-ua)# registration spike 100` | Configures registration overload protection functionality on Cisco UBE. |
| **Step 5** | **end**<br>**Example:**<br>`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** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Device> enable` | • Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* {**pots** \| **voatm** \| **vofr** \| **voip**}<br>**Example:**<br>`Device(config)# dial-peer voice 444 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **session target registrar**<br>**Example:**<br>`Device(config-dial-peer)# session target registrar` | Configures Cisco UBE to route the call to the registrar endpoint. |
| **Step 5** | **exit**<br>**Example:**<br>`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
```

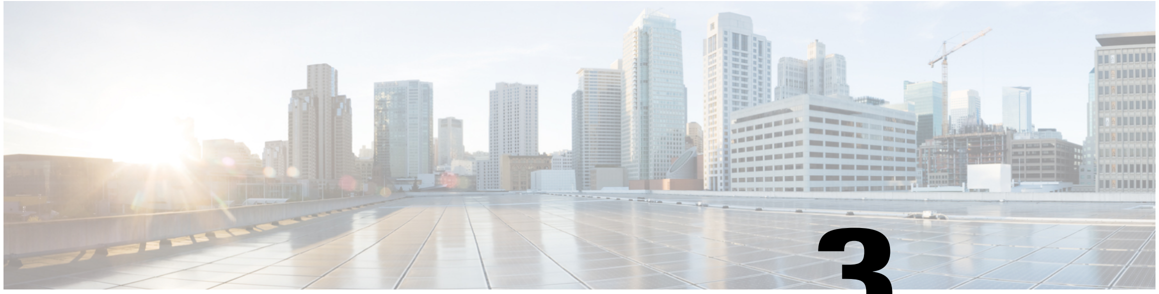
```
!
dial-peer voice 1111 voip
 destination-pattern 1234
 voice-class sip pass-thru content unsupp
 session protocol sipv2
 session target registrar
!
dial-peer voice 1111 voip
 destination-pattern 1234
 voice-class sip pass-thru content unsupp
 voice-class sip registration passthrough static rate-limit 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.<br>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**. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| 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.<br>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 3

# 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 |
| 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode. Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **dial-peer voice** *tag* **voip**<br>**Example:**<br>`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*}<br>**Example:**<br>`Device(config-dial-peer)# voice-class sip options-keepalive up-interval 12 down-interval 65 retry 3` | Monitors connectivity between endpoints.<br>• **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.<br>• **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.<br>• **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**<br>**Example:**<br>`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 . 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<br>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.<br>In Cisco IOS Release 15.0(1)M, this feature was implemented on the Cisco Unified Border Element.<br>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.<br>In Cisco IOS XE Release 3.1S, this feature was implemented on the Cisco Unified Border Element (Enterprise).<br>The following command was introduced: **voice-class sip options-keepalive** |

CHAPTER 4

# Bandwidth-Based Call Admission Control

The Bandwidth-Based Call Admission Control (CAC) feature provides the functionality to reject SIP calls when the bandwidth accounted by the SIP signaling layer exceeds the aggregate bandwidth threshold for VoIP media traffic—voice, video, and fax. This functionality helps you prevent Quality of Service (QoS) degradation of VoIP media traffic for existing calls when the bandwidth allocated for VoIP traffic is fully utilized. The Bandwidth-Based Call Admission Control feature is supported on Session Initiation Protocol (SIP) trunks of the Time Division Multiplexing (TDM) SIP gateway and the Cisco Unified Border Element (Cisco UBE).

Midcall media renegotiation can also be rejected if the configured maximum bandwidth threshold for the VoIP media traffic is exceeded. The call continues as per the previously negotiated media codecs if midcall media renegotiation is rejected.

The excess subscription of the bandwidth allocated for VoIP traffic results in VoIP media packets being dropped or delayed, irrespective of the VoIP call to which they belong. Under such circumstances, it is better to deny new calls to prevent QoS deterioration for existing VoIP call traffic. The existing traffic congestion resolution mechanisms do not differentiate between media packets of existing calls (admitted) and new calls (oversubscribed). Similarly, existing call signaling is unaware of the media traffic congestion. The Bandwidth-Based Call Admission Control feature fills this gap by rejecting new SIP calls when the bandwidth allocated for VoIP traffic is fully utilized. The actual bandwidth usage is not measured and policed. The lower-level QoS policies control the traffic characteristics for the specified traffic class.

**Note** The Bandwidth-Based Call Admission Control feature is applicable only to VoIP traffic.

# 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 Bandwidth-Based Call Admission Control

- Cisco UBE, configured with the Bandwidth-Based Call Admission Control feature, will not reject the call if the bandwidth of the SDP answer is greater than the bandwidth of the SDP offer.
- Layer 2 overhead is not included in the bandwidth calculation.
- A midcall delayed-offer (DO) to DO call is disconnected if the bandwidth requested in an offer message (200 OK) exceeds the threshold bandwidth.
- Real Time Transport Control Protocol (RTCP) and RTP Named Telephone Event (RTP-NTE) bandwidth requirement is not computed.
- The Bandwidth-Based Call Admission Control feature does not support:
    - Cisco fax relay.
    - Filtering of codecs to accommodate calls within the available bandwidth.
    - Media flow-around, Session Description Protocol (SDP) pass-through, out-of-box low-density transcoding, high-density transcoding, video transcoding, and midcall consumption functionalities.
    - Non-SIP call legs.
    - SIP-to-H32X call flows (SIP-H320, H320-SIP, SIP-H324, H324-SIP).
    - Subinterfaces for bandwidth-based CAC on an interface.

# Information About Bandwidth-Based Call Admission Control

## Maximum Bandwidth Calculation

The bandwidth requirement for each SIP call leg is calculated using the codec information available in the SDP. Here, the actual media bandwidth used is not measured.

Bandwidth in Kbps (Kilo bits per second) = [codec bytes + RTP header (12) + UDP (8) + IP Header (20 or 40)] * Packets per seconds * 8/1000

Where, codec bytes = Codec payload size, in bytes, for a given packetization interval.

RTP header = Size of the RTP header, in bytes.

UDP = Size of the UDP header, in bytes.

IP Header = Size of the IP header, in bytes. The IPV4 header is 20 bytes and the IPV6 header is 40 bytes.

Packets per second = Number of RTP packets sent or received per second. This value is as per the negotiated packetization interval. The SDP media attribute "ptime" indicates the number of packets per second.

# Bandwidth Tables

This section provides the sample maximum bandwidth calculation for audio and fax calls.

***Table 4: Audio Bandwidth Table***

| Codec and Bit Rate (Kbps) | Codec Sample Size in Bytes | Voice Payload Size in Bytes | Voice Payload Size in Milliseconds | Packets Per Second | Bandwidth for IPv4 (excluding Layer 2) in Kbps | Bandwidth for IPv6 (excluding Layer 2) in Kbps |
|---|---|---|---|---|---|---|
| G.711 (64 Kbps) | 80 | 160 | 20 | 50 | 80 | 88 |
| G.729 (8 Kbps) | 10 | 20 | 20 | 50 | 24 | 32 |
| G.723.1 (6.3 Kbps) | 24 | 24 | 30 | 33.3 | 17 | 22 |
| G.723.1 (5.3 Kbps) | 20 | 20 | 30 | 33.3 | 16 | 21 |
| G.726 (32 Kbps) | 20 | 80 | 20 | 50 | 48 | 56 |
| G.726 (24 Kbps) | 15 | 60 | 20 | 50 | 40 | 48 |
| G.726 (16 Kbps) | 10 | 40 | 20 | 50 | 32 | 40 |
| G.728 (16 Kbps) | 10 | 40 | 20 | 50 | 32 | 40 |
| G722_64k (64 Kbps) | 80 | 160 | 20 | 50 | 80 | 88 |
| ilbc_mode_20 (15.2 Kbps) | 38 | 38 | 20 | 50 | 31 | 39 |

| | | | | | | |
|---|---|---|---|---|---|---|
| ilbc_mode_30 (13.33 Kbps) | 50 | 50 | 30 | 33.3 | 24 | 29 |
| gsm (13 Kbps) | 33 | 33 | 20 | 50 | 30 | 37 |
| gsm (12 Kbps) | 32 | 32 | 20 | 50 | 29 | 37 |
| G.Clear (64 Kbps) | 80 | 160 | 20 | 50 | 80 | 88 |
| GSM AMR | — | — | — | — | 15 | 15 |
| ISAC (32 Kbps) | — | — | — | — | 37 | 37 |
| Aacld (mpeg4) | — | — | — | — | Derived from the SDP bandwidth attribute (TIAS) | Derived from the SDP bandwidth attribute (TIAS) |

*Table 5: Fax Bandwidth Table*

| T.38 Fax Bit Rate | Redundancy | Maximum Bandwidth in Kbps |
|---|---|---|
| 2400 | None | 8 |
| 2400 | Redundancy | 17 |
| 9600 (default) | None | 16 |
| 9600 (default) | Redundancy | 46 |
| 14400 | None | 20 |
| 14400 | Redundancy | 65 |
| 33600 | None | 40 |
| 33600 | Redundancy | 142 |

# How to Configure Bandwidth-Based Call Admission Control

## Configuring Bandwidth-Based Call Admission Control at the Interface Level

You can configure the Bandwidth-Based Call Admission Control feature at the interface level to reject SIP calls when the bandwidth required for the call exceeds the aggregate bandwidth threshold.

You can configure the Bandwidth-Based Call Admission Control feature for the following interfaces:

- ATM
- Ethernet (Fast Ethernet, Gigabit Ethernet)
- Loopback
- Serial

**Note** Cisco recommends that you configure a bind media to associate a specific interface for SIP calls. Otherwise, the interface used for the calls will be determined based on the best local address that can access the remote media source address (for early offer calls) or the remote signaling source address (for delayed offer calls). When you use a Loopback interface to configure CAC, you must configure an additional bind-to-bind media with the Loopback interface at the global level or the dial peer level. Configure the **bind media source-interface loopback** *number* command in service SIP configuration mode to configure a bind media.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **call threshold interface** *type number* **int-bandwidth** {**class-map** *name* [**l2-overhead** *percentage*] | **low** *low-threshold* **high** *high-threshold*} [**midcall-exceed**]
4. **end**

### DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **call threshold interface** *type number* **int-bandwidth** {**class-map** *name* [**l2-overhead** *percentage*] \| **low** *low-threshold* **high** *high-threshold*} [**midcall-exceed**]<br>**Example:**<br>`Device(config)# call threshold interface GigabitEthernet 0/0 int-bandwidth low 1000 high 20000 midcall-exceed`<br>or<br>`Device(config)# call threshold interface GigabitEthernet 0/0 int-bandwidth class-map voip-traffic l2-overhead 20 midcall-exceed` | Configures the Bandwidth-Based Call Admission Control feature at the interface level to reject SIP calls when the bandwidth required for the calls exceed the aggregate bandwidth threshold.<br>• You can configure the **call threshold interface** *type number* **low** *low-threshold* **high** *high-threshold* [**midcall-exceed**] command to apply call admission control to reject SIP calls once the accounted bandwidth reaches the *high-threshold* value and continues to be above the *low-threshold* value.<br>• You can configure the **call threshold interface** *type number* **int-bandwidth class-map** *name* [**l2-overhead** *percentage*] [**midcall-exceed**] command to use the bandwidth value provisioned in the QoS policy under the interface for VoIP media traffic for CAC. See the Modular Quality of Service Command-Line Interface Overview document at http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfmdcli.html for information on the usage of the QoS policy with Call Admission Control.<br>• SIP calls are rejected when the calculated aggregate bandwidth of VoIP media traffic on the specified interface exceeds the configured bandwidth threshold. |
| **Step 4** | **end**<br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and enters privileged EXEC mode. |

# Configuring Bandwidth-Based Call Admission Control at the Dial Peer Level

You can configure the Bandwidth-Based Call Admission Control feature at the dial peer level to reject SIP calls when the bandwidth required for the calls exceeds the aggregate bandwidth threshold.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **session protocol sipv2**
5. **max-bandwidth** *bandwidth-value* [**midcall-exceed**]
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* **voip**<br>**Example:**<br>`Device(config)# dial-peer voice 44 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **session protocol sipv2**<br>**Example:**<br>`Device(config-dial-peer)# session protocol sipv2` | Configures the Bandwidth-Based Call Admission Control feature for SIP dial peers only. |
| **Step 5** | **max-bandwidth** *bandwidth-value* [**midcall-exceed**]<br>**Example:**<br>`Device(config-dial-peer)# max-bandwidth 24 midcall-exceed` | Configures the Bandwidth-Based Call Admission Control feature at the dial peer level to reject SIP calls when the bandwidth required for the calls exceed the aggregate bandwidth threshold.<br>• Configuring the **midcall-exceed** keyword allows exceeding the bandwidth threshold during mid-call media renegotiation. Media renegotiation exceeding the bandwidth threshold is rejected by default. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **end**<br>**Example:**<br>`Device(config-dial-peer)# end` | Exits dial peer configuration mode and enters privileged EXEC mode. |

# Configuring the Bandwidth-Based Call Admission Control SIP Error Response Code Mapping

Mapping of the call rejection cause code to a specific SIP error response code is known as error response code mapping. The cause code for the call rejected because of the bandwidth-based CAC can be mapped to a SIP error response code between 400 to 600. The default SIP error response code is 488.

You can configure SIP error response codes for calls rejected by the Bandwidth-Based Call Admission Control feature at the global level, dial peer level, or both.

## Configuring Bandwidth-Based Call Admission Control SIP Error Response Code Mapping at the Global Level

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **error-code-override cac-bandwidth failure** *sip-status-code-number*
6. **end**

### DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **voice service voip**<br>**Example:**<br>`Device(config)# voice service voip` | Enters voice-service configuration mode. |
| **Step 4** | **sip**<br>**Example:**<br>`Device(conf-voi-serv)# sip` | Enters service SIP configuration mode. |
| **Step 5** | **error-code-override cac-bandwidth failure** *sip-status-code-number*<br>**Example:**<br>`Device(conf-serv-sip)# error-code-override cac-bandwidth failure 500` | Configures bandwidth-based CAC SIP error response code mapping at the global level. |
| **Step 6** | **end**<br>**Example:**<br>`Device(conf-serv-sip)# end` | Exits service SIP configuration mode and enters privileged EXEC mode. |

## Configuring Bandwidth-Based Call Admission Control SIP Error Response Code Mapping 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 error-code-override cac-bandwidth failure** {*sip-status-code-number* | **system**}
5. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* {**pots** \| **voatm** \| **vofr** \| **voip**}<br>**Example:**<br>`Device(config)# dial-peer voice 88 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **voice-class sip error-code-override cac-bandwidth failure** {*sip-status-code-number* \| **system**}<br>**Example:**<br>`Device(config-dial-peer)# voice-class sip error-code-override cac-bandwidth failure 500` | Configures bandwidth-based CAC SIP error response code mapping at the dial peer level. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-dial-peer)# end` | Exits dial peer configuration mode and enters privileged EXEC mode. |

# Verifying Bandwidth-Based Call Admission Control

Perform this task to verify the configuration for the Bandwidth-Based Call Admission Control feature on Cisco UBE. The **show** commands need not be entered in any specific order.

### SUMMARY STEPS

1. **enable**
2. **show call threshold config**
3. **show call threshold status**
4. **show call threshold stats**
5. **show dial-peer voice**

## DETAILED STEPS

**Step 1** **enable**

**Example:**

```
Device>enable
```

Enables privileged EXEC mode.

**Step 2** **show call threshold config**

**Example:**

```
Device# show call threshold config

Some resource polling interval:
  CPU_AVG interval: 60
  Memory interval:  5

IF                  Type            Value   Low     High    Enable
-----               ----            -----   ----    ----    ------
GigabitEthernet0/0  int-bandwidth   0       100     400      N/A
```

Displays the current call threshold configuration at the interface level for all resources.

**Step 3** **show call threshold status**

**Example:**

```
Device# show call threshold status

Status  IF                  Type            Value   Low     High    Enable
------  ---                 ------          ----    ----    ----    -----
Avail   GigabitEthernet0/0  int-bandwidth   0       100     400      N/A
```

Displays the availability status of resources that are configured when the Bandwidth-Based Call Admission Control feature is enabled at an interface level.

**Step 4** **show call threshold stats**

**Example:**

```
Device# show call threshold stats

Total resource check: 2
successful: 1
 failed:   1

1: ------------------------
  Failed resources: int-bandwidth,
  related interface: GigabitEthernet0/0; related option:N/A
  Recorded time: 04:49:39 UTC Wed Dec 8 2010
2: ------------------------
Successful
  All resources are available for this check.
  Recorded time: 04:29:39 UTC Wed Dec 8 2010
```

Displays the statistics of resources that are configured when the Bandwidth-Based Call Admission Control feature is enabled at an interface level.

**Step 5** **show dial-peer voice**

**Example:**

```
Device# show dial-peer voice

incoming called-number = `2000', connections/maximum = 0/unlimited,
bandwidth/maximum = 0/400,
…….
Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
Accepted Calls = 3, Refused Calls = 0,
Bandwidth CAC Accepted Calls = 3, Bandwidth CAC Refused Calls = 0
```

Displays information for the voice dial peer.

## Troubleshooting Tips

The following commands can help troubleshoot the Bandwidth-Based Call Admission Control feature:

- **debug ccsip all**
- **debug voice ccapi all**

# Configuration Examples for Bandwidth-Based Call Admission Control

## Example: Configuring Bandwidth-Based Call Admission Control at the Interface Level

The following example shows how to configure Cisco UBE to reject new SIP calls if the accounted VoIP media bandwidth on Gigabit Ethernet interface 0/0 exceeds 400 Kbps of bandwidth and continues to have a bandwidth above 100 Kbps:

```
Device> enable
Device# configure terminal
Device(config)# call threshold interface GigabitEthernet 0/0 int-bandwidth  low 100 high
400
```

The following example shows how to configure Cisco UBE to reject new SIP calls if the VoIP media bandwidth on Gigabit Ethernet interface 0/0 exceeds the configured bandwidth for priority traffic in the “voip_traffic” class:

```
Device>enable
Device# configure terminal
Device(config)# class-map match-all voip-traffic

Device(config-cmap)# policy-map voip-policy
Device(config-pmap)# class voip-traffic
Device(config-pmap-c)# priority 440
Device(config-pmap-c)# end
```

```
Device# enaconfigure terminalble
Device(config)# call threshold interface GigabitEthernet 0/0 int-bandwidth class-map
voip-traffic l2-overhead 10
```

**Note** Layer 2 overhead of 10 percent in the **call threshold** command indicates that the IP bandwidth, excluding Layer 2, is 90 percent of the configured priority bandwidth.

## Example: Configuring Bandwidth-Based Call Admission Control at the Dial Peer Level

The following example shows how to configure Cisco UBE to reject calls once the accounted aggregate bandwidth of active calls exceeds 400 Kbps for a SIP dial peer:

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 2000 voip
Device(config)# session protocol sipv2
Device(config-dial-peer)# max-bandwidth 400
```

## Example: Configuring the Bandwidth-Based Call Admission Control SIP Error Response Code Mapping at the Global Level

The following example shows how to configure Cisco UBE for bandwidth-based CAC SIP error response code mapping at the global level:

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# error-code-override cac-bandwidth 500
```

## Example: Configuring the Bandwidth-Based Call Admission Control SIP Error Response Code Mapping at the Dial Peer Level

The following example shows how to configure Cisco UBE for bandwidth-based CAC SIP error response code mapping at the dial peer level:

```
Device> enable
Device# configure terminal
Device(config)# dial-peer voice 88 voip
Device(config-dial-peer)# voice-class sip error-code-override cac-bandwidth failure 500
```

# Feature Information for Bandwidth-Based Call Admission Control

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 . An account on Cisco.com is not required.

***Table 6: Feature Information for Bandwidth-Based Call Admission Control***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Bandwidth-Based Call Admission Control | 15.2(2)T | The Bandwidth-Based Call Admission Control feature provides the functionality to reject SIP calls when the bandwidth accounted by the SIP signaling layer exceeds the aggregate bandwidth threshold for VoIP media traffic—voice, video, and fax. This functionality helps prevent QoS degradation of VoIP media traffic for existing calls when the bandwidth allocated for VoIP traffic is fully utilized.<br>The following commands were introduced or modified:<br>**call threshold interface**, **error-code-override**, **max-bandwidth**, **show call threshold**, **voice-class sip** |

| Feature Name | Releases | Feature Information |
|---|---|---|
| Bandwidth-Based Call Admission Control | Cisco IOS XE Release 3.7S | The Bandwidth-Based Call Admission Control feature provides the functionality to reject SIP calls when the bandwidth accounted by the SIP signaling layer exceeds the aggregate bandwidth threshold for VoIP media traffic—voice, video, and fax. This functionality helps prevent QoS degradation of VoIP media traffic for existing calls when the bandwidth allocated for VoIP traffic is fully utilized.<br>The following commands were introduced or modified:<br>**call threshold interface**, **error-code-override**, **max-bandwidth**, **show call threshold**, **voice-class sip** |

CHAPTER 5

# Configuring DSCP Policing and Media Bandwidth Policing

This module explains the following features:

- AS SIP—DSCP Policing
- AS SIP—Media Bandwidth Policing

The Assured Services over Session Initiation Protocol Differentiated Services Code Point (AS SIP—DSCP Policing) Policing and the AS SIP—Media Bandwidth Policing feature adds the media policy functionality to the Cisco Unified Border Element (Cisco UBE) on a per-call basis to control the bandwidth. Real Time Protocol (RTP) packets are dropped, and MIB and system logs are generated if there is any DSCP policy, marking, and media bandwidth profiling violation.

## 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 Configuring DSCP Policing and Media Bandwidth Policing

Following are the restrictions for the AS SIP—DSCP Policing feature:

- The Session Description Protocol (SDP) pass-through feature along with Resource Priority Header (RPH) to DSCP marking and policing are not supported.
- High availability (HA) is not supported.

Following are the restrictions for the AS SIP—Media Bandwidth Policing feature:

- The SDP pass-through feature along with the Media Bandwidth Policing feature are not supported.
- Flow-around cases are not applicable to the Media Bandwidth Policing feature.
- HA is not supported.

# Information About Configuring DSCP Policing and Media Bandwidth Policing

## AS SIP—DSCP Policing

The AS SIP—DSCP Policing feature provides the following functionalities:

- A mechanism to map the RPH to the DSCP values in the IP header for audio and video calls.
- A policy to match DSCP values of incoming media calls to the preconfigured value and take an action depending upon the configuration.

You must map RPH to DSCP values to provide priority and precedence for VoIP calls at all layers. DSCP policing and marking are supported as part of the AS SIP—DSCP Policing feature for RTP media. The DSCP policing functionality checks DSCP values for media packets (RTP) and informs incorrect marking of DSCP values. The DSCP marking functionality marks packets with the correct DSCP value as per the SIP RPH.

The AS SIP—DSCP Policing feature supports two new namespaces, UC and CUC. The namespace support is enabled by default and no configuration is required. Asymmetric call leg configuration is also supported: that is, you can have the RPH pass-through configuration on one call leg and RPH to DSCP policing on the another.

## AS SIP—Media Bandwidth Policing

In releases prior to Cisco IOS Release 15.2(2)T, Cisco UBE does not support media on a policing per-call basis. Hence, few endpoints negotiate the G729 codec using the SIP offer answer model and send RTP packets with the payload of G711. Few endpoints negotiate with G729 10 ms (one packet per 10 ms) but send two packets as a response to the request of 10 ms. In both cases, more bandwidth than the negotiated bandwidth is used. Cisco UBE has no mechanism to detect bandwidth violation and enforce policing on media policing approaches.

To overcome this problem, the AS SIP—Media Bandwidth Policing feature was introduced in Cisco IOS Release 15.2(2)T. This feature introduces traffic policing on the Cisco UBE to limit media bandwidth usage to the negotiated rate. Excess traffic is dropped when the traffic rate reaches the configured maximum value. The AS SIP—Media Bandwidth Policing feature is supported only on RTP packets.

The AS SIP—Media Bandwidth Policing feature identifies violations in the bandwidth and triggers the following policing actions on additional RTP packets received:

- Drops all violated packets.
- Drops all violated packets and disconnects the call once it reaches the configured number of violations.
- Ignores the violations.

You can enable system log and Simple Network Management Protocol (SNMP) trap generations to inform system administrators about policing violations.

## Resource Priority Header

RPH is a SIP header. A SIP request with a RPH is treated as follows:

- The request is given an elevated priority to access public switched telephone network (PSTN) gateway resources, such as trunk circuits.
- The request can interrupt lower priority requests at a user terminal, such as an IP phone.
- The request can carry information from one multilevel priority domain in a telephone network to another, without SIP proxies inspecting or modifying the header field.
- In SIP proxies and back-to-back user agents, requests of higher priorities can displace the existing signaling requests or bypass the PSTN gateway capacity limits in effect for lower priorities.

This RPH header provides priority and precedence at Layer 7. It is not treated the same way in lower layers.

## Differentiated Services Code Point

DSCP or differentiated services code point (DiffServ) is a computer networking architecture that specifies a simple, scalable, and coarse-grained mechanism for classifying and managing network traffic, and providing quality of service (QoS) on modern IP networks.

# How to Configure DSCP Policing and Media Bandwidth Policing Features

## Configuring AS SIP—DSCP Policing Feature at the Global Level

Perform this task to configure the AS SIP—DSCP Policing feature at the global level, that is on all dial peers.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class dscp-profile** *tag*
4. **dscp media** {**audio** | **video**} {**flah-override-override** | **flash-override** | **flsh** | **immediate** | **priority** | **routine**} {*dscp-value* | *set-af* | *set-cf* | **ef** | **zero**}
5. **violation** *number* **action** {**disconnect** | **ignore**} [**no-syslog**]
6. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **voice class dscp-profile** *tag*<br>**Example:**<br>`Router(config)# voice class dscp-profile 1` | Configures a DSCP profile and enters voice class configuration mode. |
| **Step 4** | **dscp media** {**audio** \| **video**} {**flah-override-override** \| **flash-override** \| **flsh** \| **immediate** \| **priority** \| **routine**} {*dscp-value* \| *set-af* \| *set-cf* \| **ef** \| **zero**}<br>**Example:**<br>`Router(config-class)# dscp media audio routine ef` | Specifies the RPH to DSCP mapping. |
| **Step 5** | **violation** *number* **action** {**disconnect** \| **ignore**} [**no-syslog**]<br>**Example:**<br>`Router(config-class)# violation 20000 action ignore` | Specifies the action that needs to be performed on any violation in the DSCP policy. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config-class)# end` | Exits voice class configuration mode and enters privileged EXEC mode. |

# Applying the DSCP Policing Profile at the Global Level

Perform this task to apply the DSCP policing profile at the global level, that is to apply the profile to all dial peers.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **dscp-profile** *tag*
6. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **voice service voip**<br>**Example:**<br>`Router(config)# voice service voip` | Enters voice service configuration mode. |
| **Step 4** | **sip**<br>**Example:**<br>`Router(conf-voi-serv)# sip` | Enters service SIP configuration mode. |
| **Step 5** | **dscp-profile** *tag*<br>**Example:**<br>`Router(conf-serv-sip)# dscp-profile 1` | Applies a DSCP policing profile at the global level.<br>• If a DSCP policy is applied globally and to a dial peer, the dial peer configuration takes precedence over the global configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **end**<br>**Example:**<br>`Router(conf-serv-sip)# end` | Exits service SIP configuration mode and enters privileged EXEC mode. |

# Applying the DSCP Policing Profile at the Dial Peer Level

Perform this task to apply the DSCP policing profile at the dial peer level.

When the DSCP policing profile is applied to a dial peer and the mode is configured as RPH pass-through, the policy will be enforced if there is any match for the “r-priority” value in the RPH. If there is no match in the namespace, the domain name system (DNS) will be used to match the “r-priority”.

If the RPH pass-through mode is configured, the RPH is passed as it is. The RPH is truncated if the following values are above the specified limits:

- Only the namespace is changed and there is no change in the subdomain and priority.
- Maximum namespace allowed is up to ten characters.
- Maximum subdomains supported range is from 000000 to FFFFFF.
- Maximum priority values allowed are 24 characters.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**pots** | **voatm** | **vofr** | **voip**}
4. **voice-class sip resource priority dscp-profile** *tag*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* {**pots** \| **voatm** \| **vofr** \| **voip**}<br>**Example:**<br>`Router(config)# dial-peer voice 4 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **voice-class sip resource priority dscp-profile** *tag*<br>**Example:**<br>`Router(config-dial-peer)# voice-class sip resource priority dscp-profile 1` | Applies a DSCP profile parameter.<br>• If a DSCP policy is applied globally and to a dial peer, the dial peer configuration takes precedence over the global configuration. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config-dial-peer)# end` | Exits dial peer configuration mode. |

# Enabling the SNMP Trap for the DSCP Policing Feature at the Global Level

Perform this task to enable the SNMP trap for the DSCP Policing feature at the global level, that is on all dial peers.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps voice** [**dscp-profile**] [**fallback**] [**high-ds0-util**] [**low-ds0-util**] [**media-policy**]
4. **exit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **snmp-server enable traps voice** [**dscp-profile**] [**fallback**] [**high-ds0-util**] [**low-ds0-util**] [**media-policy**]<br>**Example:**<br>`Router(config)# snmp-server enable traps voice dscp-profile` | Enables SNMP DSCP profile voice notifications. |
| **Step 4** | **exit**<br>**Example:**<br>`Router(config)# exit` | Exits global configuration mode and enters privileged EXEC mode. |

# Enabling the SNMP Trap for the DSCP Policing Feature at the Dial Peer Level

Perform this task to enable the SNMP trap for the DSCP policing feature for a specific dial peer.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **snmp enable peer-trap dscp-profile**
5. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* **voip**<br>**Example:**<br>`Router(config)# dial-peer voice 1 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **snmp enable peer-trap dscp-profile**<br>**Example:**<br>`Router(config-dial-peer)# snmp enable peer-trap dscp-profile` | Enables DSCP profile violation traps. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config-dial-peer)# end` | Exits dial peer configuration mode and enters privileged EXEC mode. |

# Verifying the AS SIP-DSCP Policing Feature

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

### SUMMARY STEPS

1. **enable**
2. **show ip interface brief**
3. **show call active voice brief**

### DETAILED STEPS

**Step 1** **enable**

**Example:**

```
Router> enable
```

Enables privileged EXEC mode.

**Step 2** **show ip interface brief**

**Example:**

```
Router# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         10.0.35.11      YES manual up                    up
GigabitEthernet0/1         10.1.3.3        YES NVRAM  administratively down down
```

Displays a brief summary of an interface's IP information and status.

**Step 3** **show call active voice brief**

**Example:**

```
Router# show call active voice brief

<ID>: <CallID> <start>.<index> +<connect> pid:<peer_id> <dir> <addr> <state>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> dscp:<packets violation> media:<packets
violation>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
  delay:<last>/<min>/<max>ms <codec> <textrelay> <transcoded

 media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>

 long duration call detected:<y/n> long duration call duration :<sec> timestamp:<time>
  MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
   last <buf event time>s dur:<Min>/<Max>s
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 Tele <int> (callID) [channel_id] tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l> i/o:<l>/<l> dBm
  MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops>
         speeds(bps): local <rx>/<tx> remote <rx>/<tx>
 Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
 bw: <req>/<act> codec: <audio>/<video>
  tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
 rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>



Telephony call-legs: 0
SIP call-legs: 2
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 2
0    : 21 23:08:52.157 IST Tue Jul 12 2011.1 +8900 pid:3 Answer 1000 active
 dur 00:00:56 tx:2766/442560 rx:2811/449760 dscp:2814 media:0
 IP 9.44.46.21:20332 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw TextRelay: off
Transcoded: No
 media inactive detected:n media contrl rcvd:n/a timestamp:n/a
 long duration call detected:n long duration call duration:n/a timestamp:n/a

0    : 22 23:08:52.707 IST Tue Jul 12 2011.1 +7780 pid:4 Originate 2000 active
 dur 00:00:57 tx:2811/449760 rx:2766/442560 dscp:2767 media:0
 IP 9.44.46.25:31290 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw TextRelay: off
Transcoded: No
 media inactive detected:n media contrl rcvd:n/a timestamp:n/a
 long duration call detected:n long duration call duration:n/a timestamp:n/a


Telephony call-legs: 0
SIP call-legs: 2
H323 call-legs: 0
Call agent controlled call-legs: 0
```

```
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 2
```

Displays call information for voice calls in progress.

# Configuring the AS SIP—Media Bandwidth Policing Profile at the Global Level

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media profile police** *tag*
4. **violation** *number* **action** {**disconnect** | **drop** | **ignore**} [**no-syslog**]
5. **overhead** {**audio** | **video**} *percentage*
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| Step 2 | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| Step 3 | **media profile police** *tag*<br>**Example:**<br>`Router(config)# media profile police 1` | Configures the media bandwidth policing profile at the global level and enters media profile configuration mode. |
| Step 4 | **violation** *number* **action** {**disconnect** \| **drop** \| **ignore**} [**no-syslog**]<br>**Example:**<br>`Router(cfg-mediaprofile)# violation 20000 action drop no-syslog` | Specifies the number of violations after which the action needs to be taken.<br>• Use the **no-syslog** keyword to configure the Cisco UBE to disable the system log. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **overhead** {**audio** \| **video**} *percentage*<br>**Example:**<br>`Router(cfg-mediaprofile)# overhead audio 10` | Configures the overhead bandwidth percentage above the negotiated bandwidth. |
| **Step 6** | **end**<br>**Example:**<br>`Router(cfg-mediaprofile)# end` | Exits media profile configuration mode and enters privileged EXEC mode. |

# Applying the Media Bandwidth Policing Profile at the Global Level

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **media police-profile** *tag*
5. **end**

## DETAILED STEPS

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **media police-profile** *tag*<br>**Example:**<br>`Router(conf-voi-serv)# media police-profile 1` | Applies the media bandwidth policing profile at the global level. |
| **Step 5** | **end**<br>**Example:**<br>`Router(conf-voi-serv)# end` | Exits voice service configuration mode and enters privileged EXEC mode. |

# Applying the Media Bandwidth Policing Profile at the Dial Peer Level

Applying the media bandwidth policing profile at the dial peer level involves two actions: applying the profile for a media class and then applying the corresponding media class to a dial peer.

Perform this task to apply the media bandwidth policing profile at the dial peer level.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media class** *tag*
4. **police profile** *tag*
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **media-class** *tag*
8. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 2 | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| Step 3 | **media class** *tag*<br>**Example:**<br>`Router(config)# media class 1` | Configures a media class and enters media class configuration mode. |
| Step 4 | **police profile** *tag*<br>**Example:**<br>`Router(cfg-mediaclass)# police profile 1` | Applies the media bandwidth policing profile to the media class. |
| Step 5 | **exit**<br>**Example:**<br>`Router(cfg-mediaclass)# exit` | Exits media class configuration mode and enters global configuration mode. |
| Step 6 | **dial-peer voice** *tag* **voip**<br>**Example:**<br>`Router(config)# dial-peer voice 1 voip` | Enters dial peer voice configuration mode. |
| Step 7 | **media-class** *tag*<br>**Example:**<br>`Router(config-dial-peer)# media-class 1` | Applies the media class at the dial peer level. |
| Step 8 | **end**<br>**Example:**<br>`Router(config-dial-peer)# end` | Exits dial peer voice configuration mode and enters privileged EXEC mode. |

# Enabling SNMP Traps for the Media Bandwidth Policing Feature at the Global Level

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps voice media-policy**
4. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **snmp-server enable traps voice media-policy**<br>**Example:**<br>`Router(config)# snmp-server enable traps voice media-policy` | Enables SNMP media policy voice traps at the global level. |
| **Step 4** | **exit**<br>**Example:**<br>`Router(config)# exit` | Exits global configuration mode and enters privileged EXEC mode. |

# Enabling SNMP Traps for the Media Bandwidth Policing Feature at the Dial Peer Level

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **snmp enable peer-trap media-policy**
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **dial-peer voice** *tag* **voip**<br>**Example:**<br>`Router(config)# dial-peer voice 4 voip` | Enters dial peer voice configuration mode. |
| **Step 4** | **snmp enable peer-trap media-policy**<br>**Example:**<br>`Router(config-dial-peer)# snmp enable peer-trap media-policy` | Enables SNMP media policy voice traps at the dial peer level. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config-dial-peer)# end` | Exits dial peer configuration mode and enters privileged EXEC mode. |

# Verifying the AS SIP-Media Bandwidth Policing Profile Feature

Perform this task to verify the configuration for AS SIP-Media Bandwidth Policing Profile feature on Cisco UBE. The **show** commands need not be entered in any specific order.

## SUMMARY STEPS

1. **enable**
2. **show call history voice brief**
3. **show call history voice stats**
4. **show call history voice stats**
5. **show call history video brief**
6. **show call history video stats**
7. **show call active voice brief**
8. **show call active voice stats**
9. **show call active video brief**
10. **show call history video stats**
11. **show dial-peer voice**

## DETAILED STEPS

### Step 1 enable

**Example:**

```
Router> enable
```

Enables privileged EXEC mode.

### Step 2 show call history voice brief

**Example:**

```
Router# show call history voice brief

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
  delay:<last>/<min>/<max>ms <codec>  <textrelay> <transcoded>

 media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>

 long duration call detected:<y/n> long duration call duration :<sec> timestamp:<time>
  MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
   last <buf event time>s dur:<Min>/<Max>s
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 Telephony <int> (callID) [channel_id] tx:<tot>/<voice>/<fax>ms <codec> noise:<lvl>dBm acom:<lvl>dBm

  MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops> disc:<cause
code>
            speeds(bps): local <rx>/<tx> remote <rx>/<tx>
 Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
 bw: <req>/<act> codec: <audio>/<video>
  tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
 rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>


Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays a truncated version of the call history table for voice calls.

**Step 3** **show call history voice stats**

**Example:**

```
Router# show call history voice stats

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
DSP/TX: PK=, SG=, NS=, DU=, VO=
DSP/RX: PK=, SG=, CF=, RX=, VO=, BS=, BP=, LP=, EP=
DSP/PD: CU=, MI=, MA=, CO=, IJ=
DSP/PE: PC=, IC=, SC=, RM=, BO=, EE=
DSP/LE: TP=, TX=, RP=, RM=, BN=, ER=, AC=
DSP/ER: RD=, TD=, RC=, TC=
DSP/IC: IC=

DSP/EC: CI=, FM=, FP=, VS=, GT=, GR=, JD=, JN=, JM=, JX=
DSP/KF: KF=, AV=, MI=, BS=, NB=, FL=, NW=, VR=
DSP/CS: CR=, AV=, MX=, CT=, TT=, OK=, CS=, SC=, TS=, DC=
DSP/RF: ML=, MC=, R1=, R2=, IF=, ID=, IE=, BL=, R0=, VR=
DSP/UC: U1=, U2=, T1=, T2=
DSP/DL: RT=, ED=

MIC Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=
EAR Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=

Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays the call history table for voice calls.

**Step 4** **show call history voice stats**

**Example:**

```
Router#  show call history voice stats

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
DSP/TX: PK=, SG=, NS=, DU=, VO=
DSP/RX: PK=, SG=, CF=, RX=, VO=, BS=, BP=, LP=, EP=
DSP/PD: CU=, MI=, MA=, CO=, IJ=
DSP/PE: PC=, IC=, SC=, RM=, BO=, EE=
DSP/LE: TP=, TX=, RP=, RM=, BN=, ER=, AC=
DSP/ER: RD=, TD=, RC=, TC=
DSP/IC: IC=

DSP/EC: CI=, FM=, FP=, VS=, GT=, GR=, JD=, JN=, JM=, JX=
DSP/KF: KF=, AV=, MI=, BS=, NB=, FL=, NW=, VR=
DSP/CS: CR=, AV=, MX=, CT=, TT=, OK=, CS=, SC=, TS=, DC=
DSP/RF: ML=, MC=, R1=, R2=, IF=, ID=, IE=, BL=, R0=, VR=
DSP/UC: U1=, U2=, T1=, T2=
DSP/DL: RT=, ED=

MIC Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=
EAR Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=

Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays information about digital signal processing (DSP) voice quality metrics.

**Step 5** **show call history video brief**

**Example:**

```
Router# show call history video brief

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
  delay:<last>/<min>/<max>ms <codec>  <textrelay> <transcoded>

 media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>

 long duration call detected:<y/n> long duration call duration :<sec> timestamp:<time>
  MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
   last <buf event time>s dur:<Min>/<Max>s
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 Telephony <int> (callID) [channel_id] tx:<tot>/<voice>/<fax>ms <codec> noise:<lvl>dBm acom:<lvl>dBm

   video: h320:<call type> tx:<video codec> <video pkts>/<video bytes> rx:<video codec> <video
pkts>/<video bytes>
  MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops> disc:<cause
code>
             speeds(bps): local <rx>/<tx> remote <rx>/<tx>
 Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
 bw: <req>/<act> codec: <audio>/<video>
  tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
 rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>

Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays a truncated version of video call history information.

**Step 6** **show call history video stats**

**Example:**

```
Router# show call history video stats

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays call history information for signaling connection control protocol (SCCP) video calls.

**Step 7** **show call active voice brief**

**Example:**

```
Router# show call active voice brief

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> pid:<peer_id> <dir> <addr> <state>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> dscp:<packets violation> media:<packets
violation> audio tos:<audio tos value> video tos:<video tos value>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
  delay:<last>/<min>/<max>ms <codec> <textrelay> <transcoded

 media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>

 long duration call detected:<y/n> long duration call duration :<sec> timestamp:<time>
  MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
   last <buf event time>s dur:<Min>/<Max>s
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 Tele <int> (callID) [channel_id] tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l> i/o:<l>/<l> dBm
  MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops>
         speeds(bps): local <rx>/<tx> remote <rx>/<tx>
 Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
 bw: <req>/<act> codec: <audio>/<video>
  tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
 rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
```

**Step 8** **show call active voice stats**

**Example:**

```
Router# show call active voice stats

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> dscp:<packets violation> audio tos:<audio
tos value> video tos:<video tos value>

DSP/TX: PK=, SG=, NS=, DU=, VO=
DSP/RX: PK=, SG=, CF=, RX=, VO=, BS=, BP=, LP=, EP=
DSP/PD: CU=, MI=, MA=, CO=, IJ=
DSP/PE: PC=, IC=, SC=, RM=, BO=, EE=
DSP/LE: TP=, TX=, RP=, RM=, BN=, ER=, AC=
DSP/ER: RD=, TD=, RC=, TC=
DSP/IC: IC=

DSP/EC: CI=, FM=, FP=, VS=, GT=, GR=, JD=, JN=, JM=, JX=
DSP/KF: KF=, AV=, MI=, BS=, NB=, FL=, NW=, VR=
DSP/CS: CR=, AV=, MX=, CT=, TT=, OK=, CS=, SC=, TS=, DC=
DSP/RF: ML=, MC=, R1=, R2=, IF=, ID=, IE=, BL=, R0=, VR=
DSP/UC: U1=, U2=, T1=, T2=
DSP/DL: RT=, ED=

MIC Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=
EAR Direction:

DSP/NR: NR=, ND=, LV=, IN=
DSP/AS: AE=, AD=, AM=, AV=, NT=, DT=, TT=, TD=, LF=, LD=

Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 0
```

Displays information about DSP voice quality metrics.

**Step 9** **show call active video brief**

### Example:

```
Router# show call active video brief

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> pid:<peer_id> <dir> <addr> <state>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> dscp:<packets violation> media:<packets
violation> audio tos:<audio tos value> video tos:<video tos value>
 IP <ip>:<udp> rtt:<time>ms pl:<play>/<gap>ms lost:<lost>/<early>/<late>
  delay:<last>/<min>/<max>ms <codec> <textrelay> <transcoded

 media inactive detected:<y/n> media cntrl rcvd:<y/n> timestamp:<time>

 long duration call detected:<y/n> long duration call duration :<sec> timestamp:<time>
  MODEMPASS <method> buf:<fills>/<drains> loss <overall%> <multipkt>/<corrected>
   last <buf event time>s dur:<Min>/<Max>s
 FR <protocol> [int dlci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 ATM <protocol> [int vpi/vci cid] vad:<y/n> dtmf:<y/n> seq:<y/n>
  <codec> (payload size)
 Tele <int> (callID) [channel_id] tx:<tot>/<v>/<fax>ms <codec> noise:<l> acom:<l> i/o:<l>/<l> dBm
   video: h320:<type> tx:<video codec> <video pkts>/<video bytes> rx:<video codec> <video pkts>/<video
 bytes>
  MODEMRELAY info:<rcvd>/<sent>/<resent> xid:<rcvd>/<sent> total:<rcvd>/<sent>/<drops>
         speeds(bps): local <rx>/<tx> remote <rx>/<tx>
 Proxy <ip>:<audio udp>,<video udp>,<tcp0>,<tcp1>,<tcp2>,<tcp3> endpt: <type>/<manf>
 bw: <req>/<act> codec: <audio>/<video>
  tx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
 rx: <audio pkts>/<audio bytes>,<video pkts>/<video bytes>,<t120 pkts>/<t120 bytes>
```

Displays a truncated version of active video call information.

**Step 10** **show call history video stats**

### Example:

```
Router# show call history video stats

<ID>: <CallID> <start>ms.<index> (<start>) +<connect> +<disc> pid:<peer_id> <direction> <addr>
  dur hh:mm:ss tx:<packets>/<bytes> rx:<packets>/<bytes> <disc-cause>(<text>) dscp:<packets violation>
 media:<packets violation> audio tos:<audio tos value> video tos:<video tos value>
Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
```

```
Call agent controlled call-legs: 0
Total call-legs: 0
```

Displays call history information for SCCP video calls.

**Step 11** **show dial-peer voice**

### Example:

```
Router# show dial-peer voice

VoiceOverIpPeer565656
        peer type = voice, system default peer = FALSE, information type = voice,
        description = `',
        tag = 565656, 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 = 565656, Admin state is up, Operation state is down,
        incoming called-number = `', connections/maximum = 0/unlimited,
        bandwidth/maximum = 0/unlimited,
        DTMF Relay = disabled,
        modem transport = system,
        URI classes:
            Incoming (Request) =
            Incoming (Via) =
            Incoming (To) =
            Incoming (From) =
            Destination =
        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,
        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 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 url = system,
voice class sip rel1xx = system,
voice class sip anat = system,
voice class sip outbound-proxy = "system",
voice class sip associate registered-number = 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 reset timer expires 183 = system,
voice class sip pass-thru headers = 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 g729 annexb-all = system,
voice class sip early-offer forced = system,
voice calss sip delay-offer forced = disable,
voice class sip negotiate cisco = system,
voice class sip block 180 = system,
voice class sip block 183 = system,
voice class sip block 181 = 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 map resp-code 181 = system,
voice class sip bind control = system,
voice class sip bind media = system,
voice class sip bandwidth audio = system,
voice class sip bandwidth video = system,
voice class sip encap clear-channel = system,
voice class sip error-code-override options-keepalive failure = system,
voice class sip error-code-override cac-bandwidth failure = 488
voice class sip calltype-video = false
voice class sip registration passthrough = System
voice class sip authenticate redirecting-number  = system,
voice class sip referto-passing = system,
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
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.
```

Displays information for voice dial peers.

# Configuration Examples for Configuring DSCP Policing and Media Bandwidth Policing

## Example: Configuring the AS SIP—DSCP Policing Feature at the Global Level

The following example shows how to configure the AS SIP—DSCP Policing feature on the Cisco UBE when the incoming invite is an RPH invite:

```
Router(config)# voice class dscp-profile 1
Router(config-class)# dscp media audio priority 11
Router(config-class)# dscp media audio flsh af11

Router(config)# voice class dscp-profile 2
Router(config-class)# dscp media audio priority 60
Router(config-class)# dscp media audio flash-override af11

Router(config)# voice class dscp-profile 3
Router(config-class)# violation 10 action disconnect

Router(config)# voice class dscp-profile 4
Router(config-class)# dscp media audio immediate 2
Router(config-class)# dscp media audio flsh 63
Router(config-class)# dscp media audio flash-override af33

Router(config)# voice class dscp-profile 5
Router(config-class)# dscp media audio immediate 1
```

## Example: Applying DSCP Policing

The following example shows how to apply DSCP policing globally and at the dial peer level:

**Note** The dial peer configuration will have precedence over the global configuration.

```
Router(config)# voice service voip
Router(conf-voi-serv)# sip
Router(conf-serv-sip)# dscp-profile 2

Router(config)# dial-peer voice 1 voip
Router(config-dial-peer)# session protocol sipv2
Router(config-dial-peer)# voice-class sip resource priority dscp-profile 1
```

## Example: Configuring the AS SIP—Media Bandwidth Policing Profile at the Global Level

```
Router(config)# media profile police 1
Router(cfg-mediaprofile)# violation 20000 action disconnect no-syslog
Router(cfg-mediaprofile)# overhead audio 15
```

## Example: Applying the Media Bandwidth Policing Profile

The following example shows how to apply the media bandwidth policing profile globally and at the dial peer level:

```
Router(config)# voice service voip
Router(conf-voi-serv)# media police-profile 1

Router(config)# media class 1
Router(cfg-mediaclass)# police profile 1
Router(cfg-mediaclass)# end
Router# configure terminal
Router(config)# dial-peer voice 4 voip
Router(config-dial-peer)# media-class 1
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |

| Related Topic | Document Title |
|---|---|
| Cisco IOS voice commands | • Cisco IOS Voice Command Reference - A through C<br>• Cisco IOS Voice Command Reference - D through I<br>• Cisco IOS Voice Command Reference - K through R<br>• Cisco IOS Voice Command Reference - S Commands<br>• Cisco IOS Voice Command Reference - T through Z |
| Modular quality of service CLI overview | Modular Quality of Service Command-Line Interface Overview |

**Technical Assistance**

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Configuring DSCP Policing and Media Bandwidth Policing

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 . An account on Cisco.com is not required.

*Table 7: Feature Information for Configuring DSCP Policing and Media Bandwidth Policing*

| Feature Name | Releases | Feature Information |
|---|---|---|
| AS SIP—DSCP Policing | 15.2(2)T | The AS SIP—DSCP Policing feature provides the following functionalities:<br>• A mechanism to map the RPH to the DSCP values in the IP header for audio and video calls.<br>• A policy to match DSCP values of incoming media calls to the preconfigured value and take an action depending upon the configuration.<br>The following commands were introduced or modified:<br>**dscp media**, **dscp-profile**, **snmp enable peer-trap**, **snmp-server enable traps voice (dscp profile)**, **violation**, **voice-class sip resource priority dscp-profile**. |
| AS SIP—Media Bandwidth Policing | 15.2(2)T | The AS SIP—Media Bandwidth Policing feature introduces traffic policing on the Cisco UBE to limit media bandwidth usage to the negotiated rate. Excess traffic is dropped when the traffic rate reaches the configured maximum value.<br>The following commands were introduced or modified:<br>**media police-profile**, **media profile police**, **overhead**, **police profile**, **violation (media profile)**. |