CUBE-UCCX Inter-Operability Best Practice Guide

Introduction

This document describes various interoperability issues encountered in a contact center solution with Cisco Unified Contact Center Express (UCCX) and Cisco Unified Border Element (CUBE) deployed for PSTN reachability.

Prerequisites

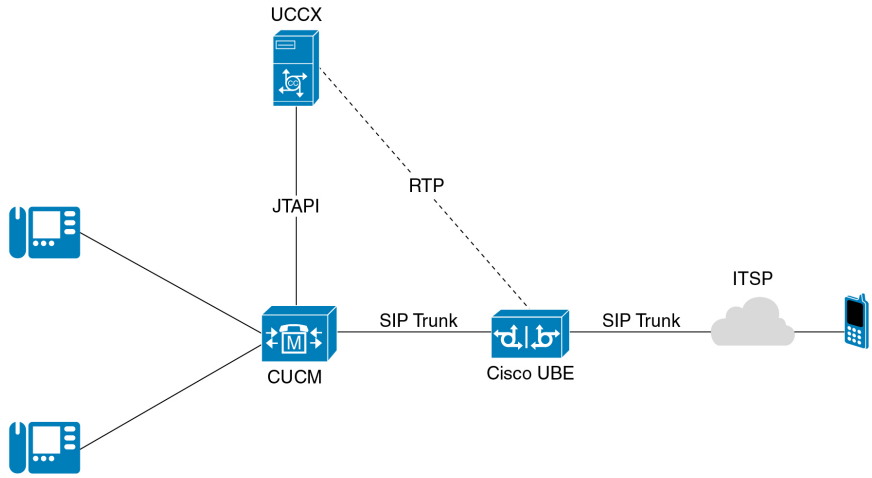
Cisco recommends that you have basic knowledge of these topics

- Configuring and using Cisco IOS voice (such as dial-peers)
- Signaling using SIP/SCCP protocols
- UCCX/CVP/Dialer platforms.

Commonly Encountered Interoperability Problems In UCCX Deployments With CUBE For PSTN Reachability

The following sub-sections discuss some of the commonly encountered interoperability problems in UCCX deployments with CUBE deployed for PSTN reachability. The document does not discuss the configuration required for the initial deployment of Cisco Unified Communications Manager (CUCM), UCCX or CUBE. Further, the document is written with the assumption that the call control protocol between the CUCM and CUBE is Session Initiation Protocol (SIP). Below is a high-level topology diagram of a UCCX deployment using CUBE for PSTN reachability.

Figure 1: Typical Deployment of UCCX With CUBE for PSTN Reachability



The commonly encountered interoperability issues are as follows:

1. DTMF Relay Failures
2. No-Way Audio for Hairpinned Calls
3. Abrupt Call Disconnection Due to Multiple Hold/Resume Sequences
4. No Hold Music or Ringback Tone for UCCX Initiated Call Transfers
5. Call Progress Analysis (CPA) Failures

DTMF Relay Failures

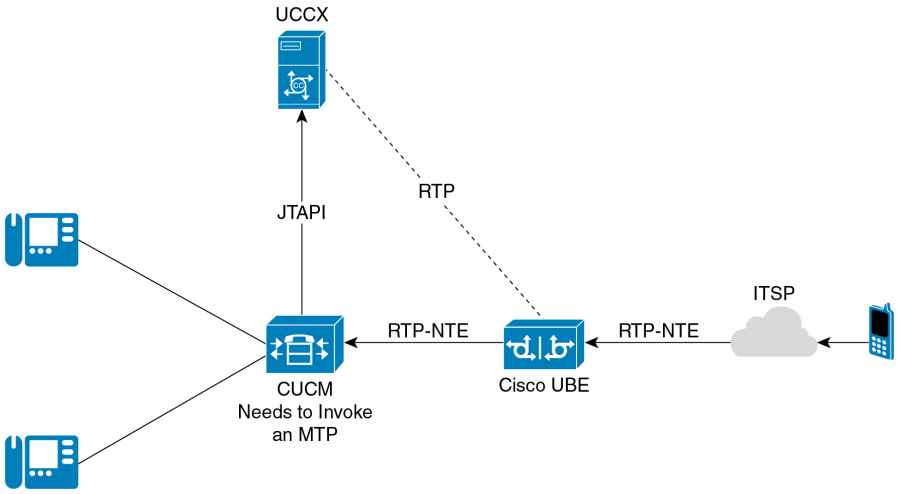
Dual Tone Multi-Frequency or DTMF is either transmitted along the signalling plane (as application protocol messages) or the media plane (within media or RTP packets). The process of transmitting digit information over IP networks—either in-band (within the media plane), out-of-band (signaling plane), or a combination of both over different call segments and usually in a mutually exclusive capacity—is called DTMF relay.

CUBE interconnects with an Internet Telephony Service Provider (ITSP) over SIP. However, most ITSPs today use the procedures of RFC 2833 or RFC 4733 to relay DTMF. RFC 2833 requires DTMF information to be included in specially formatted Real-Time Communication Protocol (RTP) packets. Hence, this is therefore an in-band method of DTMF relay. RFC 4733 iterates over RFC 2833, however, the fundamental operational principle remains the same.

UCCX supports only out-of-band DTMF relay. Therefore, in order to reliably relay DTMF tones arriving from the ITSP network to the UCCX, it is required to convert in-band tones to out-of-band signaling messages. Configuring the dial peers appropriately can natively accomplish the conversion from in-band tones to out-of-band signaling messages on CUBE. Alternatively, a Media Termination Point (MTP) registered to CUCM can also accomplish the same task.

Consider the call flow diagrammed in Figure 2 below:

Figure 2: DTMF Relay Failures from CUBE to UCCX



One of the common reasons for DTMF relay failures in the above call flow is the configuration of RTP NTE as the method of DTMF relay on the dial peer facing CUCM. As a result of this configuration, while establishing the call, the call leg between the CUBE and CUCM negotiates RTP NTE as the method of DTMF relay. As UCCX is only capable of handling out-of-band DTMF, it is required for the CUCM to convert in-band RTP NTE packets to out-of-band JTAPI signaling messages before relaying DTMF to UCCX. This conversion on CUCM is only possible if an MTP is configured and subsequently invoked for the call. If an MTP isn't invoked for the call, it results in the UCCX application receiving in-band DTMF tones. The UCCX application cannot discern any useful information from it.

Alternatively, on CUBE, you can configure the dial peers facing the CUCM that correspond to the UCCX application trigger number to negotiate an out-of-band method of DTMF relay with CUCM.

These methods include SIP KPML and/or SIP NOTIFY. Below is a configuration snippet of two dial peers created to negotiate SIP KPML and SIP NOTIFY with the CUCM.

Note: Ensure the sip trunk towards the CUBE and CUCM is set to DTMF as "no preference" and let the two end points negotiate it end to end.

Dial Peer Facing CUCM:

```
dial-peer voice 10 voip
 dtmf-relay sip-kpml sip-notify
```

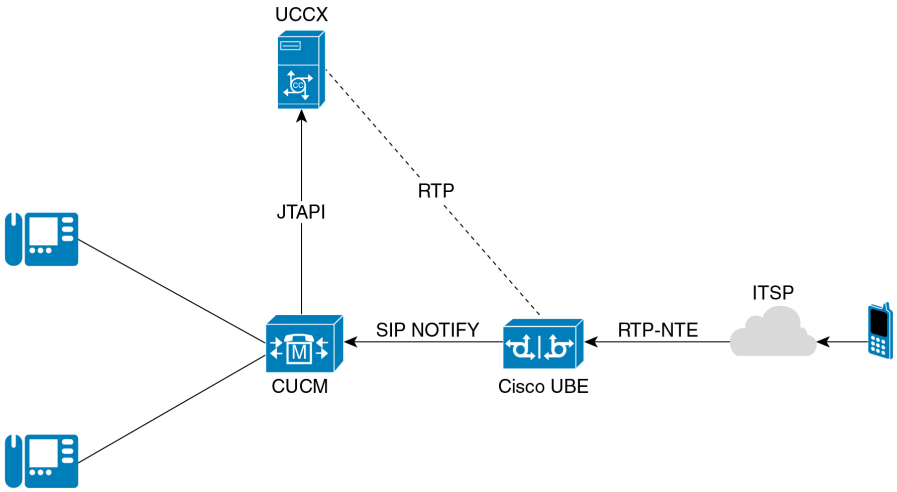
Dial Peer Facing ITSP:

```
dial-peer voice 11 voip
 dtmf-relay rtp-nte
```

CUBE negotiates and subsequently relays DTMF out-of-band through SIP NOTIFY messages to the CUCM with the above configuration. CUCM can then convert these SIP messages to JTAPI signaling before relaying to UCCX. There is no requirement for an MTP to be configured and invoked for the call. The CUBE must be in the media path however to be able to deal with this interoperability. Hence ensure that media flow-around and SDP passthrough are avoided as a part of the CUBE configuration.

```
!
      voice service voip
      default media flow-through
      sip
      default passthru content sdp
!
```

Figure 3: CUBE Relaying Out-Of-Band DTMF to CUCM

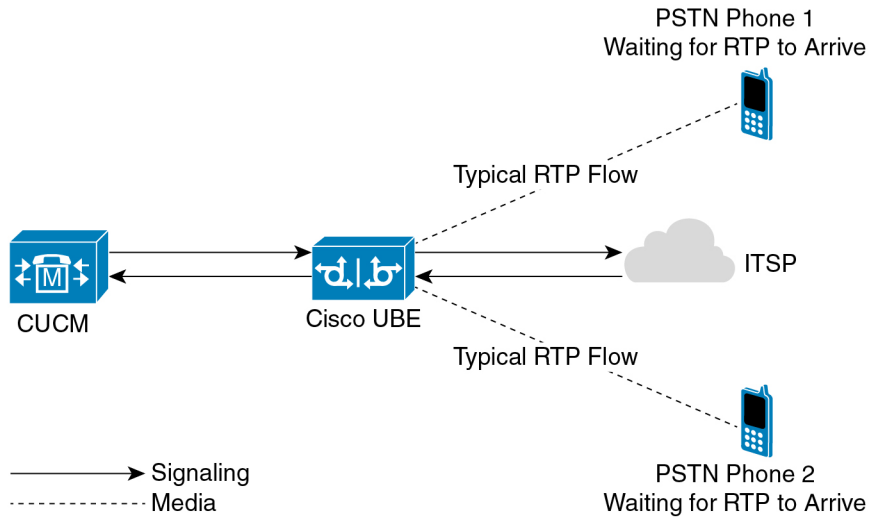


Note: The CUBE natively handles the conversion of in-band DTMF tones (RTP NTE events) it receives for the ITSP to out-of-band SIP messages that it forwards to the CUCM. However, take precautions to appropriately configure the dial peers facing the ITSP and CUCM needed on CUBE.

No-Way Audio For Hairpinned Calls

There are situations where you must transfer calls landing on UCCX to the PSTN, such that the calling and called party are both PSTN entities. Figure 4 illustrates this particular call flow referred as the hairpinned call.

Figure 4: Hairpinned Call on CUBE



In this call flow, the UCCX is a non-participant after it completes transferring the call. In this call flow, the calling and called party are on the ITSP network. RTP packets flow from the service provider network to the CUBE when the call is connected. This, then, switches these RTP packets to the service provider network. In such cases, the CUBE acts as a "media switch" sending out media packets it receives from one call leg to the other and vice versa.

Note: Most ITSPs begin streaming RTP packets for an established call only after receiving the first RTP packet from a communication peer (see RFC 7362 for more details). In this example, it is the enterprise network. However, for a hairpinned call, both the calling and called parties are on the service provider network. This creates a catch-22 situation where each call leg waits for the other to start streaming the first RTP packet leading to a no-way audio for hairpinned calls.

There are a couple of workarounds to overcome this problem from a configuration perspective, namely:

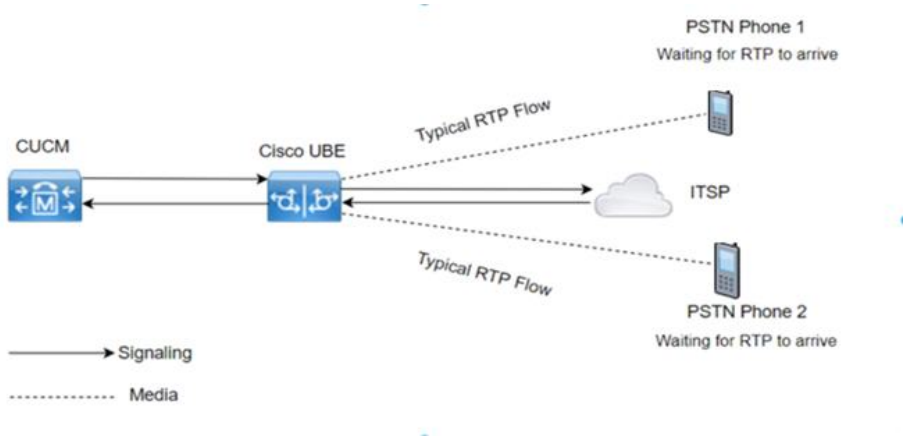
a. Invoke a CUCM-Based MTP

b. Use STUN Indication Messages

Invocation Of A CUCM-Based MTP

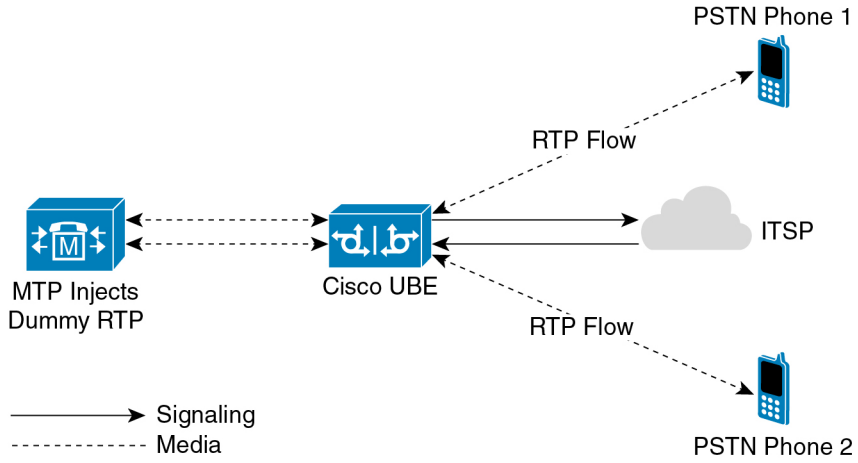
The media path is as diagrammed in Figure 5 for hair pinned calls without MTP.

Figure 5: Media Path of a Hairpinned Call on CUBE



The CUBE switches RTP packets that originate between ITSP call legs. However, with the involvement of a CUCM-based MTP, the media path is extended to involve the MTP too.

Figure 6: Media Path of Hairpinned Call with MTP Involvement



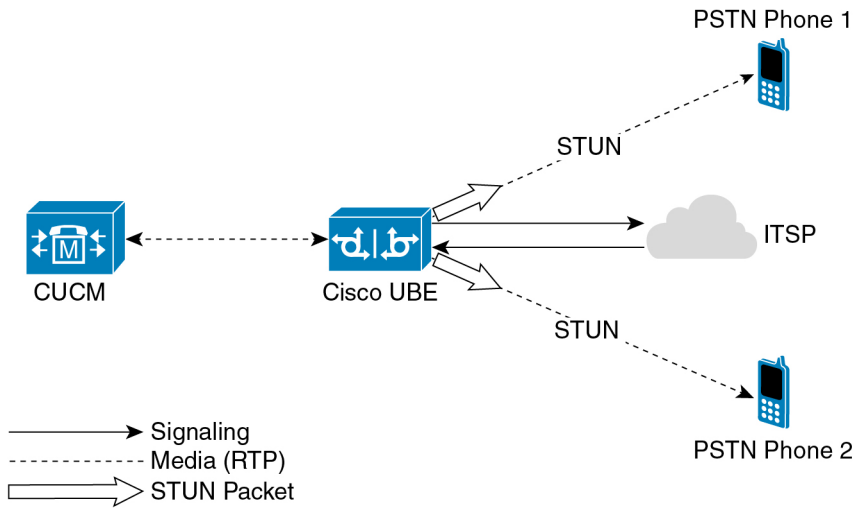
The MTP can inject RTP packets filled with silence payloads into the media stream when it is involved in the call flow. The CUBE sends the RTP packets to the service provider network to trigger RTP packet generation from the ITSP. The disadvantage of this method is the invocation of an MTP for every call established between the CUCM and CUBE.

Using Session Traversal Utilities For NAT (STUN) Indication Messages

STUN is a mechanism using which a device can discover the IP address and port number allocated to it by a NAT. This is especially useful for devices in a private address space that wish to establish communication sessions with other devices over public networks such as the Internet. For an exhaustive account of STUN, refer to RFC 5389. The CUBE can generate STUN Indication messages overloaded with a comprehension-optional attribute on the same IP address and port number as an established media

session. This results in CUBE generating STUN indication messages on the same IP and port number on both call legs of a hairpinned call to trigger RTP packet flow from the ITSP.

Figure 7: Using STUN to Trigger RTP Packet Flow



Use this configuration to enable STUN indication messages on the CUBE:

Global configuration for stun definition:

```
!
      Voice service voip
      stun
       stun flowdata agent-id 1 boot-count 4
       stun flowdata shared-secret 0 Password123$
!
```

// Create a voice class for stun parameters//

```
voice class stun-usage 200
stun usage firewall-traversal flowdata
```

// Assign the definition to the dial peer//:

```
      dial-peer voice 201 voip
       session protocol sipv2
       voice-class stun-usage 200
```

We recommend you to thoroughly test this configuration for various call flows that the CUBE is must support before committing the configuration to production.

Abrupt Call Failures Due To Multiple Hold/Resume Sequences

Call interactions such as transfers, call forwarding, and hold/resume sequences result in a large number of media target redirections resulting in several re-INVITE transactions between the CUCM and CUBE. CUBE, by default, passes across all mid-call signaling sequences such as re-INVITES and UPDATES from one call leg to the other. As a result, the CUBE forwards to the ITSP numerous re-INVITE transactions that a CUCM initiates during call-forward, call transfer or a hold/resume sequence. Most ITSPs place an upper threshold on the number of mid-call transactions (re-INVITES or UPDATES) they are willing to handle. If this threshold is surpassed, the ITSP disconnects the call leading to mid-call failures on the enterprise network. You can configure the CUBE to pass across only mid-call transactions resulting in a change of media parameters of the call such as change in the codec or the addition of a new media type (for example video or fax) to the call. This can avoid a surge of mid-call signaling exchange between the CUBE and ITSP during call interactions such as transfers, call forwarding and hold/resume sequences. This configuration ensures that mundane mid-call SIP transactions are handled locally on the CUBE, while only the transactions that result in changes of the media parameters are sent across to the ITSP. The configuration required to enable this behavior on CUBE is as follows:

```
!
    Voice service voip
    Sip
    Mid-call signalling passthrough media-change
!
```

Or on the dial peer level (This is on both inbound and outbound voip dial peers)

```
!
    Dial-peer voice 10 voip
    voice-class sip mid-call signaling passthru media-change
!
```

UCCX Call Transfer Has No Hold Music/Ringback

There are situations where callers into a call center queue are transferred to an agent and do not hear any music or ringback tone during the transfer event. The former assumes that the call transfer has failed.

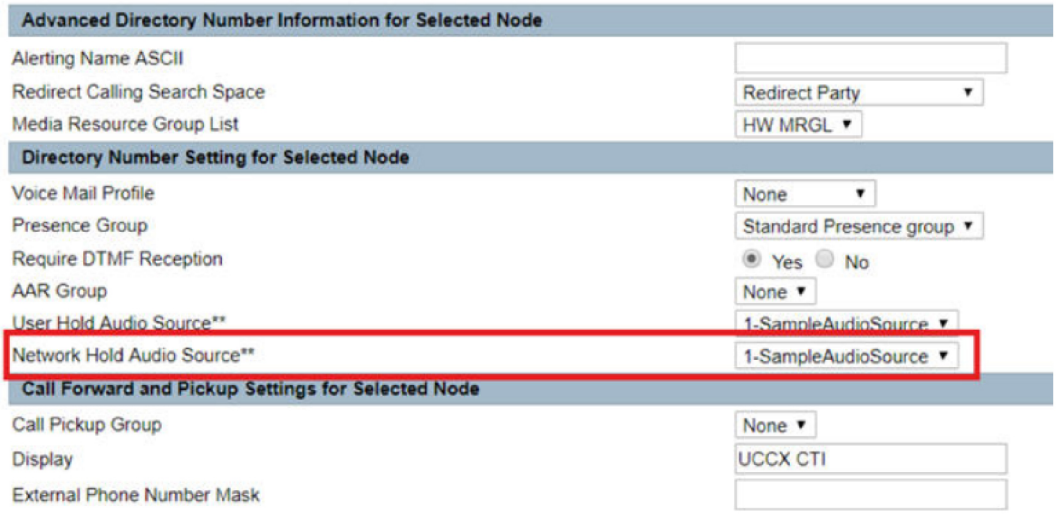
When UCCX performs a consult transfer, it's the CTI Ports' Network Hold MOH Audio Source that presents the music to the remote peer. The Call Hold and Call Unhold step in the script controls this on the UCCX.

The MOH server streams the Music On Hold, directly. As a first step, upload an Audio Source file encapsulating a ringback tone to all the CM/MOH servers in the cluster. Select the audio source file on the required UCCX Call Control Group. This step ensures that a caller hears ringback tone during a call transfer.

The following describes the configuration:

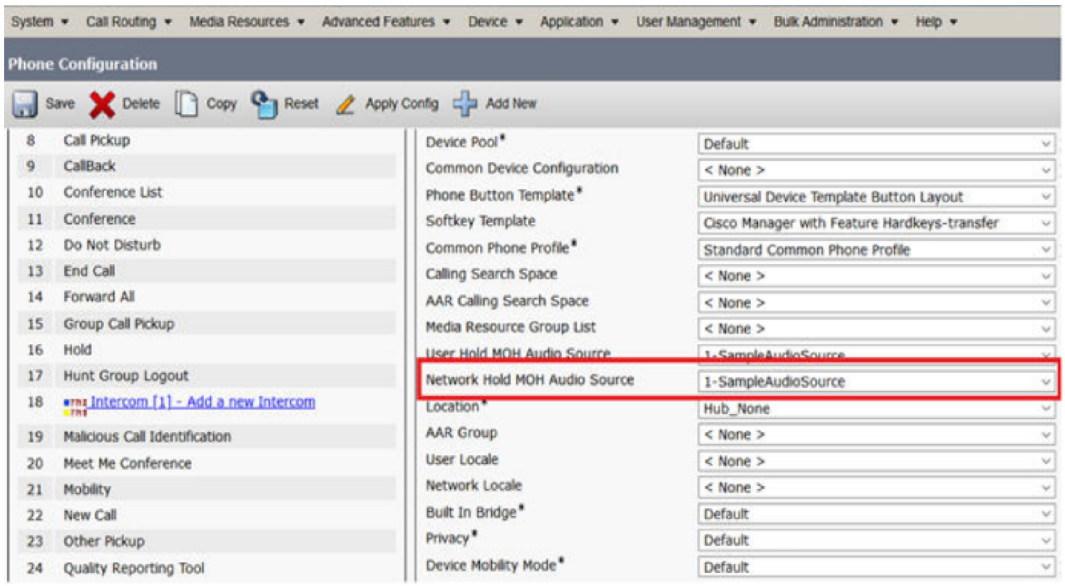
Cisco Unified CCX Administration >> Subsystems >> Cisco Unified CM Telephony >> Call Control Group >> Choose the required CCG >> Show More >> User Hold Audio Source/Network Hold Audio Source. Figure 8 provides a screenshot of the configuration page on UCCX.

Figure 8: Network Hold Audio Source on UCCX



The administrator must configure the Network Hold MOH audio under CTI devices in the UCCX. The CUCM CTI port configuration page reflects this configuration. Figure 9 provides a screenshot of the configuration page on CUCM.

Figure 9: Network Hold Audio Source on CUCM



Some customers also like to play a ringback tone while the agent's phone is ringing. Do not assume that the Annunciator Media Resource hosted on the CUCM provides this ringback. The Annunciator does provide ringback but not in the case of consult transfer. During consult transfer, the MOH resource is invoked and the audio file is played, accordingly. Upload the ringback wav audio file under the MOH audio file to play a ringback tone instead of music. This step ensures ringback from the file starts acting like the "music".

Alternatively, using the Blind Transfer mechanism triggers the Annunciator resource to play local ringback from CUCM.

In either case, the MRGL of the SIP trunk invokes the MOH or Annunciator resource. Please ensure that the resources are registered, present and the region settings are appropriately setup.

RTP Source Validation

Bi-directional RTP streams setup between two entities as a result of an offer/answer exchange require either entity to stream RTP to IP address and port specified in the peer entities' SDP body. However, an entity is free to choose any source port number for RTP transmission. Accordingly, RTP sources are classified as either symmetric or asymmetric. A symmetric RTP source uses the same IP address and port number for media transmission and reception. An asymmetric RTP source uses a different IP address and/or port number for media transmission and reception.

Parties with malicious intent are always trying to find exploits and circumvent current security and RTP constructs. RTP Source Validation is a feature integrated in Cisco Voice Routers that allows them to drop untrusted, inbound RTP traffic.

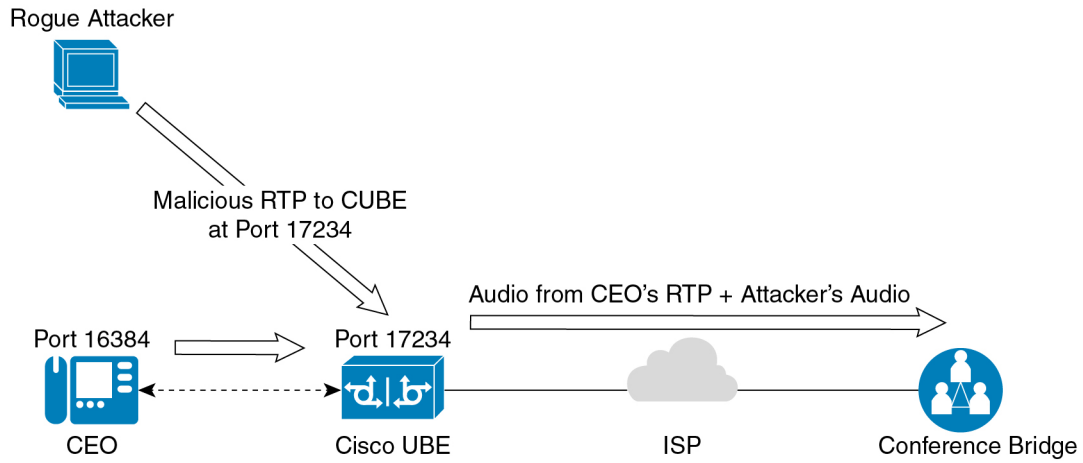
In IOS and IOS-XE, this feature makes the Voice Routers drop inbound RTP Traffic from unknown IP addresses or ports. In other words, the Voice Router drops packets received from an IP Address or Port that was not negotiated through the voice protocol signaling.

This feature was introduced in 15.5(3)M9 IOS versions and is present in all IOS-XE version routers (ISR G3/4K routers).

On classic IOS, CUBE defaults to accept RTP media from multiple sources. It mixes the audio out at the destination. This behavior can allow hackers to take advantage. A hacker can play malicious audio packets into the call, if the call's destination port is known using either unencrypted SIP signaling or through a brute-force attack. One use case is an attacker playing disruptive audio at the conference bridge's side of a CEO's investor disclosure call.

Figure 10 illustrates this attack vector.

Figure 10: Exploiting RTP Vulnerabilities



In Figure 10, the IP Phone and the CUBE SBC negotiate RTP port 16384 within the SIP dialogue. The CUBE expects to source the RTP from port 16384. However, if a malicious attacker sends its rogue packet sourced from a different port and the same IP address to the CUBE without validating the source port, it ends up passing the rogue RTP audio towards the PSTN network. This can disrupt the audio on the conference bridge.

How Source Validation Affects CUCM/UCCX Deployments For MOH Streaming

When CUCM defaults a call to hold status, the CUCM advertises a well-known "dummy" port (port number 4000) through signaling. However, it actually streams the RTP from an ephemeral port (32768-65535). This behavior is a result of setting the "Duplex Streaming Enabled" Enterprise Parameter on CUCM to "false". A consequence of this behavior is that the CUBE drops RTP packets encapsulating hold music leading to silence for the party placed on hold (see the signaling exchange below). Hence, set "Enable Duplex Streaming" parameter to "true" and avoid this behavior. The screenshot provided in Figure 11 highlights the "Enable Duplex Streaming" enterprise parameter setting on CUCM.

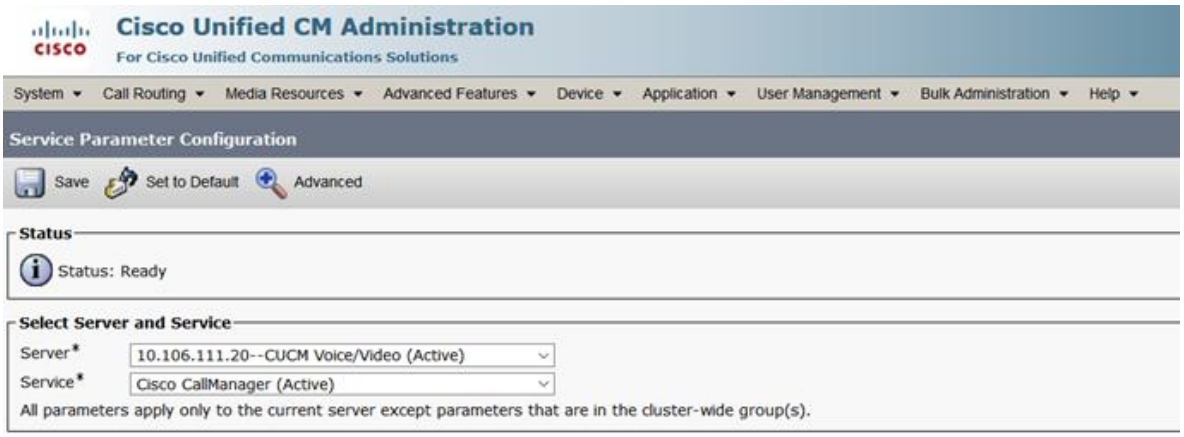


Figure 11: Duplex Streaming Enabled Parameter Set to "False"

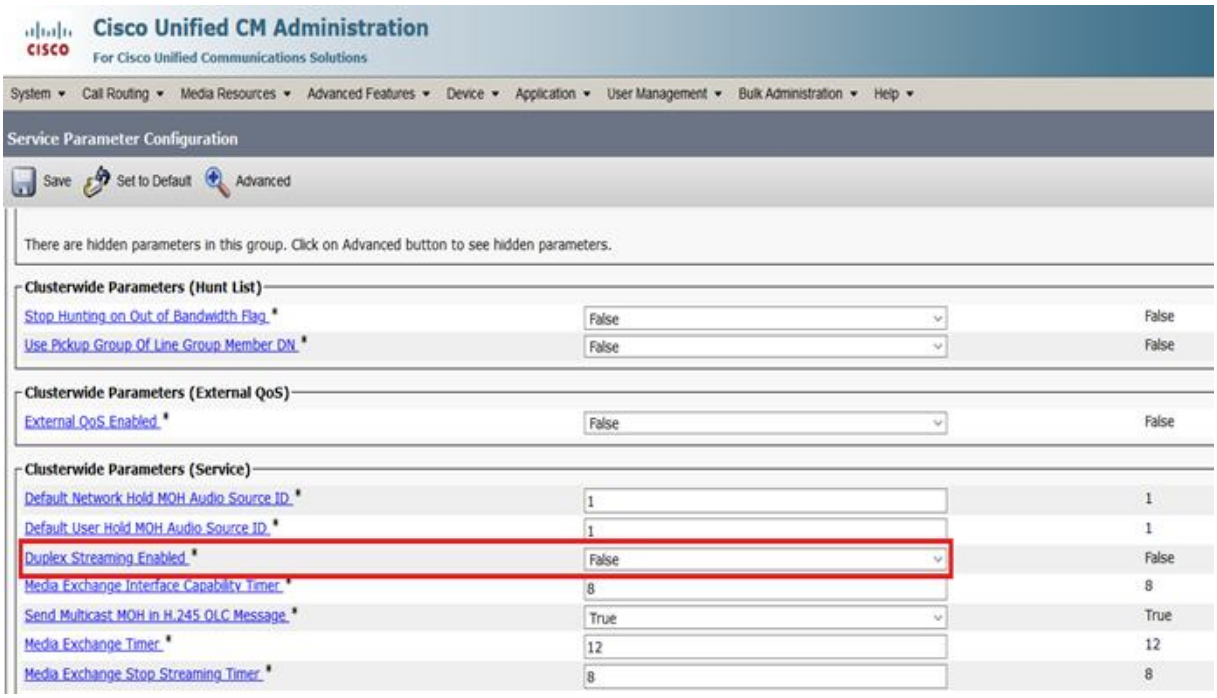
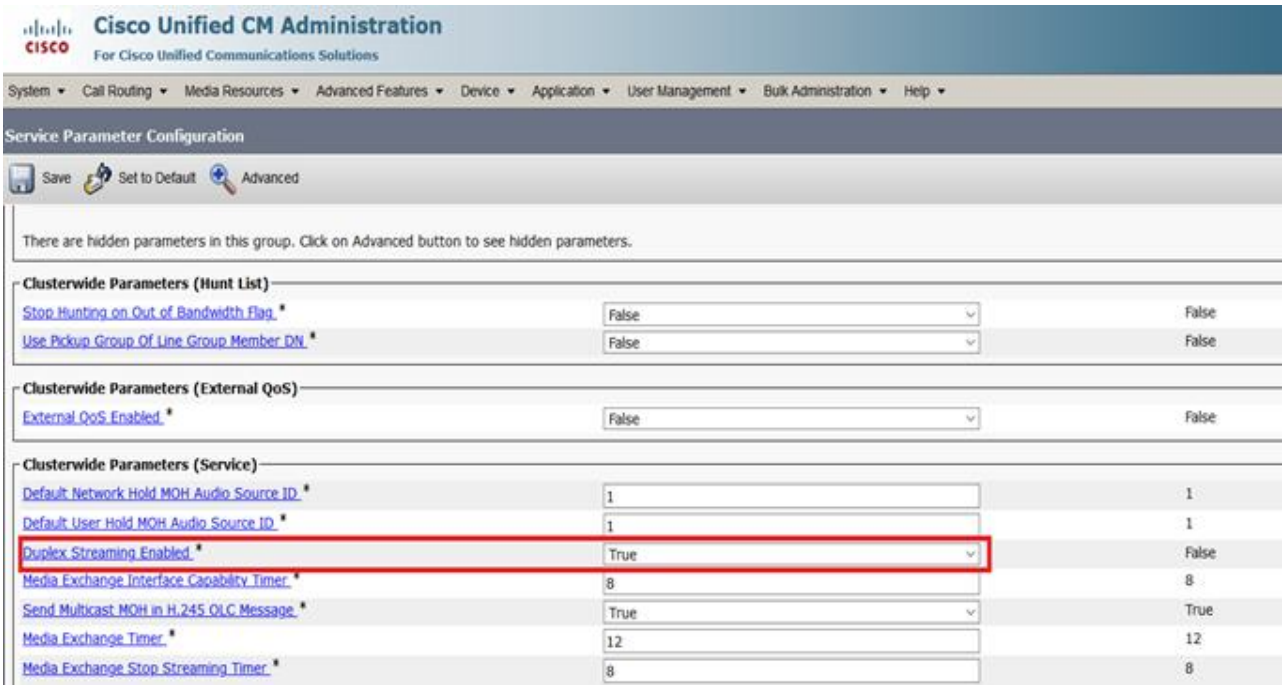


Figure 12: Duplex Streaming Enabled Parameter Changed to "True"



The following debug ccsip messages snipped on CUBE provides an overview of the port numbers negotiated during SIP signaling.

```
//-1/xxxxxxxxxxxx/SIP/Msg/ccsipDisplayMsg:

Received:

ACK sip:6002@Router-IP-Address:5060 SIP/2.0

Via: SIP/2.0/UDP CUCM-IP-Address:5060;branch=z9hG4bK4a424fed85

From: <sip:12345@CUCM-IP-Address>;tag=4071~842740d9-7186-4740-ada2-
23e5d1b91316-46404063

To: <sip:6002@Router-IP-Address>;tag=2FF652-51D

Date: Thu, 15 Jan 2019 16:39:50 GMT

Call-ID: 3EDDD9E4-614B11E9-800D9C4B-C5465DB2@Router-IP-Address

User-Agent: Cisco-CUCM12.0


c=IN IP4 CUCM-IP-Address (MoH Server)

t=0 0

m=audio 4000 RTP/AVP 0

a=X-cisco-media:umoh

a=ptime:20

a=rtpmap:0 PCMU/8000

a=sendonly
```

Due to the “Enable Duplex Streaming” enterprise parameter settings set to False by default, the SIP ACK has an SDP body that advertises port number 4000. However, the CUCM still streams hold music sourced from an ephemeral port (32768-65535). This results in the CUBE dropping hold music packets as they are sourced from a different port than the one advertised in the SDP (which is 4000).

As mentioned previously, IOS and IOS-XE software versions tend to handle source port validation differently – IOS-XE requires RTP packets to be sourced from the same port advertised in SDP, whereas IOS ignores the source port from which packets are sourced. However, enable the following commands on IOS releases starting from and after 15.5(3)M9 to enforce source port validation.

Configuration

```
!
Configure terminal
voice rtp source-filter
!
```

Use the “debug voip rtp error” debug on IOS to verify dropping packets based on the source port behavior. This “debug voip rtp packet” snippet demonstrates how a packet drops due to failure of source port validation.

```
voip_rtp_recv_fs_input:ERROR Port validation failed, dropping RTP packet.

Expected port: 4000, Received port: 16384
```

Use the following command to override the source port validation feature on IOS-XE platforms for streams that are “recvonly” from the perspective of CUBE (for example, the stream setup to send hold music from CUCM).

```
!
      Voice service voip
!
      Nat force-on
!
```

Note: This command is ineffective for streams that are not “recvonly” from the perspective of CUBE. This does not work when pass-through content sdp is configured.

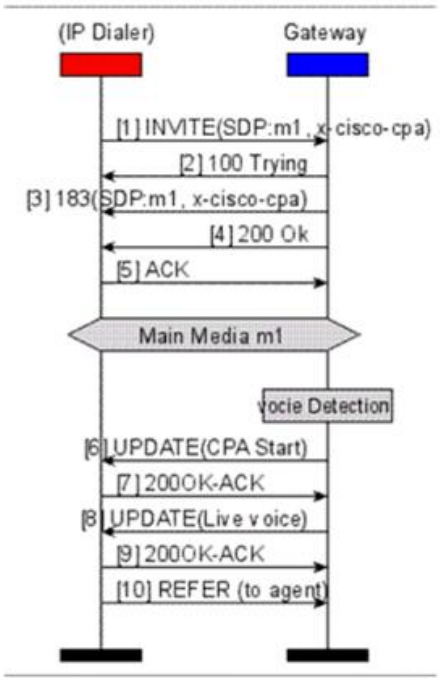
UCCX Outbound Dialer With Call Progress Analysis

Call progress analysis (CPA) is a DSP algorithm that analyzes the Real-Time Transport Protocol (RTP) voice stream to look for special information tones (SIT), fax or modem tones, human speech, and answering machine tones. You can use call progress analysis to ensure that an outbound dialer only connects a human speaker to a call center agent instead of an answering machine/bot/voicemail.

Conventionally, CPA was supported on a SIP-POTS gateway as every POTS call leg requires the involvement of a DSP. Owing to large adoption of SIP trunking to interconnect with telephony service providers, support for this feature was extended to CUBE. However, to ensure this feature works on CUBE, it is required to configure and invoke an LTI transcoder on CUBE. Figure 13 provides an overview of signaling messages exchanged between the dialer and CUBE during call progress analysis.

Figure 13: Typical CPA Call Flow

Typical CPA call flow



There are certain aspects that need to be considered for CPA to work as expected though the CPA mechanism has been tested and validated to work well on CUBE.

Connection-Reuse

Use the "connection-reuse" command for SIP transmitted over an unreliable transport protocol, such as UDP, to transmit SIP messages sourced from the 5060/5061 SIP listen port. An ephermal port is the default source port for messages without the "connection-reuse" command. Another consequence of the connection-reuse command is that the CUBE uses the source port of incoming messages as the destination port for outgoing responses. The SIP Dialer on UCCX expects responses to be sent to the port number specified in the Via: header field of the SIP request though it sources SIP requests from a given port number. The UCCX Dialer drops responses from CUBE, if it does not honor the port number in the Via: header field. Configure the "connection-reuse via-port" command to overcome this behavior and ensure that CUBE sends responses to the port number advertised in the Via: header field value. Below is a configuration snippet of the command.

In sip-ua mode:

```
!
sip-ua
connection-reuse via-port
!
```

In voice class tenant mode:

```
!
voice class tenant 1
connection-reuse via-port
!
```

Invoking A Transcoder For DTMF Interworking And CPA Concurrently

CUBE setup for inter working between Voice In-band DTMF and RTP-NTE requires the configuration and subsequent invocation of an LTI Transcoder. There are two requirements to ensure that an LTI transcoder is invoked for interworking raw in-band DTMF tones and RTP-NTE, namely:

- The dial-peer for the call leg negotiating raw-inband DTMF must not have any method of DTMF configured.
- The dial-peer for call leg negotiating RTP-NTE must have dtmf-relay rtp-nte configured.

Raw Inband to RTP-NTE DTMF inter-working is not supported with CPA. This means you cannot detect CPA if the UCCX dialer uses raw Inband DTMF relay mechanism. Hence ensure interop with Service provider to use RTP-NTE based DTMF so that no transcoding is required for the interworking of Raw Voice In-band < -- > RTP-NTE.

CPA detection fails if for the same call, the transcoder is invoked to interwork DTMF and detect SIT tones. Therefore, it is required to ensure that for CPA calls, the LTI transcoder is used only for one function – namely the detection of SIT tones. It is therefore of paramount importance to verify that the ITSP supports RTP-NTE if there is a requirement to configure and use Call Progress Analysis.

Other things to keep in mind while configuring the CPA solution:

- Only SIP-to-SIP Early Offer (EO-to-EO) call flows are supported on the CUBE.
- Session Description Protocol (SDP) passthrough and flow-around media calls are not supported.
- Only the G711 flavor of codec is supported.

- CUBE High Availability (HA) is not supported.
- Skinny Client Control Protocol (SCCP)-based digital signal processor (DSP) farming is not supported.

Obtaining Documentation And Submitting A Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. The RSS feeds are a free service.