Contents

Introduction

This document describes troubleshoot steps for media forking from a Cisco IP phone to record calls on a MediaSense server.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Unified Communications Manager (CUCM)
- Cisco MediaSense

Components Used

The information in this document is based on these software and hardware versions:

- CUCM Version 10.5.2.10000-5
- Cisco MediaSense 10.0.1.10000-95

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Infromation

Cisco MediaSense is a network-based platform that provides voice and video media recording capabilities for devices in network using Session Initiation Protocol (SIP). Fully integrated into Cisco's Unified Communications architecture, MediaSense automatically captures and stores every Voice over IP (VoIP) conversation on devices which are appropriately configured CUCM.

1. MediaSense accepts audio codec in the below formats:
   - g.711 µLaw and aLaw
   - g.722
   - g.729, g.729a, g.729b
   - Advance Audio Coding - Low Delay (AAC-LD) also known as MPEG Audio Layer 4 - Low-overhead MPEG-4 Audio Transport Multiplex (MP4A/LATM)
2. MediaSense Video in H.264 encoding

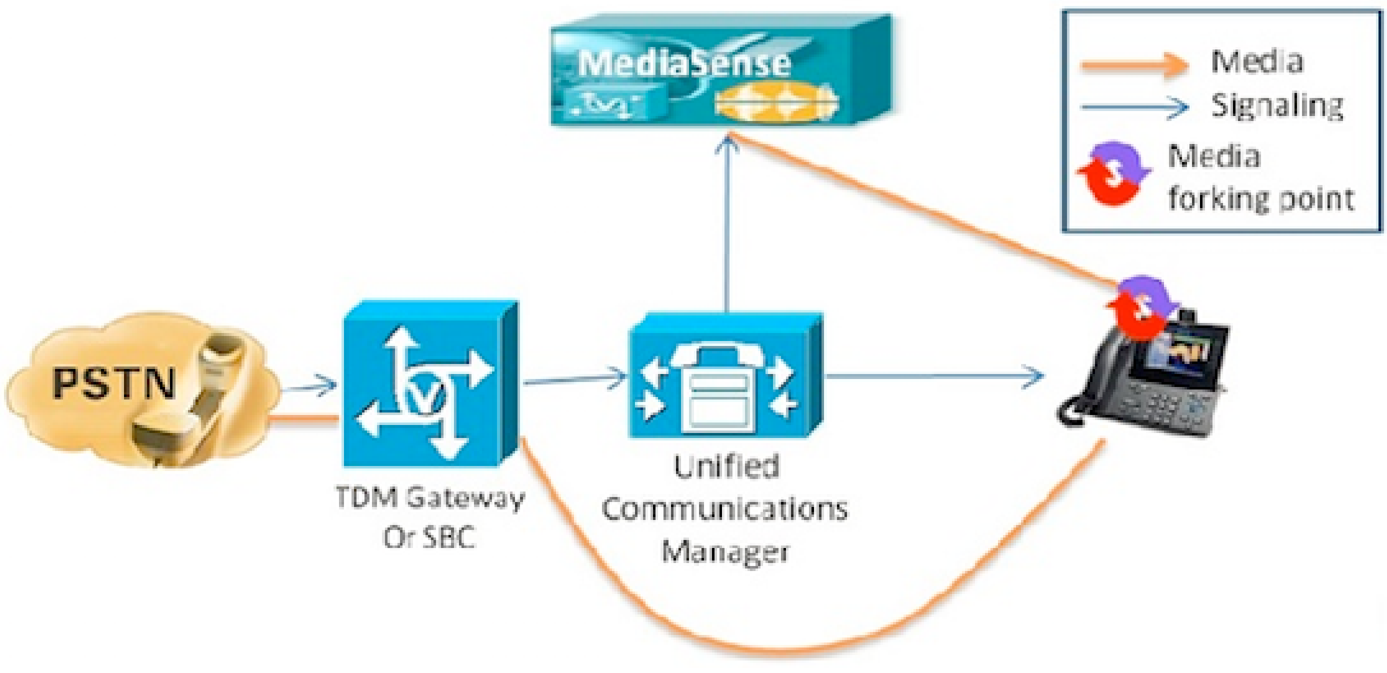
Scenario

1. Basic Unified Communications Manager deployment - Internal-to-External
2. Basic Unified Communications Manager deployment - Internal-to-Internal

From the perspective of MediaSense, there is actually no difference between two scenarios.

In both cases, media forked by a phone is sent to the recording device where the forked streams are captured. They are distinguished here because there is a significant difference in their behavior at the solution level.

As shown in this image, Unified Communications Manager Deployment - Internal-to-External.

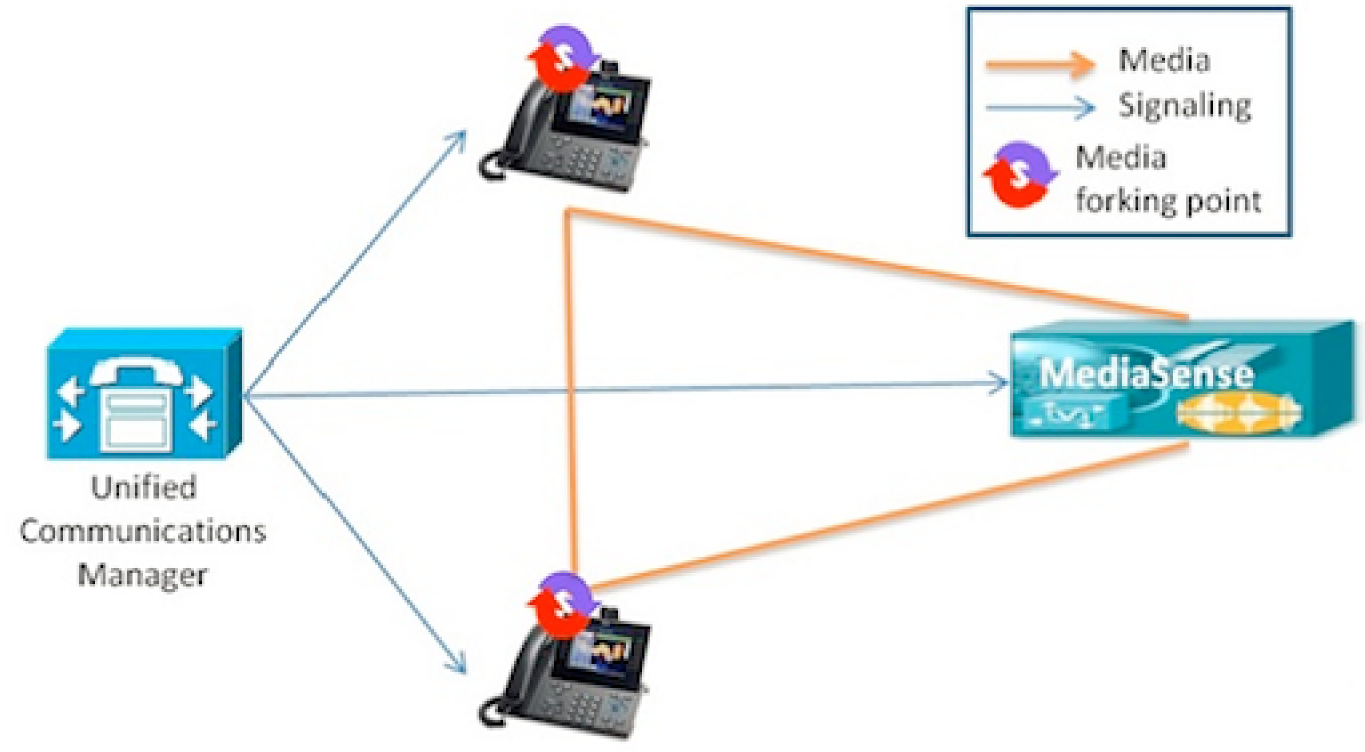


This shows a basic Unified Communications Manager deployment where Cisco IP phone call with an external caller is recorded. This applies to both inbound and outbound calls, as long as the inside phone is configured with an appropriate recording profile.

Once the connection is established from a signaling perspective, media flows directly from the forking phone to the recording server.

If the call is transferred away from this phone, the recording session ends. The next segment of the call will be captured only if the phone which takes up the call is configured for recording.

As shown in this image, Unified Communications Manager Deployment - Internal-to-Internal.



This shows a basic Unified Communications Manager deployment where the call is between internal users who are within the enterprise. It is important that one of the phones be configured for recording. In case both phones are configured for recording, then two separate recording sessions will be captured.

Troubleshoot

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

Step 1.Check the configuration on MediaSense and CUCM.

CUCM

- Controlled devices and Permission information in Application user (AXL).
- Recording profile and destination address
- SIP trunk pointing to MediaSense.
- Route Pattern

MediaSense

You can verify basic configuration using **show tech call_control_service** command on MediaSense command line after system installation.

This command displays information about the Cisco MediaSense Call Control Service that runs on the system.

The Cisco MediaSense Call Control Service should be running for this command to execute successfully.

System Information captured in the output.

```
 admin:show tech call_control_service

<html> <head> <title>mediasense</title> </head> <body> <pre>
--------------------------------
 Core: ver=10.0.1 FCS, op=SHORT
Started at Mon Jul 13 10:55:53 PDT 2015
Report at Tue Jul 21 02:05:26 PDT 2015
Running at mediasense, processors=6, pId=28270
framework: state=In Service; {AMS_ADAPTER= IN_SERVICE, SIP_ADAPTER=IN_SERVICE,
RECORDING_ADAPTER=IN_SERVICE}
logLevel=DEBUG, traceMask=0x307, DEBUG traceMask=0x100

System Info:
Memory: used=46.509 MB(13.671 MB), alloc=790.458 MB(0.0 MB)
CPU: avrLoad=0.37, procTime=00:10:18
Threads=176, peakThreads=224
```

Recording Sessions information in the **show tech call_control_service** output.

```
SessionManagerImpl: size=0
Recording Sessions: started=17, completed=17 (100.0000%), errors=0, processing=0,
maxProcessing=1, meanTime=38.310 sec, stDev=76.242 sec, maxTime=00:05:16, lastTime=38291 mSec
Recording Setup Time: started=17, completed=17 (100.0000%), errors=0, processing=0,
maxProcessing=1, meanTime=201 mSec, stDev=34 mSec, maxTime=308 mSec, lastTime=142 mSec
```

SIP Adapter information in the **show tech call_control_service** output.

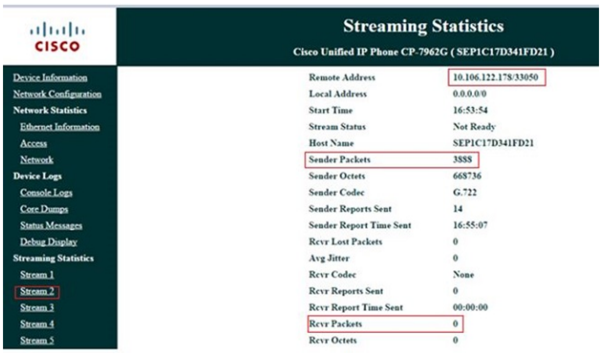
```
Sip Adapter:
LocalAddress=10.106.122.178:5060; RemoteAddresses [sip:10.106.122.174:5060
sip:10.106.122.175:5060 ], controlTransport=tcp
based on Cisco Caffeine SIP Stack, version=3.1.3.502, nonBlockingTCP=true,
closeConnectionOnTimeout=false
state=AcceptCalls, blockingMode=NONE
SdpUtil: m=audio %d RTP/AVP 102 0 8 9 18, m=video %d RTP/AVP 97
Executor: activeCount=0, poolSize=0, largestPoolSize=2, queueSize=0
```

> **Tip**: Refer to in order to setup Call Recording

Step 2.

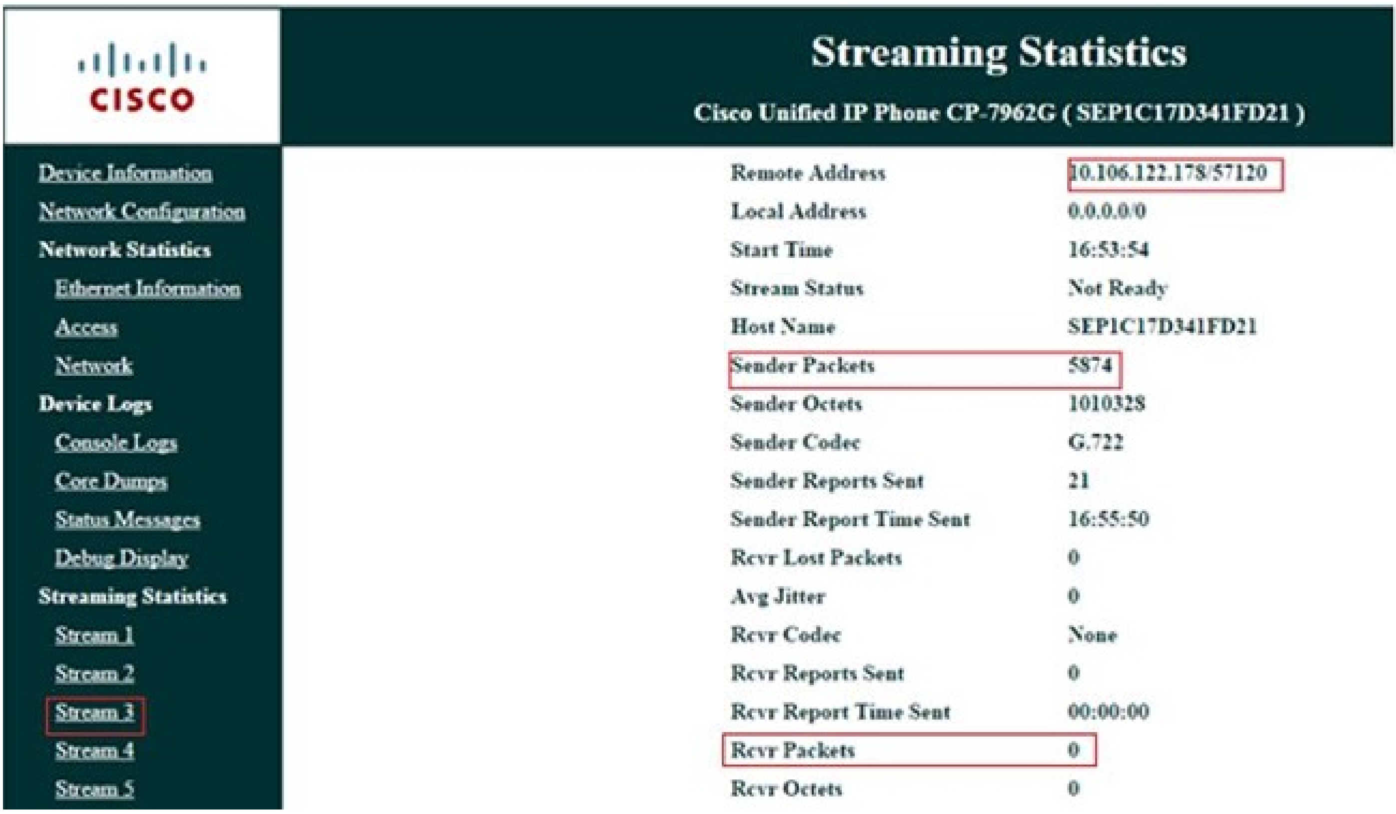
Stream 1 will be the call to the external caller. Stream 2 will contain the information about the forked call to the MediaSense server. Reciever packets will always remain zero for forked calls.

As shown in this image, Near end media streaming to MediaSense.



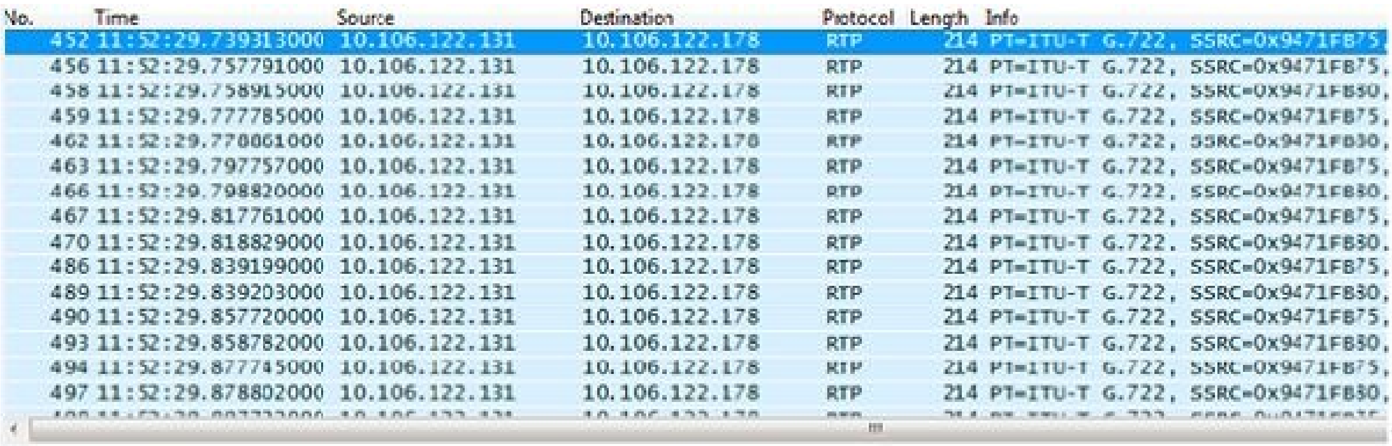
Far end media streaming to MediaSense

As shown in this image, Streaming information for Far-end media recieved in Stream 1 is forked in Stream 3.



You can verify it by taking Packet Capture on the phone.

As shown in this image, Phone PCap.



No.	Time	Source	Destination	Protocol	Length	Info
452	11:52:29.739313000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
456	11:52:29.757791000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
458	11:52:29.758915000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,
459	11:52:29.777785000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
462	11:52:29.778061000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,
463	11:52:29.797757000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
466	11:52:29.798820000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,
467	11:52:29.817761000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
470	11:52:29.818829000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30.
486	11:52:29.839199000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
489	11:52:29.839203000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,
490	11:52:29.857720000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
493	11:52:29.858782000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,
494	11:52:29.877745000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB75,
497	11:52:29.878802000	10.106.122.131	10.106.122.178	RTP	214	PT=ITU-T G.722, SSRC=0x9471FB30,

Tip: Refer to Collecting Packet Capture from IP Phones

Step 3.

The example taken here contains IP call from SIP phone with Extension 4011 to SCCP phone with Extension 4009. The recording destination number is 7878.

CUCM Log Analysis

INVITE sent from SIP phone to CUCM.

```
 06053008.002 |08:39:47.013 |AppInfo  |SIPTcp - wait_SdlReadRsp: Incoming SIP TCP message from
10.106.122.153 on port 53979 index 44 with 2126 bytes:
[50171,NET]
INVITE sip:4009@10.106.122.174;user=phone SIP/2.0
Via: SIP/2.0/TCP 10.106.122.153:53979;branch=z9hG4bK22e1618f
From: "4011" <sip:4011@10.106.122.174>;tag=203a0782d99f04115d77007a-7abfc08c
To: <sip:4009@10.106.122.174>
Call-ID: 203a0782-d99f000c-57711fea-6ba95503@10.106.122.153
Max-Forwards: 70
Date: Thu, 16 Jul 2015 15:39:46 GMT
CSeq: 101 INVITE
 User-Agent: Cisco-CP8945/9.4.2
Contact: <sip:48a499a0-f78e-4baa-a287-5c6eeb0f2fe7@10.106.122.153:53979;transport=tcp>;video
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE,SUBSCRIBE,INFO
Remote-Party-ID: "4011" <sip:4011@10.106.122.174>;party=calling;id-
type=subscriber;privacy=off;screen=yes
Supported: replaces,join,sdp-anat,norefersub,resource-priority,extended-refer,X-cisco-
callinfo,X-cisco-serviceuri,X-cisco-escapecodes,X-cisco-service-control,X-cisco-srtp-fallback,X-
cisco-monrec,X-cisco-config,X-cisco-sis-7.0.0,X-cisco-xsi-8.5.1
Allow-Events: kpml,dialog
Recv-Info: conference
Recv-Info: x-cisco-conference
Content-Length: 986
Content-Type: application/sdp
Content-Disposition: session;handling=optional

v=0
o=Cisco-SIPUA 15743 0 IN IP4 10.106.122.153
s=SIP Call
b=AS:2000
t=0 0
m=audio 16420 RTP/AVP 102 9 0 8 116 18 101
c=IN IP4 10.106.122.153
a=trafficclass:conversational.audio.avconf.aq:admitted
a=rtpmap:102 L16/16000
a=rtpmap:9 G722/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 iLBC/8000
a=fmtp:116 mode=20
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
```

UserAgent is a Cisco 8945 IP Phone sending an As to CUCM.

CUCM sends ACK to SIP phone when SCCP phone answers the call and the session gets established.

```
06053236.001 |08:39:49.777 |AppInfo  |SIPTcp - wait_SdlSPISignal: Outgoing SIP TCP message to
10.106.122.153 on port 53979 index 44
[50174,NET]
SIP/2.0 200 OK
Via: SIP/2.0/TCP 10.106.122.153:53979;branch=z9hG4bK22e1618f
From: "4011" <sip:4011@10.106.122.174>;tag=203a0782d99f04115d77007a-7abfc08c
To: <sip:4009@10.106.122.174>;tag=16789~78868996-a8aa-4784-b765-86098b176d95-32833193
Date: Thu, 16 Jul 2015 15:39:47 GMT
```

```
Call-ID: 203a0782-d99f000c-57711fea-6ba95503@10.106.122.153
CSeq: 101 INVITE
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY
Allow-Events: presence
Supported: replaces
Server: Cisco-CUCM10.5
Call-Info: <urn:x-cisco-remotecc:callinfo>; security= NotAuthenticated; orientation= to; gci= 1-
7171; isVoip; call-instance= 1
Send-Info: conference, x-cisco-conference
Remote-Party-ID: <sip:4009@10.106.122.174>;party=called;screen=yes;privacy=off
Remote-Party-ID: <sip:4009@10.106.122.174;user=phone>;party=x-cisco-original-called;privacy=off
Contact: <sip:4009@10.106.122.174:5060;transport=tcp>
Content-Type: application/sdp
Content-Length: 435

v=0
o=CiscoSystemsCCM-SIP 16789 1 IN IP4 10.106.122.174
s=SIP Call
c=IN IP4 10.106.122.131
b=AS:64
t=0 0
m=audio 18840 RTP/AVP 9 101
a=ptime:20
a=rtpmap:9 G722/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=trafficclass:conversational.audio.aq:admitted
```

Phone presses Record soft key indicating that the user invokes the recording feature.

```
06053271.001 |08:39:52.681 |AppInfo  |StationInit: (0000045) SoftKeyEvent
softKeyEvent=74(Record) lineInstance=1 callReference=32833194.
```

Codec gets locked for recording.

```
06053274.002 |08:39:52.681 |AppInfo  | StationCdpc: star_MediaExchangeAgenaQueryCapability -
Device SEP1C17D341FD21, codec locked due to recording,  codecType=6
```

Built-in Bridge (BiB) resource gets allocated.

```
06053309.000 |08:39:52.682 |SdlSig   |AllocateBibResourceRes
|resource_rsvp                    |MediaResourceCdpc(1,100,139,52)
|BuiltInBridgeControl(1,100,239,6) |1,100,14,269032.3452^10.106.122.131^SEP1C17D341FD21 |[R:N-
H:0,N:0,L:0,V:0,Z:0,D:0]  CI=32833195 BridgeDn= b00123906001 Pid=100,1,63,45 SsType=16777245
SsKey=43 deviceCap=0
```

CUCM dials in BiB resource.

```
06053318.008 |08:39:52.683 |AppInfo  ||PretransformCallingPartyNumber=
|CallingPartyNumber=
|DialingPartition=
|DialingPattern= b00123906001
|FullyQualifiedCalledPartyNumber= b00123906001
```

BiB then dials to MediaSense recording number 7878.

```
06053358.013 |08:39:52.686 |AppInfo  ||PretransformCallingPartyNumber=b00123906001
|CallingPartyNumber= b00123906001
|DialingPartition=
|DialingPattern= 7878
|FullyQualifiedCalledPartyNumber= 7878
```

INVITE is sent to MediaSense.

```
06053416.001 |08:39:52.690 |AppInfo  |SIPTcp - wait_SdlSPISignal: Outgoing SIP TCP message to
```

```
10.106.122.178 on port 5060 index 71
[50176,NET]
INVITE sip:7878@10.106.122.178:5060 SIP/2.0
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK14432e0a687
From: <sip: 4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
To: <sip:7878@10.106.122.178>
Date: Thu, 16 Jul 2015 15:39:52 GMT
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
Supported: timer,resource-priority,replaces
Min-SE: 1800
User-Agent: Cisco-CUCM10.5
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY
CSeq: 101 INVITE
Expires: 180
Allow-Events: presence, kpml
Supported: X-cisco-srtp-fallback
Supported: Geolocation
Call-Info: <sip:10.106.122.174:5060>;method="NOTIFY;Event=telephone-event;Duration=500"
Cisco-Guid: 3841694080-0000065536-0000000071-2927258122
Session-Expires: 1800
P-Asserted-Identity: <sip:4009@10.106.122.174>
Remote-Party-ID: <sip:4009@10.106.122.174>;party=calling;screen=yes;privacy=off
Contact: <sip:4009@10.106.122.174:5060;transport=tcp>;isFocus
Max-Forwards: 70
Content-Length: 0
```

200 OK from MediaSense when the recording call is established.

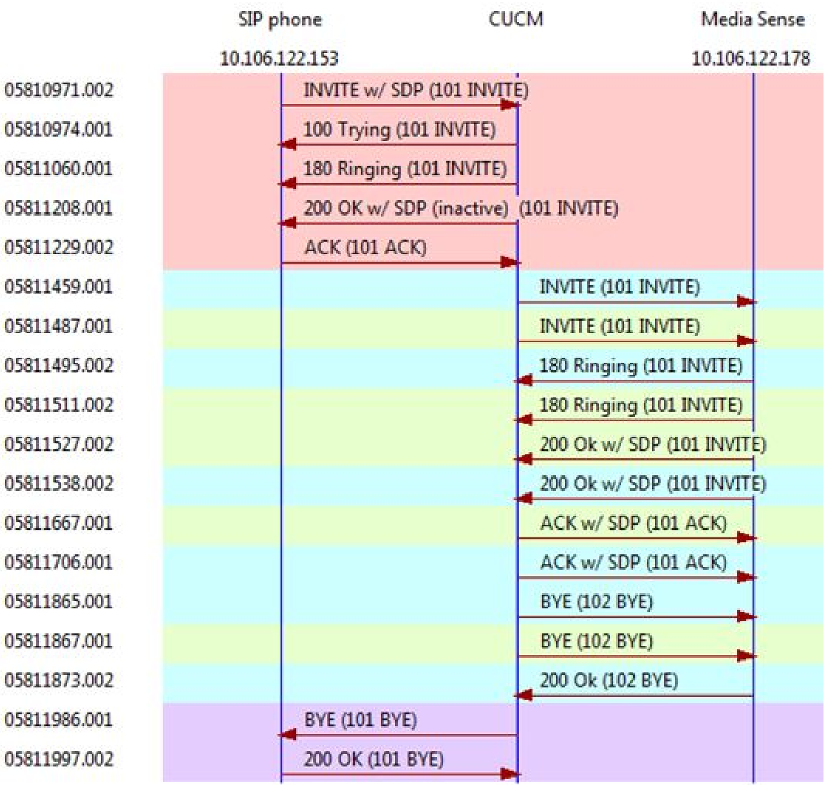
```
06053554.002 |08:39:52.831 |AppInfo  |SIPTcp - wait_SdlReadRsp: Incoming SIP TCP message from
10.106.122.178 on port 5060 index 71 with 1013 bytes:
[50181,NET]
SIP/2.0 200 Ok
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK14432e0a687
To: <sip:7878@10.106.122.178>;tag=ds606d34cb
From: <sip:4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
CSeq: 101 INVITE
Content-Length: 313
Contact: <sip:7878@10.106.122.178:5060;transport=tcp>
Content-Type: application/sdp
Allow: INVITE, BYE, CANCEL, ACK, NOTIFY, INFO, UPDATE
Server: MediaSense/10.x

v=0
o=CiscoORA 3197 1 IN IP4 10.106.122.178
s=SIP Call
c=IN IP4 10.106.122.178
t=0 0
m=audio 42120 RTP/AVP 102 0 8 9 18
a=rtpmap:102 MP4A-LATM/90000
a=fmtp:102 profile-level-id=24;object=23;bitrate=64000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:9 G722/8000
a=rtpmap:18 G729/8000
a= recvonly
```

ACK to MediaSense.

```
 06053719.001 |08:39:52.842 |AppInfo  |SIPTcp - wait_SdlSPISignal: Outgoing SIP TCP message to
10.106.122.178 on port 5060 index 71
[50183,NET]
ACK sip:7878@10.106.122.178:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK147605d100d
From: <sip:4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
To: <sip:7878@10.106.122.178>;tag=ds606d34cb
Date: Thu, 16 Jul 2015 15:39:52 GMT
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
User-Agent: Cisco-CUCM10.5
Max-Forwards: 70
CSeq: 101 ACK
Allow-Events: presence, kpml
Content-Type: application/sdp
Content-Length: 260

v=0
o=CiscoSystemsCCM-SIP 16791 1 IN IP4 10.106.122.174
s=SIP Call
c=IN IP4 10.106.122.131
b=TIAS:64000
b=CT:64
b=AS:64
t=0 0
m=audio 4000 RTP/AVP 9 101
a=ptime:20
a=rtpmap:9 G722/8000
a= sendonly
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Same process is repeated for far end stream. CUCM dials in BiB, BiB will dial the recording number and an SIP session will be established between CUCM and MediaSense.

As shown in this image, the Signaling diagram.



MediaSense Log Analysis

INVITE from CUCM to establish call-recording for near end (Audio from SIP IP Phone)

```
0000010803: 10.106.122.178: Jul 16 2015 08:39:52.694 -0700: %CCBU_CALL_CONTROL-6-BORDER_MESSAGE:
{Thrd=Pool-sip-thread-25} %[message_string=process new Invitation: SipCall-25,
INBOUND_RECORDING, null, State=ALERTED: , processing=1
INVITE sip:7878@10.106.122.178:5060 SIP/2.0
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK14432e0a687
Max-Forwards: 69
To: <sip:7878@10.106.122.178>
From: <sip:4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
CSeq: 101 INVITE
Content-Length: 0
Date: Thu, 16 Jul 2015 15:39:52 GMT
Supported: timer,resource-priority,replaces
Supported: X-cisco-srtp-fallback
Supported: Geolocation
Min-SE: 1800
User-Agent: Cisco-CUCM10.5
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY
Expires: 180
Allow-Events: presence, kpml
Call-Info: <sip:10.106.122.174:5060>;method="NOTIFY;Event=telephone-event;Duration=500"
Cisco-Guid: 3841694080-0000065536-0000000071-2927258122
Session-Expires: 1800
P-Asserted-Identity: <sip:4009@10.106.122.174>
Remote-Party-ID: <sip:4009@10.106.122.174>;party=calling;screen=yes;privacy=off
Contact: <sip:4009@10.106.122.174:5060;transport=tcp>;isfocus
 ]: Border Message
0000010804: 10.106.122.178: Jul 16 2015 08:39:52.694 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-25} -preProcessInvitation SipCall-25, INBOUND_RECORDING, null,
State=ALERTED: ciscoGuidHeader=Cisco-Guid: 3841694080-0000065536-0000000071-2927258122

0000010808: 10.106.122.178: Jul 16 2015 08:39:52.695 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-25} -postProcessInvitation SipCall-25, INBOUND_RECORDING, NEAR_END,
State=ALERTED: from=4009, displayName=null, xRefci=32833194, endPointType=NEAR_END,
xNearDevice=SEP1C17D341FD21, ucmCiscoGuid=null, nearEndClusterId=StandAloneCluster, and
farEndClusterId=StandAloneCluster


0000010809: 10.106.122.178: Jul 16 2015 08:39:52.695 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-25} -postProcessInvitation SipCall-25, INBOUND_RECORDING, NEAR_END,
State=ALERTED: created MediaResources: [AUDIO-MediaResource-25: SipCall-25, INBOUND_RECORDING,
NEAR_END, State=ALERTED, weight=1, ip= 10.106.122.174]
```

INVITE from CUCM to establish call-recording for far end (Audio from SCCP IP Phone).

```
 0000010818: 10.106.122.178: Jul 16 2015 08:39:52.700 -0700: %CCBU_CALL_CONTROL-6-
BORDER_MESSAGE: {Thrd=Pool-sip-thread-26} %[message_string=process new Invitation: SipCall-26,
INBOUND_RECORDING, null, State=ALERTED: , processing=2
INVITE sip:7878@10.106.122.178:5060 SIP/2.0
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK14578497f79
Max-Forwards: 69
To: <sip:7878@10.106.122.178>
From: <sip:4009@10.106.122.174;x-farend;x-refci=32833194;x-nearendclusterid=StandAloneCluster;x-
nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-farendrefci=32833193;x-
farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
```

```
farendaddr=4011>;tag=16792~78868996-a8aa-4784-b765-86098b176d95-32833201
Call-ID: e4fb9980-5a71d048-b1-ae7a6a0a@10.106.122.174
CSeq: 101 INVITE
Content-Length: 0
Date: Thu, 16 Jul 2015 15:39:52 GMT
Supported: timer,resource-priority,replaces
Supported: X-cisco-srtp-fallback
Supported: Geolocation
Min-SE: 1800
User-Agent: Cisco-CUCM10.5
Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY
Expires: 180
Allow-Events: presence, kpml
Call-Info: <sip:10.106.122.174:5060>;method="NOTIFY;Event=telephone-event;Duration=500"
Cisco-Guid: 3841694080-0000065536-0000000072-2927258122
Session-Expires: 1800
P-Asserted-Identity: <sip:4009@10.106.122.174>
Remote-Party-ID: <sip:4009@10.106.122.174>;party=calling;screen=yes;privacy=off
Contact: <sip:4009@10.106.122.174:5060;transport=tcp>;isfocus
 ]: Border Message
0000010819: 10.106.122.178: Jul 16 2015 08:39:52.700 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-26} -preProcessInvitation SipCall-26, INBOUND_RECORDING, null,
State=ALERTED: ciscoGuidHeader=Cisco-Guid: 3841694080-0000065536-0000000072-2927258122

0000010823: 10.106.122.178: Jul 16 2015 08:39:52.701 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-26} -postProcessInvitation SipCall-26, INBOUND_RECORDING, NEAR_END,
State=ALERTED: from=4009, displayName=null, xRefci=32833194, endPointType=FAR_END,
xNearDevice=null, ucmCiscoGuid=null, nearEndClusterId=StandAloneCluster, and
farEndClusterId=StandAloneCluster

0000010824: 10.106.122.178: Jul 16 2015 08:39:52.701 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-26} -postProcessInvitation SipCall-26, INBOUND_RECORDING, NEAR_END,
State=ALERTED: created MediaResources: [AUDIO-MediaResource-26: SipCall-26, INBOUND_RECORDING,
FAR_END, State=ALERTED, weight=1, ip= 10.106.122.174
```

Session ID created for the call once both SIP leg for Near End and Far End recording information is captured on MediaSense.

```
 0000010830: 10.106.122.178: Jul 16 2015 08:39:52.703 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=Pool-sip-thread-26} -Core: dispatch StartRecordingRequestEvent: SipRequestContextImpl-76,
type=Sip, Session:  d14e97859bff1, INITIALIZING, call=SipCall-26, INBOUND_RECORDING, FAR_END,
State=ALERTED, firstCall=SipCall-25, INBOUND_RECORDING, NEAR_END, State=ALERTED,
requestedAudioPorts=2, requestedVideoPorts=0, append=false, audioSdp=null to Recording Adapter
```

200 OK and ACK for near end call.

```
 0000010846: 10.106.122.178: Jul 16 2015 08:39:52.829 -0700: %CCBU_CALL_CONTROL-6-
BORDER_MESSAGE: {Thrd=Pool-capture-thread-38} %[message_string=SipCall-25, INBOUND_RECORDING,
NEAR_END, State=ALERTED send 200 Ok:
SIP/2.0 200 Ok
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK14432e0a687
To: <sip:7878@10.106.122.178>;tag=ds606d34cb
From: <sip:4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
CSeq: 101 INVITE
Content-Length: 313
Contact: <sip:7878@10.106.122.178:5060;transport=tcp>
Content-Type: application/sdp
Allow: INVITE, BYE, CANCEL, ACK, NOTIFY, INFO, UPDATE
Server: MediaSense/10.x
```

```
v=0
o=CiscoORA 3197 1 IN IP4 10.106.122.178
s=SIP Call
c=IN IP4 10.106.122.178
t=0 0
m=audio 42120 RTP/AVP 102 0 8 9 18
a=rtpmap:102 MP4A-LATM/90000
a=fmtp:102 profile-level-id=24;object=23;bitrate=64000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:9 G722/8000
a=rtpmap:18 G729/8000
a= recvonly
 ACK sip:7878@10.106.122.178:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 10.106.122.174:5060;branch=z9hG4bK147605d100d
Max-Forwards: 69
To: <sip:7878@10.106.122.178>;tag=ds606d34cb
From: <sip:4009@10.106.122.174;x-nearend;x-refci=32833194;x-
nearendclusterid=StandAloneCluster;x-nearenddevice=SEP1C17D341FD21;x-nearendaddr=4009;x-
farendrefci=32833193;x-farendclusterid=StandAloneCluster;x-farenddevice=SEP203A0782D99F;x-
farendaddr=4011>;tag=16791~78868996-a8aa-4784-b765-86098b176d95-32833198
Call-ID: e4fb9980-5a71d048-b0-ae7a6a0a@10.106.122.174
CSeq: 101 ACK
Content-Length: 260
Date: Thu, 16 Jul 2015 15:39:52 GMT
User-Agent: Cisco-CUCM10.5
Allow-Events: presence, kpml
Content-Type: application/sdp

v=0
o=CiscoSystemsCCM-SIP 16791 1 IN IP4 10.106.122.174
s=SIP Call
c=IN IP4 10.106.122.131
b=TIAS:64000
b=CT:64
b=AS:64
t=0 0
m=audio 4000 RTP/AVP 9 101
a=ptime:20
a=rtpmap:9 G722/8000
a= sendonly
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Similar event will be captured once the Media Sense answers the call. Note that ACK sent contains port 4000 and indicates **sendonly**.

Session information after both SIP dialog established.


```
 {"sessionData": {
"callControllerIP": "10.106.122.174",
"callControllerType": "Cisco-CUCM",
"endPoints": [
{
"clusterid": "StandAloneCluster",
"conference": false,
"device": " SEP1C17D341FD21",
"dn": " 4009",
"startDate": 1437061192882,
"tracks": [{
"codec": " G722",
"location": "/common",
```

```
"mediaState": "ACTIVE",
"startDate": 1437061192882,
"track": 0,
"type": "AUDIO"
}],
"type": "NEAR_END",
"xRefci": "32833194"
},
{
"clusterid": "StandAloneCluster",
"conference": false,
"device": "SEP203A0782D99F",
"dn": "4011",
"startDate": 1437061192882,
"tracks": [{
"codec": "G722",
"location": "/common",
"mediaState": "ACTIVE",
"startDate": 1437061192882,
"track": 1,
"type": "AUDIO"
}],
"type": "FAR_END",
"xRefci": "32833193"
}
],
"operationType": "ADD",
"recordingServer": "10.106.122.178",
"rtspUrl": "rtsp://10.106.122.178/d14e97859bff1",
"sessionName": "d14e97859bff1",
"sipServer": "10.106.122.178",
"startDate": 1437061192882,
"state": "ACTIVE",
"version": 7
```


When the call is disconnected phone stops recording.


```
 0000010897: 10.106.122.178: Jul 16 2015 08:40:01.525 -0700: %CCBU_CALL_CONTROL-7-TRACE:
{Thrd=DIALOG_CALLBACK.7} -Core: dispatch  StopRecordingRequestEvent: SipRequestContextImpl-78,
type=Sip, Session: d14e97859bff1, ACTIVE, call=SipCall-26, INBOUND_RECORDING, FAR_END,
State=DISCONNECTED, firstCall=null to Recording Adapter
0000009368: 10.106.122.178: Jul 16 2015 08:40:01.762 -0700: %CCBU_COMMON-6-VSMS HTTP Info:
{Thrd=Pool-capture-thread-39} %[HTTP Response Body=<Session>
<diskusage>
<recording name="d14e97859bff1-TRACK0"  size="1"  repository="/common" />
<recording name="d14e97859bff1-TRACK1"  size="1"  repository="/common" />
</diskusage>
<rtsplink>/archive/d14e97859bff1</rtsplink>
```


> **Note**: In this area, you notice that there is a size in the recording attributes. This example shows that **size="1"**, which means MediaSense did receive the audio from CUCM. If you notice **size="0"**, it means MediaSense did not receive the audio from CUCM.

Finally the session closes.


```
 {"sessionData": {
"callControllerIP": "10.106.122.174",
"callControllerType": "Cisco-CUCM",
```

```
"endDate": 1437061201522,
"endPoints": [
{
"clusterid": "StandAloneCluster",
"conference": false,
"device": " SEP1C17D341FD21",
"dn": " 4009",
"startDate": 1437061192882,
"tracks": [{
"codec": "G722",
"location": "/common",
"mediaState": "ACTIVE",
"size": 1,
"startDate": 1437061192882,
"track": 0,
"type": "AUDIO"
}],
"type": " NEAR_END",
"xRefci": "32833194"
},
{
"clusterid": "StandAloneCluster",
"conference": false,
"device": " SEP203A0782D99F",
"dn": " 4011",
"startDate": 1437061192882,
"tracks": [{
"codec": "G722",
"location": "/common",
"mediaState": "ACTIVE",
"size": 1,
"startDate": 1437061192882,
"track": 1,
"type": "AUDIO"
}],
"type": " FAR_END",
"xRefci": "32833193"
}
],
"operationType": "EXISTING",
"recordingServer": "10.106.122.178",
"rtspUrl": "rtsp://10.106.122.178/archive/d14e97859bff1",
"sessionName": " d14e97859bff1",
"sipServer": "10.106.122.178",
"startDate": 1437061192882,
"state": " CLOSED",
"version": 11
```


Log collection from MediaSense

Step 1.

As shown in this image, MediaSense Serviceabilty.

Step 2.

Please run **utils network capture eth0 file packets count 100000 size all** in order to enable packet capture on MediaSense.

As shown in this image, Packet capture on MediaSense.

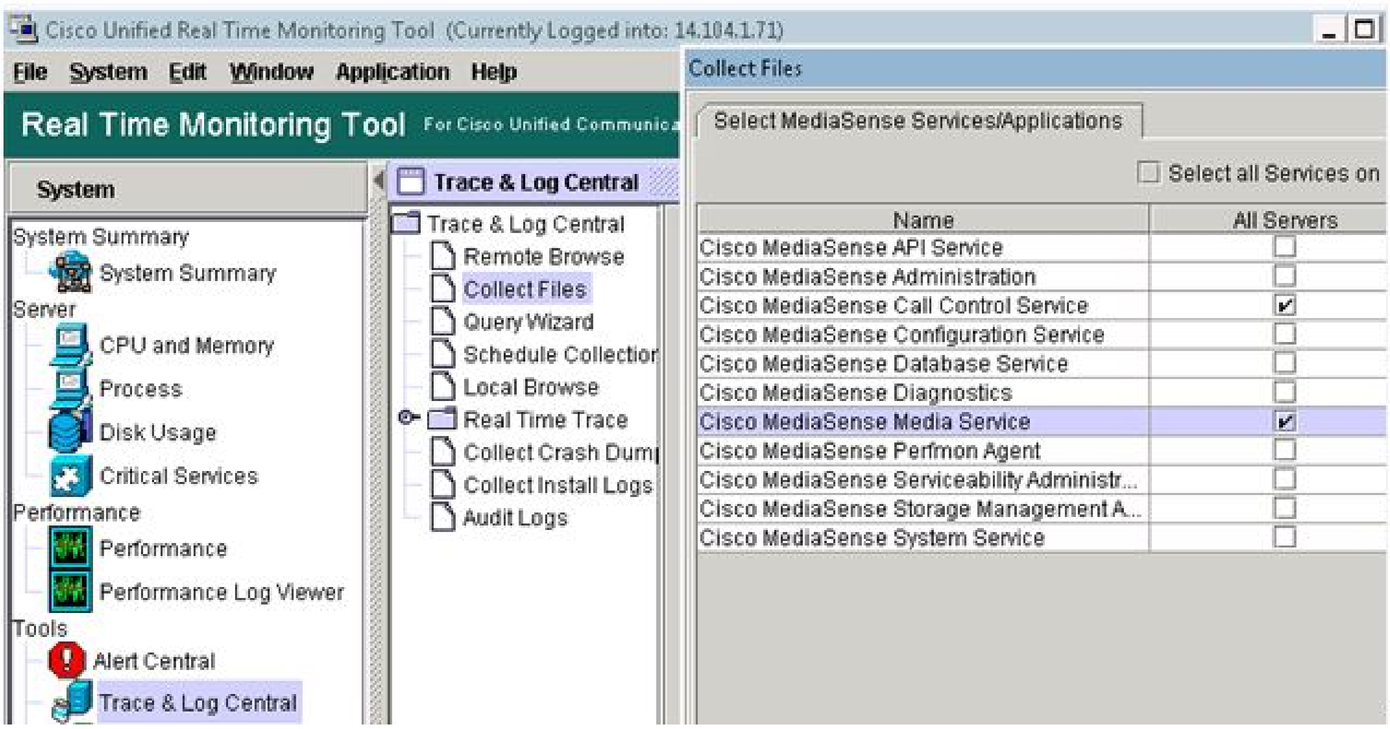
```
admin:utils network capture eth0 file packets count 100000 size all
Executing command with options:
 size=ALL                 count=100000             interface=eth0
 src=                     dest=                    port=
 ip=

Control-C pressed

admin:
```

Step 3.

Connect to MediaSense server using RTMT.

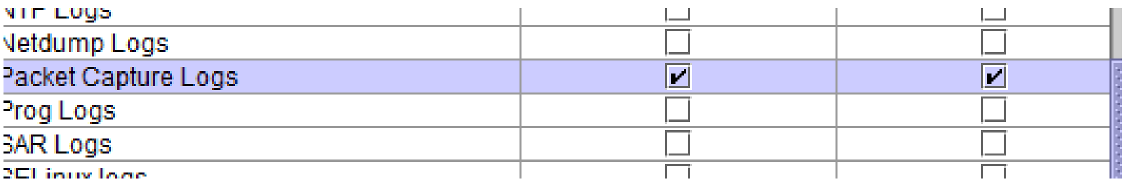
Navigate to **Trace & Log Central** > **Collect Files**

As shown in this image, Real Time Monitoring Tool.



Click **Next** and select **packet capture**

As shown in this image, Real Time Monitoring Tool.



Select time accordingly.

Some useful commands:

1. **utils media recording_sessions**

The **utils media recording_sessions** file fileName command generates an html file with a detailed list of the last 100 recording sessions processed by this Cisco MediaSense server. Confirm that the Cisco MediaSense Call Control Service is running before you execute this command. The file is saved to the platform/cli/ folder and can be downloaded using the file get activelog platform/cli/fileName command.

Command: **utils media recording_sessions** file fileName

Details:

- **file** is a mandatory parameter that outputs the information to a file.
- **fileName** is a mandatory parameter that defines the name of the .html file.
- When you issue this command, you get the following response: Cisco MediaSense Call Control Service Recording sessions saved to platform/cli/<filename>.html. You can now download it using: file get activelog platform/cli/<filename>.html   You can then retrieve the file from that directory and save it to a location of your choice.

Example:

- **utils media recording_sessions** file sessions.html Cisco MediaSense. Call Control Service Recording sessions saved to platform/cli/sessions.html. You can now download it using: file get activelog platform/cli/sessions.html

2. **utils system maintenance**

The command **utils system maintenance** operation enables or disables maintenance mode on Cisco MediaSense , or displays the Cisco MediaSense maintenance mode status. While it is in maintenance mode, Cisco MediaSense cannot process any recording requests or API requests.

Cisco MediaSense reboots when it enters maintenance mode. Any streaming activities end abruptly. Any active recordings end in a CLOSED_ERROR state. Cisco MediaSense reboots again when maintenance mode is disabled and it re-enters normal mode.

Command: **utils system maintenance** operation

Details: operation specifies what the command does.

Valid operations include:

- enable
- disable
- status

Examples:

- utils system maintenance enable
- utils system maintenance disable
- utils system maintenance status

Some basic issues

MediaSense Doc Wiki

Known Defects

CSCup24364 : C all recording not working for calls with no caller id get error message.

CSCui13760 : MediaSense does not support removal of node from cluster.

CSCtn45420 : MediaSense call recording fails with Camelot SIP endpoint.

CSCut09446 : MediaSense UI does not populate CUCM configuration & API user config.

CSCuo95309 : MediaSense Search and Play Recordings not populated from other node.

CSCuq20108 : From header to getting truncated when using escaped characters.