Configure Cisco Meeting Server (CMS) Version 3 Next Generation Streamer And Uploader

Contents

Introduction

This document describes the steps to configure and troubleshoot Cisco Meeting Server (CMS) integration with Next Generation Streamer and Uploader. The Next Generation Streamer was introduced from CMS version 3.0 and is Session Initiation Protocol(SIP)-Based.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- CMS Callbridge(s) Version 3.0 or later with Recording/Streaming license(s). (one recording license will allow one streaming call)
- Vbrick Distributed Media Engine (DME) (used for publishing the live stream from CMS Streaming service)
- Vbrick Rev (optional: only required if Live Streaming need to be shared outside internal network or multicast
- Network File System (NFS) directory is needed, and it can be setup on Windows Server or Linux.
- For Windows server, follow the steps toDeploy Network File system (NFS) on Windows
- For Linux, follow the steps toDeploy Network File systemon Linux

Components Used

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

- CMS 3.2 and above with 'Recorder' and/or 'Streamer' license(s). (recorder license will also allow you to stream)
- VBrick Distributed Media Engine (DME) 3.15.0 Rhel7

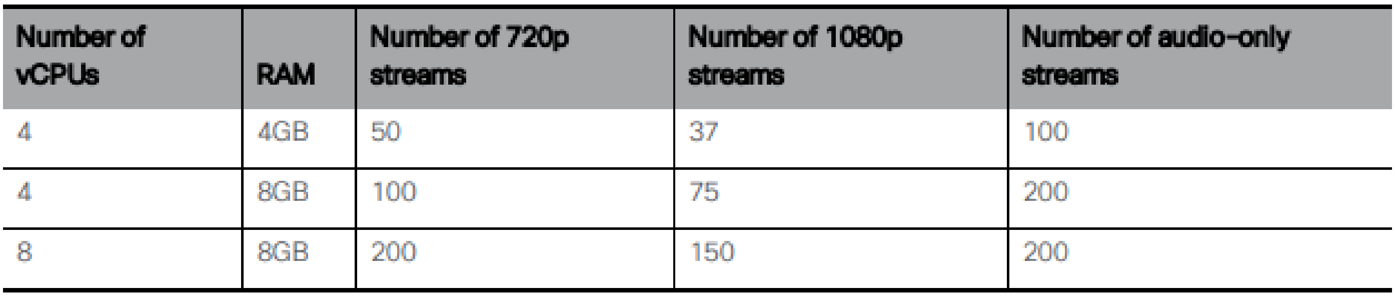
- Vbrick Rev (Uploader is used with vBrick Rev server. No manual importing of recordings is required)
- Windows Server 2012 R2 with NFS

Background Information

CMS Version 2.1 and later introduced support for live streaming with the CMS streamer using standard Real-Time Messaging Protocol (RTMP). In CMS 3.0, the Next Generation Streamer was introduced which is a SIP-based component. Prior version than 3.0 used Extensible Messaging and Presence Protocol (XMPP) . CMS version 3.1 and above support RTMPS protocol and thus communcation between the CMS streamer component and external server can be encrypted. This allows for the CMS streamer to integrate with any streaming platform that supports RTMP(S) (Youtube, Facebook, Wowza, and so on). Currently the CMS Streamer has been tested with Vbrick DME as an external streaming server and is the recommended platform for integration.

Live Streaming (Webcast) integration with VBrick DME allow users to watch any live streamed CMS conference anywhere inside the network from different devices. Additionally, when VBrick Rev is used along VBrick DME, this extends this capability for viewing from outside the internal network for every VBrick Rev authorized user.Also, CMS Uploader component simplifies the work flow for uploading Meeting Server recordings to the video content manager, Vbrick, from a configured NFS connected to a Meeting Server. No manual importing of recordings is required. Once the Uploader component is configured and enabled, recordings are pushed from the NFS to Vbrick.

Streamer



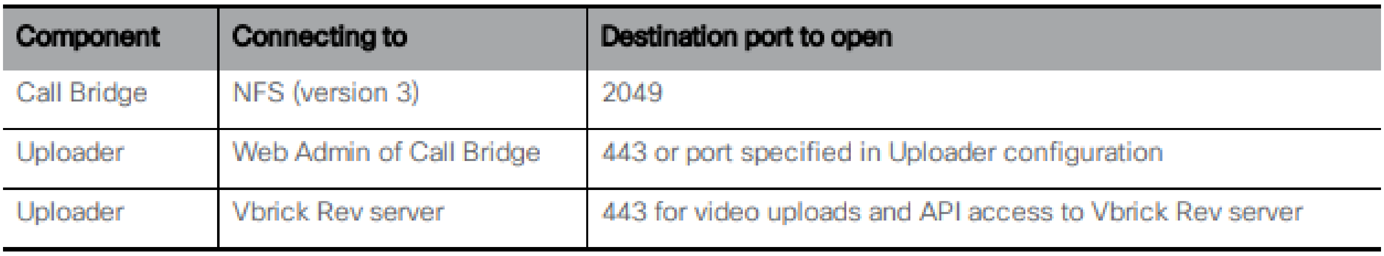
Number of vCPUs	RAM	Number of 720p streams	Number of 1080p streams	Number of audio-only streams
4	4GB	50	37	100
4	8GB	100	75	200
8	8GB	200	150	200

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

Uploader

The Uploader component can be installed on the same server as the Recorder component, or on a separate server. If installed on the same server as the Recorder, then add a couple of vCPUs for it to use. If run on a different server, then use the same server specification as for the Recorder: dedicated VM with a minimum of 4 physical cores and 4GB of RAM.

The Meeting Server running the Uploader will require Read and Write permissions for the Network File Sharing (NFS). The Uploader must run on a different Meeting Server and not on Call Bridge hosting the conferences.

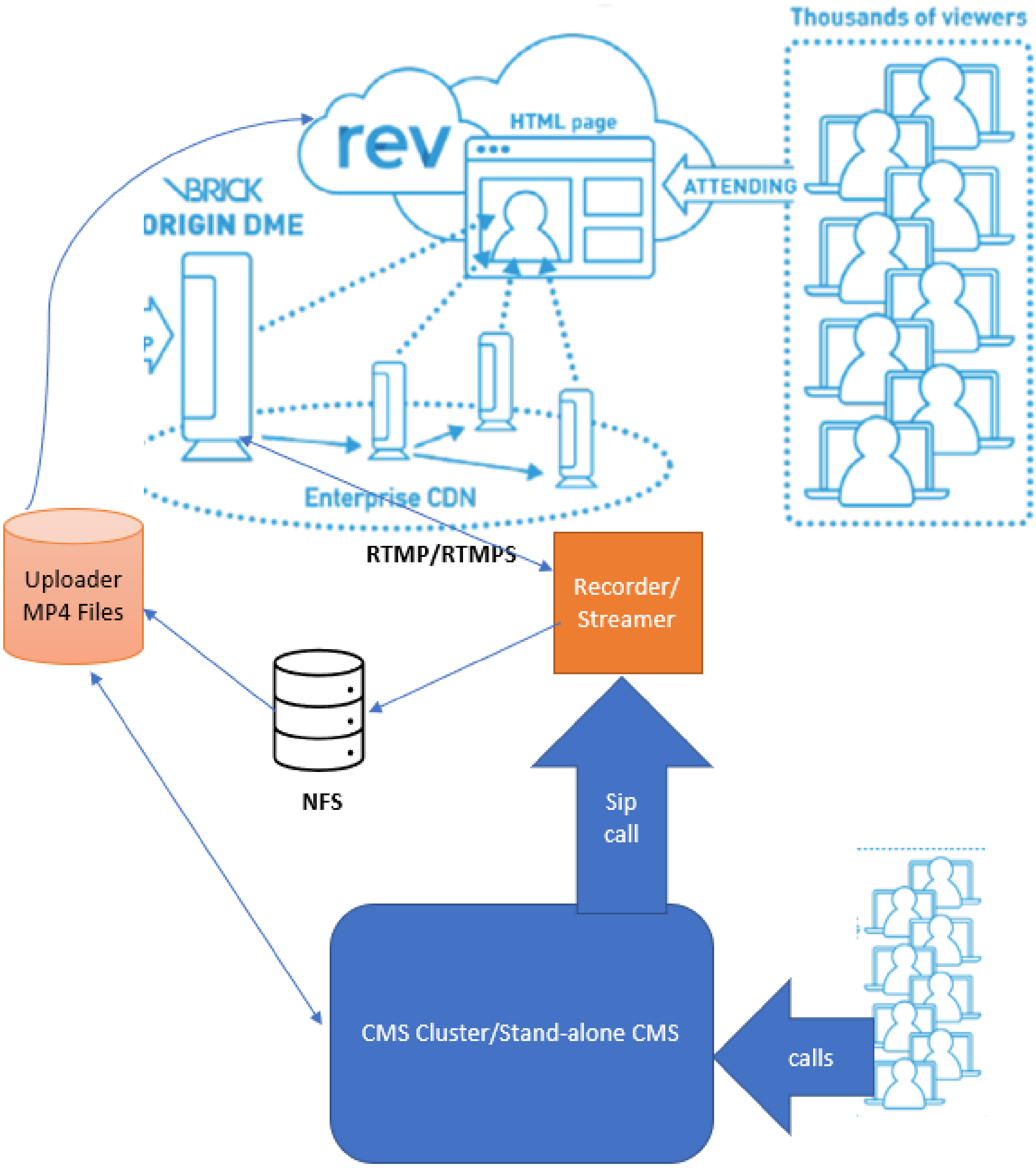


Component	Connecting to	Destination port to open
Call Bridge	NFS (version 3)	2049
Uploader	Web Admin of Call Bridge	443 or port specified in Uploader configuration
Uploader	Vbrick Rev server	443 for video uploads and API access to Vbrick Rev server

Configure

Network Diagram

There are several scenarios supported to deploy Streamer and Uploader with CMS such as: single callbridge with multiple streaming servers, a callbridge cluster with a single streaming server and callbridge cluster with multiple streaming servers. This document is based on a basic deployment with a cluster of callbridge connecting to a single streamer, Uploader server as all the configuration steps with this scenario apply to other scenarios as well.



As shown in the above image

CMS CallBridge Cluster

CMS Streamer/Recorder

CMS Uploader

VBrick DME For Streaming

VBrick Rev For Streaming

Configurations

Streamer

It is assumed that the callbridge is already setup and accepting calls.

Step 1. Certificates

The new streamer components do not require to listen https connections, however, it listen to SIP connections , the streamer server must have a valid certificate for TLS communication.

```
streamer> pki csr tac CN:.*.tptac9.com subjectAltName:streamer.tptac9.com .............. ......
Created key file tac.key and CSR tac.csr CSR file tac.csr ready for download via SFTP
```

Get the certificate signed from local Certificate Authority (CA). Upload the generate certificate using Secure File Transfer Protocol (SFTP) to the Streamer server. Checked to confirm the certificates uploaded successfully.

In this document wildcard certificates are used for streamer. Please use the certificate guide for reference.

https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Deployment_Guide/Version-3-1/Certificate-Guidelines-for-all-Deployments-3-1.pdf

```
streamer> pki list User supplied certificates and keys: tac.key tac.cer ROOTCA.cer example.key
example.csr tac.csr
```

Step 2. MMP/SSH Configuration

- Configure the listening interface of the streamer and the SIP TCP and TLS ports to listen on using the MMP command
  streamer sip listen <interface> <tcp-port|none> <tls-port|none>

  streamer> streamer sip listen a 7000 7001

  To use only TLS sip connection. Configure TCP sip connection as "none" the command is below

  streamer> streamer sip listen a none 7001

- Apply certificates for the streamer server

  streamer> streamer sip certs tac.key tac.cer

- Select the quality for streaming
  streamer> streamer sip resolution 720p

- Enable streamer

streamer> streamer enable

- Optionally, if TLS is configured, you can perform TLS verification for SIP on the streamer

streamer> tls sip trust ROOTCA.cer

> **Note**: For the TLS connection to be secure we recommend enabling TLS verification.
> streamer> tls sip verify enable

- Verfiy the configuration information entered above is correct

```
streamer> streamer Enabled : false SIP interfaces : tcp a:7000, tls a:7001 SIP key file :
tac.key SIP certificate file : tac.cer SIP CA Bundle file : none SIP Resolution : 720p SIP
traffic trace : Disabled Call Limit : none
```

- Enable the streamer by using command: "**streamer enable**". All message must show "SUCCESS" as below

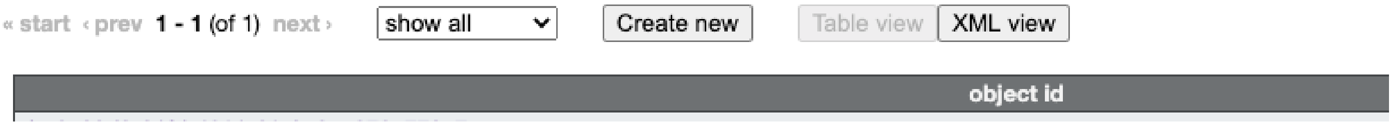
```
streamer> streamer enable SUCCESS: Key and certificate pair match SUCCESS: Streamer enabled
```

Step 3. API configuration

This configuration is performed in the CMS hosting CallBridge. Below, API on Webadmin interface of CMS. Any REST client like Postman, Poster can also be used to perform this.
Once the new SIP streamer is enabled, it can be configured and used in the Call Bridge under /callProfiles for sipStreamerUri.

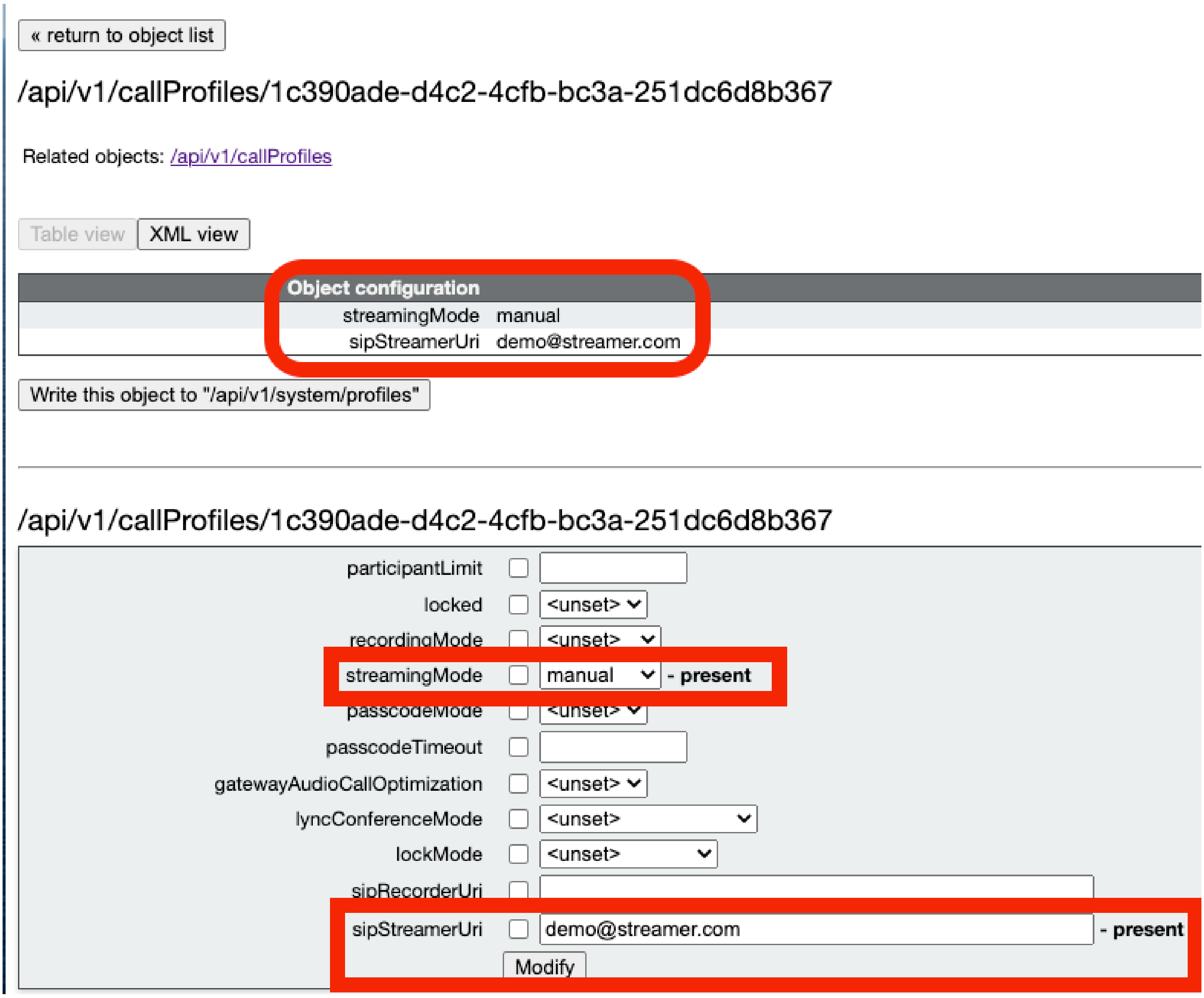
To use API on CMS server. Go To Webadmin > Configuration > API

- Create Call Profile



- Configure sipStreamuri=demo@streamer.com & StreamingMode= Automatic/Manual

The user part of the configured "sipStreamuri" (i.e. the part before '@' symbol) has no significant meaning, and for the new SIP streamer component, although required, it can usually be anything, e.g. "streamer@streamer.com".The important part of the URI is the "domain" part.



/api/v1/callProfiles/1c390ade-d4c2-4cfb-bc3a-251dc6d8b367

Related objects: /api/v1/callProfiles

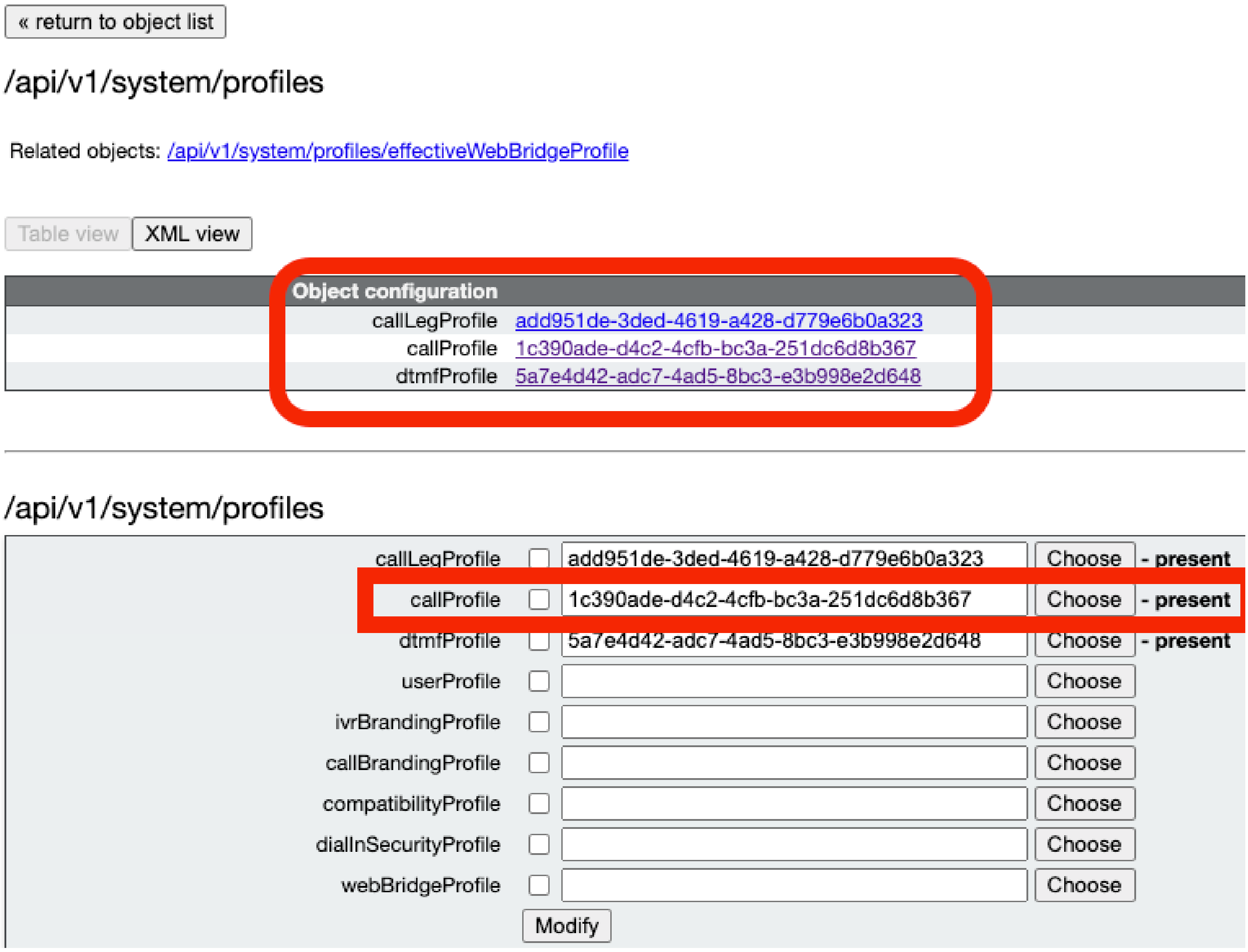
Object configuration	
streamingMode	manual
sipStreamerUri	demo@streamer.com

/api/v1/callProfiles/1c390ade-d4c2-4cfb-bc3a-251dc6d8b367

participantLimit	□		
locked	□	<unset>	
recordingMode	□	<unset>	
streamingMode	□	manual	- **present**
passcodeMode	□	<unset>	
passcodeTimeout	□		
gatewayAudioCallOptimization	□	<unset>	
lyncConferenceMode	□	<unset>	
lockMode	□	<unset>	
sipRecorderUri	□		
sipStreamerUri	□	demo@streamer.com	- **present**

Configure sipStreamuri=demo@streamer.com & StreamingMode= Automatic/Manual

- Add the created streamer callProfile above to the /system/profiles. This is a global configuration and the configured "sipStreamerUri" will be used for streamer operation.



/api/v1/system/profiles

Related objects: /api/v1/system/profiles/effectiveWebBridgeProfile

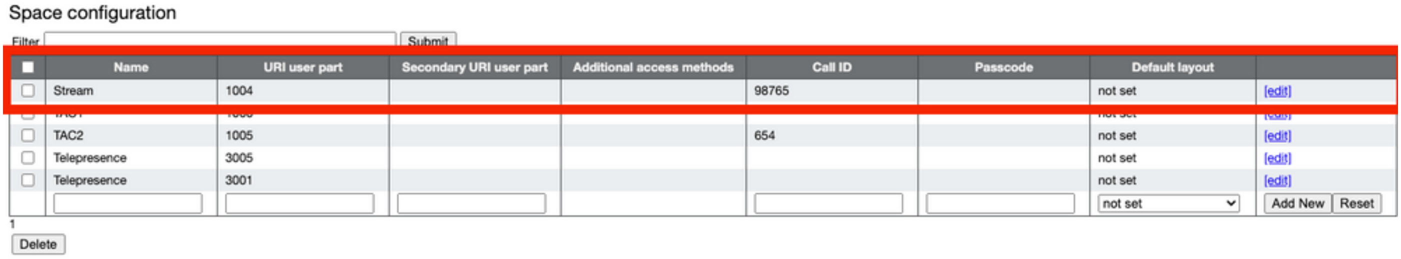
Object configuration	
callLegProfile	add951de-3ded-4619-a428-d779e6b0a323
callProfile	1c390ade-d4c2-4cfb-bc3a-251dc6d8b367
dtmfProfile	5a7e4d42-adc7-4ad5-8bc3-e3b998e2d648

/api/v1/system/profiles

callLegProfile	□	add951de-3ded-4619-a428-d779e6b0a323	Choose	- present
callProfile	□	1c390ade-d4c2-4cfb-bc3a-251dc6d8b367	Choose	- present
dtmfProfile	□	5a7e4d42-adc7-4ad5-8bc3-e3b998e2d648	Choose	- present
userProfile	□		Choose	
ivrBrandingProfile	□		Choose	
callBrandingProfile	□		Choose	
compatibilityProfile	□		Choose	
dialInSecurityProfile	□		Choose	
webBridgeProfile	□		Choose	

Add the callProfile to system/profiles

- Add the VBrick 'streamURL' to the space(s) used for streaming. For reference, a space called 'Stream' was created using the CMS web interface



Space configuration

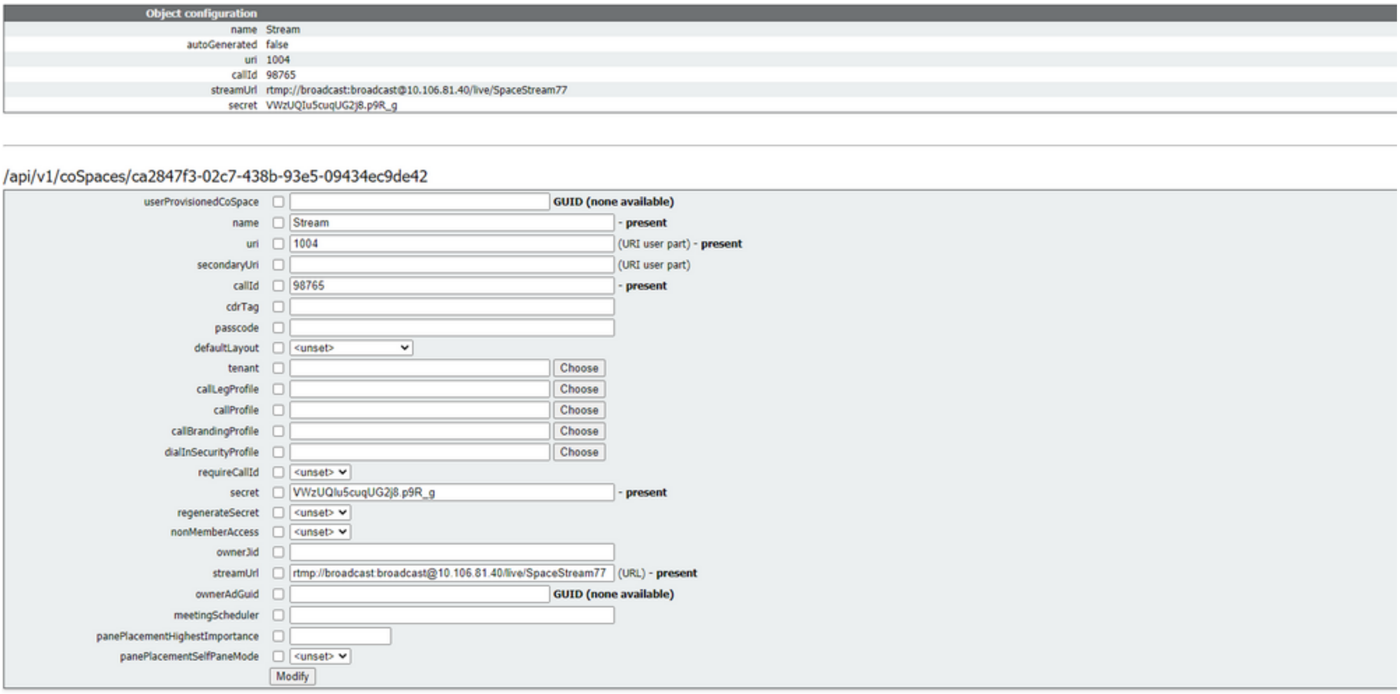
■	Name	URI user part	Secondary URI user part	Additional access methods	Call ID	Passcode	Default layout	
□	Stream	1004			98765		not set	[edit]
□	[illegible]	[illegible]					[illegible]	[edit]
□	TAC2	1005			654		not set	[edit]
□	Telepresence	3005					not set	[edit]
□	Telepresence	3001					not set	[edit]
							not set	Add New Reset

1

Space used for Streaming

- Modify the space to add “StreamURL”. The 'streamURL' in the following format:

  rtmp://<VBrickBroadcastUsername>:<VBrickBroadcastPassword>@<VBrick IP or FQDN>/live/NameoftheStream. In example I configured it as “rtmp://broadcast:broadcast@10.106.81.40/live/SpaceStream77”



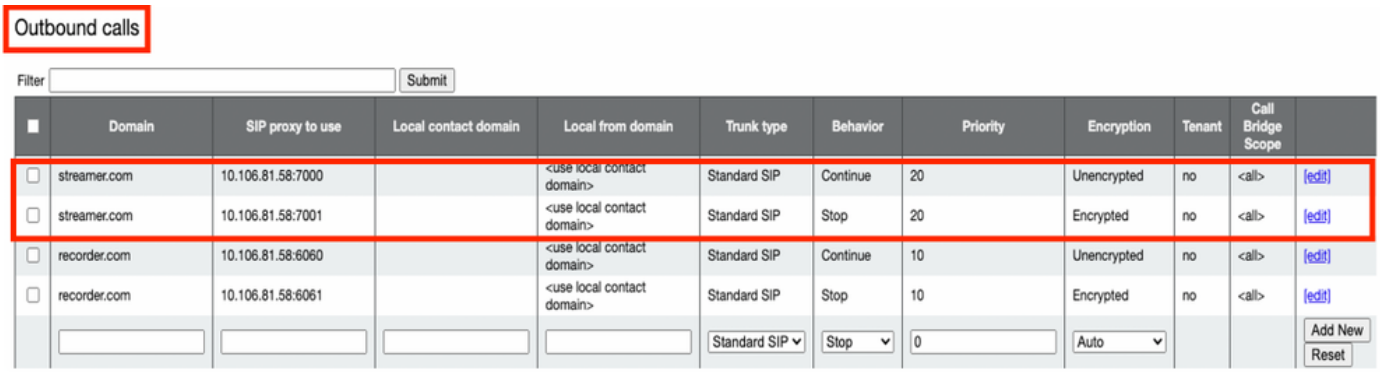
Object configuration	
name	Stream
autoGenerated	false
uri	1004
callId	98765
streamUrl	rtmp://broadcast:broadcast@10.106.81.40/live/SpaceStream77
secret	VWzUQIu5cuqUG2j8.p9R_g

/api/v1/coSpaces/ca2847f3-02c7-438b-93e5-09434ec9de42

userProvisionedCoSpace	□		GUID (none available)
name	□	Stream	- present
uri	□	1004	(URI user part) - present
secondaryUri	□		(URI user part)
callId	□	98765	- present
cdrTag	□		
passcode	□		
defaultLayout	□	<unset>	
tenant	□		Choose
callLegProfile	□		Choose
callProfile	□		Choose
callBrandingProfile	□		Choose
dialInSecurityProfile	□		Choose
requireCallId	□	<unset>	
secret	□	VWzUQIu5cuqUG2j8.p9R_g	- present
regenerateSecret	□	<unset>	
nonMemberAccess	□	<unset>	
ownerJid	□		
streamUrl	□	rtmp://broadcast:broadcast@10.106.81.40/live/SpaceStream77	(URL) - present
ownerAdGuid	□		GUID (none available)
meetingScheduler	□		
panePlacementHighestImportance	□		
panePlacementSelfPaneMode	□	<unset>	
		Modify	

Step 4. Create "Outbound Rules"

Configure, a custom URI that maps to an outboundDialPlan rule (the domain can be anything for example "streamer.com"). Please configure an outboundDialPlan rule to match the domain used in streamerUri to route.



Outbound calls

Filter [ ] Submit

■	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	Call Bridge Scope	
□	streamer.com	10.106.81.58:7000		<use local contact domain>	Standard SIP	Continue	20	Unencrypted	no	<all>	[edit]
□	streamer.com	10.106.81.58:7001		<use local contact domain>	Standard SIP	Stop	20	Encrypted	no	<all>	[edit]
□	recorder.com	10.106.81.58:6060		<use local contact domain>	Standard SIP	Continue	10	Unencrypted	no	<all>	[edit]
□	recorder.com	10.106.81.58:6061		<use local contact domain>	Standard SIP	Stop	10	Encrypted	no	<all>	[edit]
					Standard SIP	Stop	0	Auto			Add New Reset

Create outbound rules

As shown in the above image, for SIP streamer, if default ports for SIP (5060,5061) are not used, then It is mandatory to specify ports in the configuration of the streamer and include the following port number to connect to the "**sip proxy to use**" field when outboundDialPlanRule is configured for the service.

```
streamer> streamer Enabled : true SIP interfaces : tcp a:7000, tls a:7001 SIP key file : tac.key
SIP certificate file : tac.cer SIP CA Bundle file : none SIP Resolution : 720p SIP traffic trace
: Disabled Call Limit : none
```

Uploader

- Specify the NFS and directory where recordings will be stored that the Uploader will monitor
  streamer> uploader nfs 192.168.15.38:Recording

- Specify the Meeting Server that the Uploader will query for recording information
  streamer> uploader cms host join.mextp.local

- Specify the Web Admin port on the Meeting Server running the Call Bridge

```
streamer> uploader cms port 445
```

- Specify the user with API access on the Meeting Server running the Call Bridge

```
streamer> uploader cms user apiadmin
streamer> uploader cms password
Please enter password:
```

- Add the certificate bundle from CMS to the Meeting Server trust store

  Create a certificate bundle (crt-bundle) holding a copy of the Root CA's certificate and all intermediate certificates in the chain for the Web Admin on the Meeting Server running the Call Bridge.

```
streamer> uploader cms trust ROOTCA.cer
```

- Configure the Vbrick host and the port to which the Uploader will connect

```
streamer> uploader rev host ciscotac.rev-na.demo.vbrick.com
streamer> uploader rev port 443
```

  Note: The port defaults to 443 unless otherwise specified

- Add a Vbrick Rev user who has API permission to upload video recordings

```
streamer> uploader rev user tacuser
streamer> uploader rev password
Please enter password:
```

- Add the certificate bundle to the Vbrick Rev trust store

Create a certificate bundle (crt-bundle) holding a copy of the Root CA's certificate and all intermediate certificates in the chain for the Vbrick Rev serve

```
streamer> uploader rev trust vbrickbundle.cer
```

- Check uploader configuration and enable uploader

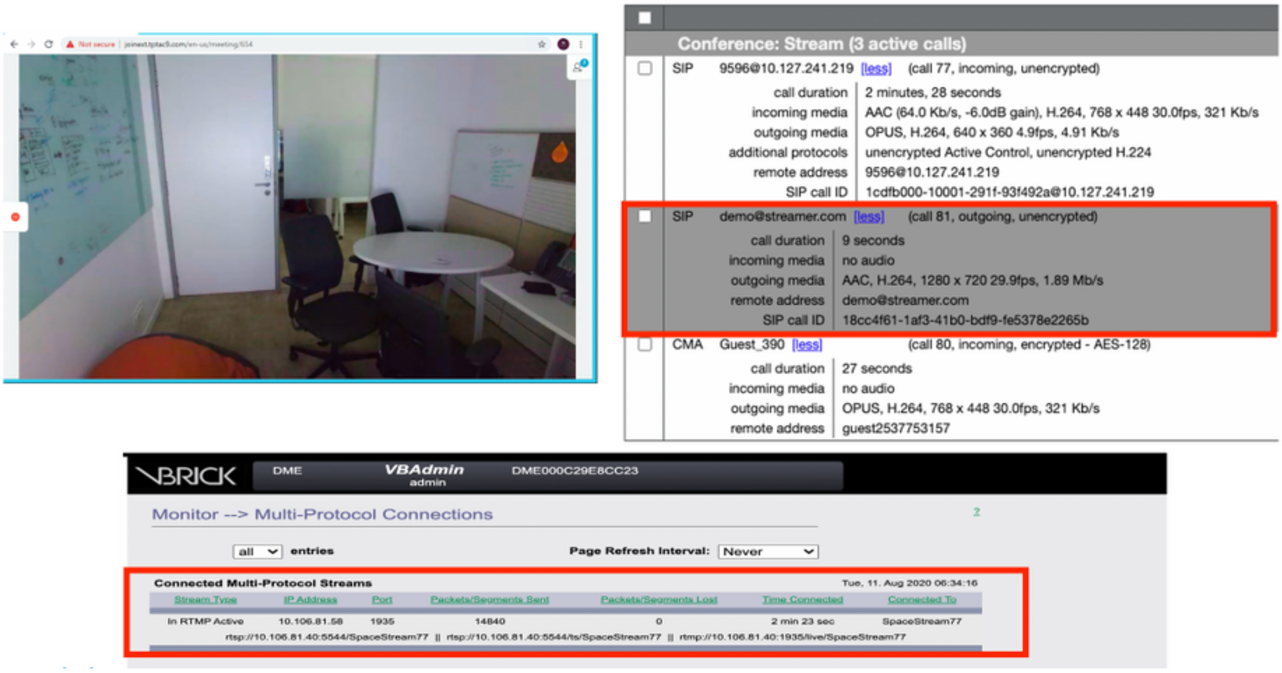
```
streamer> uploader Enabled : false NFS hostname : 192.168.15.38 NFS directory : Recording CMS host : join.mextp.local CMS port : 445 CMS user : apiadmin CMS trust bundle : ROOTCA.cer Vbrick Rev hostname : ciscotac.rev-na.demo.vbrick.com Vbrick Rev port : 443 Vbrick Rev username : tacuser Vbrick Rev trust bundle : brick.cer View access : Public cospace_member_access : edit recording_owned_by_cospace_owner : false fallback_owner : admin comments_enabled : true ratings_enabled : true downloads_enabled : true active_upon_upload : true delete_after_upload : false
```

If configuration is correct, use "**uploader enable**" command to enable the Uploader component. All message must show "SUCCESS" as shown below.

```
streamer> uploader enable SUCCESS: uploader enabled
```

Verify

Streamer



Conference: Stream (3 active calls)

□ SIP 9596@10.127.241.219 [less] (call 77, incoming, unencrypted)

call duration	2 minutes, 28 seconds
incoming media	AAC (64.0 Kb/s, -6.0dB gain), H.264, 768 x 448 30.0fps, 321 Kb/s
outgoing media	OPUS, H.264, 640 x 360 4.9fps, 4.91 Kb/s
additional protocols	unencrypted Active Control, unencrypted H.224
remote address	9596@10.127.241.219
SIP call ID	1cdfb000-10001-291f-93f492a@10.127.241.219

□ SIP demo@streamer.com [less] (call 81, outgoing, unencrypted)

call duration	9 seconds
incoming media	no audio
outgoing media	AAC, H.264, 1280 x 720 29.9fps, 1.89 Mb/s
remote address	demo@streamer.com
SIP call ID	18cc4f61-1af3-41b0-bdf9-fe5378e2265b

□ CMA Guest_390 [less] (call 80, incoming, encrypted - AES-128)

call duration	27 seconds
incoming media	no audio
outgoing media	OPUS, H.264, 768 x 448 30.0fps, 321 Kb/s
remote address	guest2537753157

Monitor --> Multi-Protocol Connections

all entries   Page Refresh Interval: Never

Connected Multi-Protocol Streams   Tue, 11. Aug 2020 06:34:16

Stream Type	IP Address	Port	Packets/Segments Sent	Packets/Segments Lost	Time Connected	Connected To
In RTMP Active	10.106.81.58	1935	14840	0	2 min 23 sec	SpaceStream77

rtsp://10.106.81.40:5544/SpaceStream77 || rtsp://10.106.81.40:5544/ts/SpaceStream77 || rtmp://10.106.81.40:1935/live/SpaceStream77

Streaming working and a sip streaming call connected

Uploader

You can see log for a successful event in syslog follow of uploader.

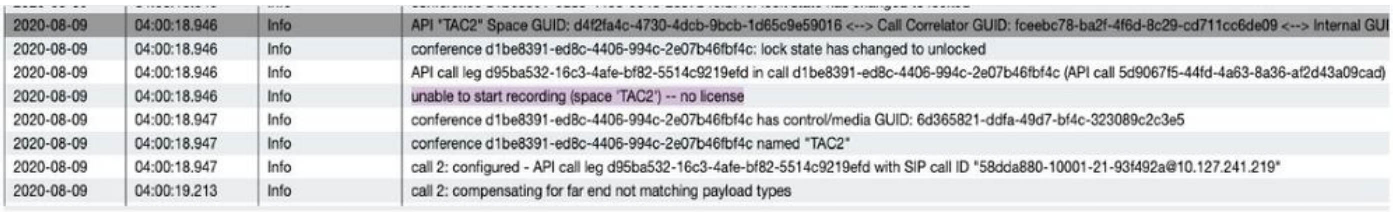
```
Jun 17 22:24:41.867 user.info cms-02 Uploader[1]: scanning directory:
/mnt/recordings/forwardedCalls Jun 17 22:24:41.867 user.info cms-02 Uploader[1]: scanning
directory: /mnt/recordings/spaces Jun 17 22:24:41.869 user.info cms-02 Uploader[1]: checking the
status of /mnt/recordings/spaces/8a7076e2-6db6-47e9-98ee-3bd063e32559/20210618032309+0000_vid-
id=c4605aaf-dc49-4cd7-9174-c46185ba1983@vbrick.mp4 Jun 17 22:24:41.870 user.info cms-02
Uploader[1]: Getting from: https://ciscotac.rev-na.demo.vbrick.com:443/api/v1/videos/c4605aaf-
dc49-4cd7-9174-c46185ba1983/status Jun 17 22:24:42.035 user.info cms-02 Uploader[1]: Received
vbrick response status code: 200 Jun 17 22:24:42.035 user.info cms-02 Uploader[1]: vbrick
response: main.vbrickStatusResp{Status:"Ready"} Jun 17 22:24:42.035 user.info cms-02
Uploader[1]: file 20210618032309+0000_vid-id=c4605aaf-dc49-4cd7-9174-c46185ba1983@vbrick.mp4 vid
c4605aaf-dc49-4cd7-9174-c46185ba1983 status Ready Jun 17 22:24:42.035 user.info cms-02
Uploader[1]: Getting from: https://ciscotac.rev-na.demo.vbrick.com:443/api/v1/videos/c4605aaf-
dc49-4cd7-9174-c46185ba1983/playback-url Jun 17 22:24:42.200 user.info cms-02 Uploader[1]:
Received vbrick response 200
```

Troubleshooting

Streamer

1. No License

Streamer required "recorder" licnese on the server having callbridge component. If that is not there, or insufficient license is present, then errors as shown below will be seen in Event Logs.



```
2020-08-09   04:00:18.946   Info   API "TAC2" Space GUID: d4f2fa4c-4730-4dcb-9bcb-1d65c9e59016 <--> Call Correlator GUID: fceebc78-ba2f-4f6d-8c29-cd711cc6de09 <--> Internal GUI
2020-08-09   04:00:18.946   Info   conference d1be8391-ed8c-4406-994c-2e07b46fbf4c: lock state has changed to unlocked
2020-08-09   04:00:18.946   Info   API call leg d95ba532-16c3-4afe-bf82-5514c9219efd in call d1be8391-ed8c-4406-994c-2e07b46fbf4c (API call 5d9067f5-44fd-4a63-8a36-af2d43a09cad)
2020-08-09   04:00:18.946   Info   unable to start recording (space 'TAC2') -- no license
2020-08-09   04:00:18.947   Info   conference d1be8391-ed8c-4406-994c-2e07b46fbf4c has control/media GUID: 6d365821-ddfa-49d7-bf4c-323089c2c3e5
2020-08-09   04:00:18.947   Info   conference d1be8391-ed8c-4406-994c-2e07b46fbf4c named "TAC2"
2020-08-09   04:00:18.947   Info   call 2: configured - API call leg d95ba532-16c3-4afe-bf82-5514c9219efd with SIP call ID "58dda880-10001-21-93f492a@10.127.241.219"
2020-08-09   04:00:19.213   Info   call 2: compensating for far end not matching payload types
```

Make sure to add required license. Status of license can be checked CLI using the command "license"

```
cms1> license Feature: callbridge status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: turn status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: webbridge status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: customizations status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: local_license_mode status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: recording status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: personal status: Activated expiry: 2023-Apr-28 (690 days remain) Feature: shared status: Activated expiry: 2023-Apr-28 (690 days remain)
```

2.TLS port

• If TLS port configured and no certificates are applied. Configure certs for streamer to use TLS

• If certificate are not available. Configure the TCP port only

```
streamer> streamer sip listen a 7000 7001 streamer> streamer enable FAILURE: TLS port set but no certificates configured FAILURE: Streamer configuration not complete
```

Now you have 2 options, either to remove TLS Port, or to add SIP TLS Trust and Streamer Certificate

Cisco recommends to have TLS Port enabled.

3. RTMP Sream not configured correctly

You will see error in logs

```
daemon.info streamer streamer-sip[2280]: 144500.368 : INFO : call 3: retrieved stream URL from RTCP: "rtmp://broadcast:broadcast@10.106.81.40/test" daemon.info streamer streamer-sip[2280]: 144500.368 : INFO : call 3: parsing rtmp://broadcast:broadcast@10.106.81.40/test daemon.info streamer streamer-sip[2280]: 144500.368 : INFO : call 3: RTMP stream="test" daemon.info streamer streamer-sip[2280]: 144500.368 : INFO : call 3: RTMP server="rtmp://10.106.81.40:1935/test" daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: Connected to RTMP server daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: C2 pending - len 1536 daemon.info streamer streamer-sip[2280]: 144500.370 : : call 3: snd: create new chunk stream 2 daemon.info streamer streamer-sip[2280]: 144500.370 : : call 3: snd: create new chunk stream 3 daemon.info streamer streamer-sip[2280]: 144500.370 : INFO : call 3: RTMP sent chunk size of 4096 and connect message daemon.info streamer streamer-sip[2280]: 144500.410 : ERROR : call 3: connection : far end closed connection 5
```

Please check the procedure in streamer configuration and configure RTMP URL correctly in the format "**rtmp://<VBrickBroadcastUsername>:<VBrickBroadcastPassword>@<VBrick IP or FQDN>/live/NameoftheStream"**

4.Call Routing Related Issues

Because the CMS streamer is a SIP-based client and as discussed earlier, it requires routing to be in place. This could cause scenarios where calls might fail. Consider this example, where the CMS Callbridge sent an outbound call, but it failed with the following 'transaction timeout - no provisional responses sending INVITE' error

```
2021-06-28  17:37:02.412  Info     user 'guest300535034' starting streaming (space 'test')
2021-06-28  17:37:02.413  Info     API call leg bc0917df-589c-4628-887d-79481d322fed in call
63f0b174-831e-4a12-b4ee-27186d4162af (API call 00286960-9af9-4d5d-9ca7-20dd40425292)
2021-06-28  17:37:02.413  Info     call 44: outgoing SIP call to "demo@streamer.com" from space
"test"
2021-06-28  17:37:02.413  Info     call 44: configured - API call leg bc0917df-589c-4628-887d-
79481d322fed with SIP call ID "7d37a80e-7996-4e8d-aa87-77c9d4729cec"
2021-06-28  17:37:04.482  Info     call 42: receiver report 1 interval for rx video 0 = 6113ms
(period 6108ms) 00000000
2021-06-28  17:37:22.074  Info     call 44: falling back to unencrypted control connection...
2021-06-28  17:37:54.075  Info     call 44: ending; local SIP teardown with reason 7
(transaction timeout - no provisional responses sending INVITE) - not connected after 0:52
2021-06-28  17:37:54.075  Info     call 44: destroying API call leg bc0917df-589c-4628-887d-
79481d322fed
2021-06-28  17:37:54.076  Info     streaming call leg for space 'test' disconnected with reason
7 (transaction timeout - no provisional responses sending INVITE)
```

Review the Outbound calls settings on the CMS Callbridge servers to validate the location it is being sent to and if being set correctly. Also check if the callprofile is configured with correct streamer URI and same is associated with Cospace.

Uploader

1. Vbrick detials not correct.

You can see error in Uploader log

```
Jun 27 11:29:27.864 user.info streamer Uploader[1]: Received vbrick response 500 Jun 27
11:29:27.864 user.info streamer Uploader[1]: posting to:
https://sales.vbrick.com:443/api/v1/user/login Jun 27 11:29:47.870 user.info streamer
Uploader[1]: Received vbrick response 500 Jun 27 11:29:47.870 user.err streamer Uploader[1]:
Failed to initialise Vbrick Client Jun 27 11:29:47.870 user.err streamer Uploader[1]: vbrick
returned status code: 500
```

Make sure to have configured the correct credentials and port for the vbrick server. Also make sure Uploader should be able to reach CMS callbridge webadmin port.