Stream capturing from a SIP call

Overview

WCS can work as a WebRTC-SIP gateway. In this case, audio and video stream of a SIP call made through WCS can be captured and played in a browser or republished to other server.

Typical use case

1. A video call is established between WCS and a SIP device (SIP MCU, conference server or a SIP softphone)
2. WCS receives audio and video data from this SIP device
3. The WCS server redirects the received audio and video traffic to an RTMP server or another device capable of receiving and processing an RTMP stream

Supported protocols

- WebRTC
- RTMP
- SIP

Supported SIP codecs

- Video: H.264, VP8
- Audio: G.711, Speex, Opus

Supported RTMP codecs

- Video: H.264
- Audio: AAC, G.711, Speex

Supported WebRTC codecs

- Video: H.264, VP8
- Audio: Opus, G.711

REST API

Capturing and republishing of SIP calls is managed using REST API queries.

The REST query is an HTTP/HTTPS POST query as follows:
- HTTP: `http://test.flashphoner.com:8081/rest-api/call/startup` - HTTPS: `https://test.flashphoner.com:8444/rest-api/call/startup`

Where:
- `test.flashphoner.com` - is the address of the WCS server
- `8081` - is the standard REST / HTTP port of the WCS server
- `8444` - is the standard HTTPS port
- `rest-api` - is the required part of the URL
- `/call/startup` - the REST method used

General rules

1. Each SIP call can be associated with just one RTMP stream. If a new SIP call is initiated with the same RTMP URL and stream name as the existing call, that second call is declined by the server with the HTTP status of `409 Conflict`. However, publishing of a call to an RTMP stream using the `/push/startup` REST query does not limit the number of RTMP streams created for one call.
2. SIP Call ID of a call must be unique. An attempt to initiate a new SIP call with an already existing Call ID is declined by the WCS server with the HTTP status of `409 Conflict`.

REST methodss

/call/startup

Start a SIP call

REQUEST EXAMPLE

```
POST /rest-api/call/startup HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "callId":"123456711",
    "callee":"10000",
    "toStream":"stream1",
    "rtmpUrl":"rtmp://localhost:1935/live/",
    "rtmpStream":"rtmp_stream1",
    "hasAudio":true,
    "hasVideo":true,
    "sipLogin":"10009",
    "sipAuthenticationName":"10009",
    "sipPassword":"1234",
    "sipDomain":"226.226.225.226",
```

```
    "sipOutboundProxy":"226.226.225.226",
    "sipPort":"5060",
    "appKey":"defaultApp",
    "sipRegisterRequired":false
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
409	Conflict

/call/find

Find a SIP calls by some criteria

REQUEST EXAMPLE

```
POST /rest-api/call/find HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "callId": "R08NQya-5NMe5v7q-JNkboaS-CGMlFi"
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

[
    {
        "custom": {},
        "nodeId": null,
        "appKey": null,
        "sessionId": null,
        "callId": "R08NQya-5NMe5v7q-JNkboaS-CGMlFi",
        "parentCallId": null,
        "incoming": false,
        "status": "ESTABLISHED",
        "sipStatus": 200,
        "rtmpUrl": null,
        "rtmpStream": null,
```

```
        "streamName": null,
        "rtmpStreamStatus": null,
        "caller": "001",
        "callee": "002",
        "hasAudio": true,
        "hasVideo": false,
        "sdp": ...,
        "visibleName": "001",
        "inviteParameters": null,
        "mediaProvider": "Flash",
        "sipMessageRaw": null,
        "isMsrp": false,
        "target": null,
        "holdForTransfer": false
    }
]
```

RETURN CODES

Code	Reason
200	OK
404	Not found

/call/find_all

Find all the SIP calls

REQUEST EXAMPLE

```
POST /rest-api/call/find_all HTTP/1.1
Host: localhost:8081
Content-Type: application/json
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json

[
    {
        "custom": {},
        "nodeId": null,
        "appKey": null,
        "sessionId": null,
        "callId": "R08NQya-5NMe5v7q-JNkboaS-CGMlFi",
        "parentCallId": null,
        "incoming": false,
        "status": "ESTABLISHED",
        "sipStatus": 200,
        "rtmpUrl": null,
        "rtmpStream": null,
```

```
        "streamName": null,
        "rtmpStreamStatus": null,
        "caller": "001",
        "callee": "002",
        "hasAudio": true,
        "hasVideo": false,
        "sdp": ...,
        "visibleName": "001",
        "inviteParameters": null,
        "mediaProvider": "Flash",
        "sipMessageRaw": null,
        "isMsrp": false,
        "target": null,
        "holdForTransfer": false
    },
    ...
]
```

RETURN CODES

Code	Reason
200	OK
404	Not found

/call/terminate

Terminate the SIP call

REQUEST EXAMPLE

```
POST /rest-api/call/terminate HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "callId":"123456711"
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
404	Not found

/call/send_dtmf

Send a DTMF signal to the SIP call

REQUEST EXAMPLE

```
POST /rest-api/call/terminate HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
    "callId":"123456711",
    "dtmf":"9",
    "type":"RFC2833"
}
```

RESPONSE EXAMPLE

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
```

RETURN CODES

Code	Reason
200	OK
404	Not found

Parameters

Parameter name	Description	Example
callId	SIP Call ID - a unique identifier string	`Xq2tlLcX89tTjaji`
callee	SIP callee	`10001`
toStream	Name of the stream on the WCS server the call is published to	`call_stream1`
rtmpUrl	RTMP server ingest point address	`rtmp://rtmp-server.flashphoner.com:1935/live`
rtmpStream	Name of the RTMP stream on the RTMP server	`streamName2`

Parameter name	Description	Example
hasAudio	If `true`, SDP will have the `sendrecv` parameter in audio. If `false`, it gets `recvonly`.	`true`
hasVideo	If `true`, SDP will have the `sendrecv` parameter in video. If `false`, it gets `recvonly`.	`true`
status	Call status on the WCS server	`ESTABLISHED`
sipStatus	Associated SIP-status	`200`
rtmpStreamStatus	Status of the RTMP stream: `RTMP_STREAM_WAIT` - RTMP-stream is initializing `RTMP_STREAM_ACTIVE` - RTMP-stream has initialized and connection is established `RTMP_CONNECTION_LOST` - RTMP-connection is lost `RTMP_CONNECTION_FAILED` - RTMP-connection was not established	`RTMP_STREAM_ACTIVE`
caller	SIP caller	
visibleName	Displayed name of the caller	

SDP parameters `recvonly` and `sendrecv`

There are two main modes for SIP calls initiated by REST API:

1. `sendrecv`

```
v=0
o=Flashphoner 0 1437391553771 IN IP4 sip.flashphoner.com
s=Flashphoner/1.0
c=IN IP4 sip.flashphoner.com
t=0 0
m=audio 31022 RTP/AVP 8 0
c=IN IP4 46.101.139.106
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=ptime:20
a=rtcp:31023 IN IP4 sip.flashphoner.com
a=sendrecv
a=ssrc:1478013757 cname:rtp/audio/Xq2tlLcX89tTjaji
m=video 31024 RTP/AVP 112 113
```

```
c=IN IP4 sip.flashphoner.com
a=rtpmap:112 H264/90000
a=fmtp:112 packetization-mode=1; profile-level-id=420020
a=rtpmap:113 H264/90000
a=fmtp:113 packetization-mode=0; profile-level-id=420020
a=rtcp-fb:* ccm fir
a=rtcp-fb:* nack
a=rtcp-fb:* nack pli
a=rtcp:31025 IN IP4 sip.flashphoner.com
a=sendrecv
a=ssrc:979076678 cname:rtp/video/Xq2tlLcX89tTjaji
```

2. `recvonly`
3. SIP call parameters

```
hasAudio: false
hasVideo: false
```

4. SDP

```
v=0
o=Flashphoner 0 1437391553771 IN IP4 sip.flashphoner.com
s=Flashphoner/1.0
c=IN IP4 sip.flashphoner.com
t=0 0
m=audio 31022 RTP/AVP 8 0
c=IN IP4 46.101.139.106
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
a=ptime:20
a=rtcp:31023 IN IP4 sip.flashphoner.com
a=recvonly
a=ssrc:1478013757 cname:rtp/audio/Xq2tlLcX89tTjaji
m=video 31024 RTP/AVP 112 113
c=IN IP4 sip.flashphoner.com
a=rtpmap:112 H264/90000
a=fmtp:112 packetization-mode=1; profile-level-id=420020
a=rtpmap:113 H264/90000
a=fmtp:113 packetization-mode=0; profile-level-id=420020
a=rtcp-fb:* ccm fir
a=rtcp-fb:* nack
a=rtcp-fb:* nack pli
a=rtcp:31025 IN IP4 sip.flashphoner.com
a=recvonly
a=ssrc:979076678 cname:rtp/video/Xq2tlLcX89tTjaji
```

In both cases WCS does not send RTP audio and video traffic, because it is the REST client that is the initiator of the call, which is not the source of audio and video streams. WCS can explicitly set in SDP that there will be no audio and video traffic from its side (the `recvonly` mode).

If your SIP-device is a softphone or another SIP phone, most likely it will drop calls (in the `sendrecv` mode) within approximately a minute after connection is established. This is because of lack of RTP traffic from WCS.

Some softphones correctly support the `recvonly` mode, for example, MicroSIP. In other softphones like Bria, RTP activity timer can be manually increased to provide longer duration of a call in the `sendrecv` mode.

If your SIP device is an MCU or a SIP conference server, it should work correctly with the `recvonly` mode, and long calls can be established.

Additional call status information

WCS uses the built-in `callApp` REST hook application to send intermediate call statuses.

Examples

`TRYING`, `RTMP_STREAM_WAIT`

```
{
  "nodeId" : "w9NiNKZCtjK6C4vz1zVnzGWBJGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2tlLcX89tTjaji_3",
  "incoming" : false,
  "status" : "TRYING",
  "sipStatus" : 100,
  "rtmpUrl" : "rtmp://rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
  "rtmpStreamStatus" : "RTMP_STREAM_WAIT",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

`ESTABLISHED`, `RTMP_STREAM_ACTIVE`

```
{
  "nodeId" : "w9NiNKZCtjK6C4vz1zVnzGWBJGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2tlLcX89tTjaji_3",
  "incoming" : false,
  "status" : "ESTABLISHED",
  "sipStatus" : 200,
  "rtmpUrl" : "rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
```

```
  "rtmpStreamStatus" : "RTMP_STREAM_ACTIVE",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

`ESTABLISHED`, `RTMP_CONNECTION_LOST`

```
{
  "nodeId" : "w9NiNKZCtjK6C4vz1zVnzGWBJGkA2Cke@192.168.88.101",
  "appKey" : "callApp",
  "sessionId" : "127.0.0.1:1403649870519623722",
  "callId" : "Xq2tlLcX89tTjaji_3",
  "incoming" : false,
  "status" : "ESTABLISHED",
  "sipStatus" : 200,
  "rtmpUrl" : "rtmp.flashphoner.com:1935/live",
  "rtmpStream" : "streamName2",
  "rtmpStreamStatus" : "RTMP_CONNECTION_LOST",
  "caller" : "3000",
  "callee" : "3002",
  "hasAudio" : true,
  "hasVideo" : true,
  "visibleName" : "3000",
  "mediaProvider" : "Flash",
  "isMsrp" : false
}
```

These are notifications that are sent only locally on the server side via the internal REST interface. See the REST Hooks section to get more information about internal REST applicaitons. Also, a third-party web application can be created to receive notifications from the WCS server.

Known issues

1. SIP call stream may be played unsmoothly via HLS without transcoding

Symptoms

When republishing SIP as RTMP stream to Wowza servers and when receiving a stream from Wowza via HLS, a subscriber can see freezes, short time non-synchronous playback.

Solution

Enable transcoding on the server by the following parameter in flashphoner.properties file:

```
disable_streaming_proxy=true
```

2. Audio only SIP call stream requires a proper constraints to be played

Symptoms

When SIP call is redirecting to stream with SIP as Stream function, the audio only call stream does not play as WebRTC from WCS.

Solution

The audio SIP call stream should be played as audio only stream in a browser by explicitly constraints setting in player script when stream is created, for example

```
session.createStream({constraints:{audio:true,video:false}}).play();
```

3. SIP login and authentication name should not contain any spaces or special characters

Symptoms

SIP call cannot be created, `/call/startup` REST API query returns

```
HTTP/1.1 500 Internal Server Error
Access-Control-Allow-Origin: *
Content-Type: application/json

{
   "error": "Internal Server Error",
   "exception":
"com.flashphoner.rest.server.exception.InternalErrorException",
   "message": "SIP login or authentication name contains reserved
symbols",
   "path": "/rest-api/call/startup",
   "status": 500,
   "timestamp": 1559029484840
}
```

Solution

According to RFC 3621, `SIP Login` and `SIP Authentication name` should not contain any of unescaped spaces and special characters and should not be enclosed in angle brackets `<>`.

For example, this is not allowed by the standard

```
sipLogin='Ralf C12441@host.com'
sipAuthenticationName='Ralf C'
sipPassword='demo'
sipVisibleName='null'
```

and this is allowed

```
sipLogin='Ralf_C12441'
sipAuthenticationName='Ralf_C'
sipPassword='demo'
sipVisibleName='Ralf C'
```

4. RTP traffic buffering should be enabled in some cases when republishing SIP as Stream or SIP as RTMP

Symptoms

Audio and video may be out of sync when playing a SIP call stream

Solution

Update WCS to build 5.2.1910 and enable RTP traffic buffering

```
rtp_in_buffer=true
```