Захват потока с SIP-звонка

Описание

WCS может выступать в качестве WebRTC-SIP шлюза. При этом аудио- и видеопоток SIP-звонка, произведенного через WCS, может быть захвачен и воспроизведен в браузере, либо ретранслирован на другой сервер.

Типичный сценарий использования

1. Между WCS и SIP-устройством (SIP MCU, сервер конференций или SIP-софтфон) установлен видеозвонок
2. WCS получает аудио и видео данные с этого SIP-устройства
3. Полученный аудио и видео трафик WCS-сервер перенаправляет на RTMP-сервер или другое устройство, способное принять и обработать RTMP-поток

Поддерживаемые протоколы

- WebRTC
- RTMP
- SIP

Поддерживаемые кодеки для SIP

- Видеокодеки: H.264, VP8
- Аудиокодеки: G.711, Speex, Opus

Поддерживаемые кодеки для RTMP

- Видеокодеки: H.264
- Аудиокодеки: AAC, G.711, Speex

Поддерживаемые кодеки для WebRTC

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

REST API

Захват и ретрансляция SIP-звонков управляется при помощи REST API вызовов.

REST-запрос должен быть HTTP/HTTPS POST запросом в таком виде:

- HTTP: `http://sip-as-rtmp.flashphoner.com:8081/rest-api/call/startup`
- HTTPS: `https://sip-as-rtmp.flashphoner.com:8444/rest-api/call/startup`

Здесь:

- `sip-as-rtmp.flashphoner.com` - адрес WCS-сервера
- `8081` - стандартный REST / HTTP порт WCS-сервера
- `8444` - стандартный HTTPS порт
- `rest-api` - обязательная часть URL
- `/call/startup` - используемый REST-метод

Общие правила

1. Каждый SIP-звонок при создании может быть ассоциирован только с одним RTMP-потоком. В случае, если инициируется новый SIP-звонок с тем же RTMP URL и именем потока, как у существующего звонка, этот второй звонок будет отклонен сервером с HTTP статусом `409 Conflict`. Однако, при ретрансляции звонка в RTMP-поток при помощи REST-запроса `/push/startup`, количество RTMP-потоков, создаваемых из одного звонка, не ограничивается.
2. SIP Call ID звонка должен быть уникальным. Попытка инициировать новый SIP-звонок с уже существующим Call ID будет отклонена WCS-сервером с HTTP статусом `409 Conflict`.

REST-методы

/call/startup

Начать SIP звонок

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

Найти SIP звонок по указанным критериям

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

Найти все SIP звонки

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

Завершить SIP звонок

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

Отправить сигнал DTMF в SIP звонок

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

Параметры

Параметр	Описание	Пример
callId	Уникальный идентификатор звонка	`Xq2tlLcX89tTjaji`
callee	Вызываемый абонент	`10001`
toStream	Имя потока, который будет опубликован на WCS сервере из SIP звонка	`call_stream1`

Параметр	Описание	Пример
rtmpUrl	Входной URL RTMP сервера для публикации потока звонка	`rtmp://rtmp-server.flashphoner.com:1935/live`
rtmpStream	Имя потока на RTMP сервере (stream key)	`streamName2`
hasAudio	Если `true`, SDP будет иметь `sendrecv` параметр в аудио части. Если `false`, то `recvonly`	`true`
hasVideo	Если `true`, SDP будет иметь `sendrecv` параметр в видео части. Если `false`, то `recvonly`	`true`
status	Статус звонка на WCS-сервере	`ESTABLISHED`
sipStatus	Соответствующий статус звонка на SIP стороне	`200`
rtmpStreamStatus	Статус RTMP-потока: `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	Вызывающий абонент	
visibleName	Отображаемое имя вызывающего абонента	

SDP параметры `recvonly` и `sendrecv`

Существует два основных режима для SIP звонков, созданных по 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
```

В обоих случаях WCS не отправляет RTP аудио и видео трафик, т.к. инициатором звонка выступает REST-клиент, который не является источником аудио и видео потоков.

При этом WCS может явно указать в SDP, что с его стороны не будет аудио и видео трафика (режим `recvonly`).

Если ваше SIP-устройство - это софтфон или другой SIP-телефон, он скорее всего будет сбрасывать звонок (в режиме `sendrecv`) примерно в течение минуты после установки соединения. Это происходит из-за отсутствия RTP-трафика со стороны WCS.

Некоторые софтфоны корректно поддерживают режим `recvonly`, например MicroSIP. В других софтфонах, таких как Bria, таймер проверки RTP-активности может быть увеличен, для того чтобы получить большую длительность звонка в режиме `sendrecv`.

Если ваше SIP-устройство - это MCU или сервер SIP-конференций, скорее всего оно корректно отработает режим `recvonly`, и звонок сможет быть установлен на длительное время.

Дополнительная информация по статусам звонка

WCS использует внутреннее приложение `callApp` для передачи промежуточных статусов на бэкенд сервер.

Примеры

`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
}
```

Это уведомления, которые передаются встроенному или внешнему бэкенд серверу через внутренний REST-интерфейс. См. раздел REST hooks, чтобы получить больше информации о внутренних REST-приложениях. Кроме этого, может быть создано стороннее web-приложение, которое будет получать уведомления с WCS-сервера.

Известные проблемы

1. Поток SIP звонка может играть неплавно как HLS без транскодинга

Симптомы

При ретрансляции SIP как RTMP на серверы Wowza и получении потока с Wowza по HLS зритель наблюдает фризы, кратковременную рассинхронизацию.

Решение

Включить транскодинг на сервере, указав настройку в файле flashphoner.properties

```
disable_streaming_proxy=true
```

2. Поток из аудио SIP звонка может не играть как WebRTC без указания соответствующих ограничений

Симптомы

При ретрансляции SIP as Stream поток аудиозвонка не воспроизводится по WebRTC в браузере

Решение

Поток аудиозвонка необходимо воспроизводить в браузере как аудиопоток, указав ограничение явным образом при создании потока в скрипте плеера, например

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

3. Параметры `SIP Login`, `SIP Authentification name` не должны содержать пробелов и спецсимволов

Симптомы

Звонок не совершается, при создании звонка при помощи REST API `/call/startup` возвращается

```
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
}
```

Решение

Согласно RFC 3621, `SIP Login` и `SIP Authentification name` не должны содержать неэкранированных пробелов, спецсимволов и не должны заключаться в угловые скобки `<>`.

Например, такое заполнение полей не соответствует стандарту

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

а такое соответствует

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

4. При ретрансляции видеозвонка в поток в некоторых случаях необходимо включить буферизацию RTP трафика

Симптомы

При видеозвонках на некоторые софтфоны заметна рассинхронизация между видео и аудио при проигрывании потока

Решение

Обновить WCS до сборки 5.2.1910 и включить буферизацию RTP трафика

```
rtp_in_buffer=true
```