

# In a player via RTSP

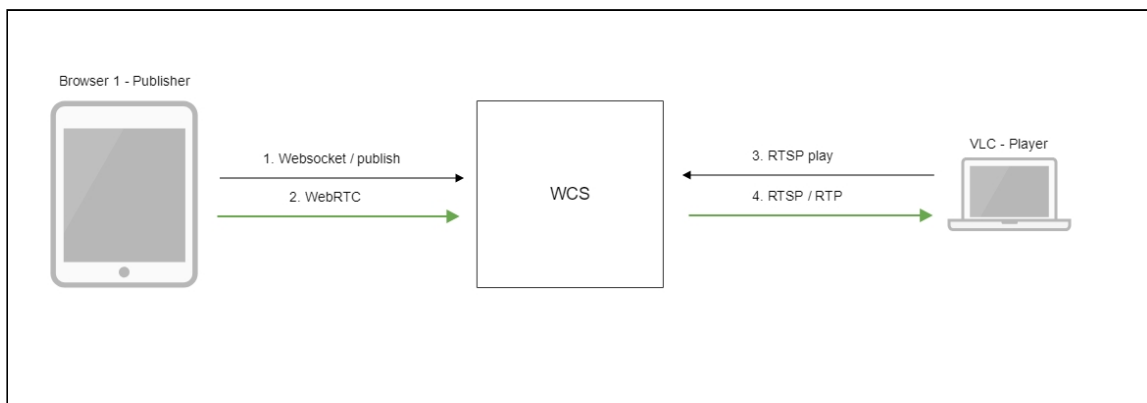
## Overview

A stream published on the WCS server can be played via RTSP in a third-party player. In this case, WCS itself acts as an [RTSP-source](#).

## Codecs supported

- Video: H.264, VP8, H265 (since build [5.2.1577](#))
- Audio: AAC, G.711, Speex

## Operation flowchart

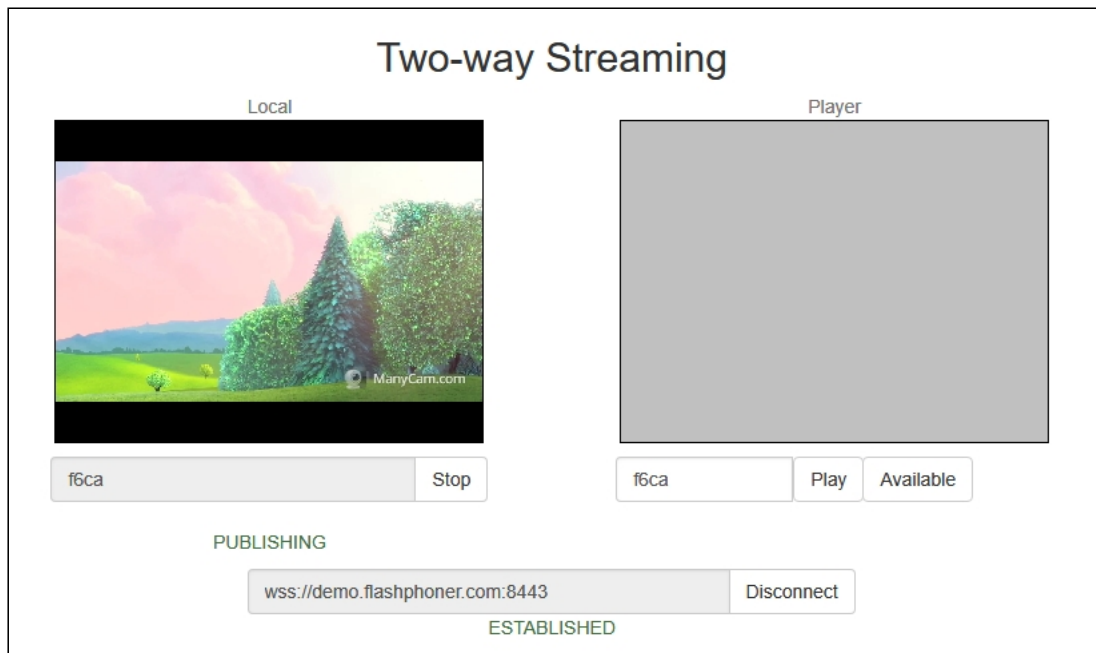


1. The browser establishes a connection to the server via Websocket
2. The browser captures the camera and the microphone and sends the WebRTC stream to the server
3. VLC Player establishes a connection to the server via RTSP
4. VLC Player receives the stream from the server and plays it

## Quick manual on testing

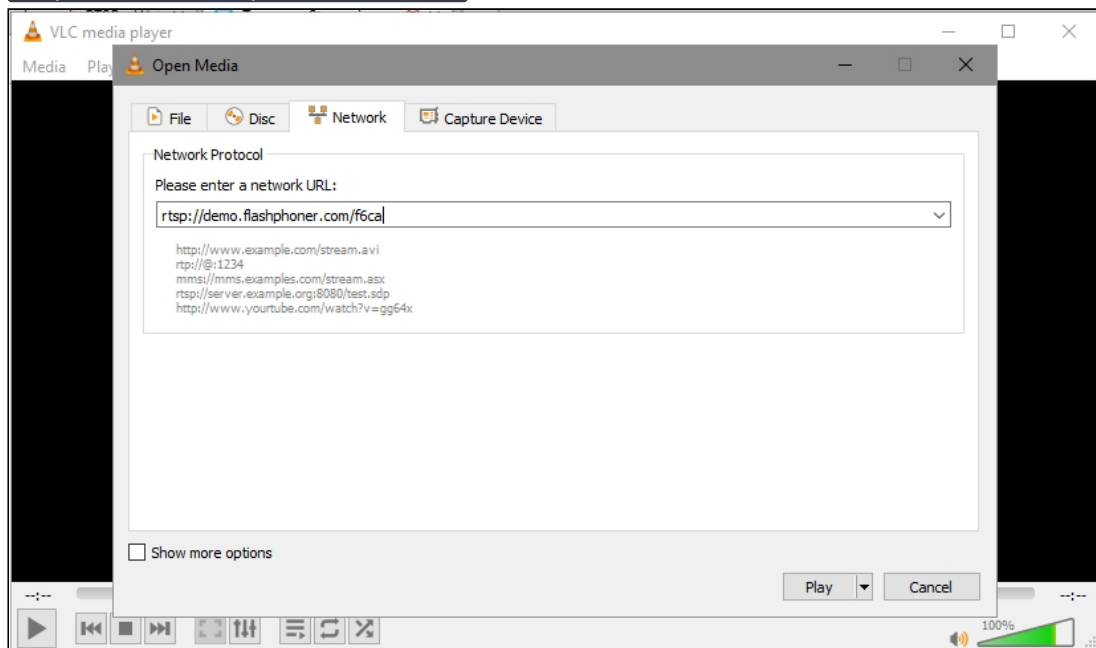
1. For the test we use:
2. the demo server at `demo.flashphoner.com`;
3. the [Two Way Streaming](#) web application to publish the stream;
4. VLC Player to play the stream.

5. Open the Two Way Streaming application. Click **Connect**, then **Publish**. Copy the identifier of the stream:

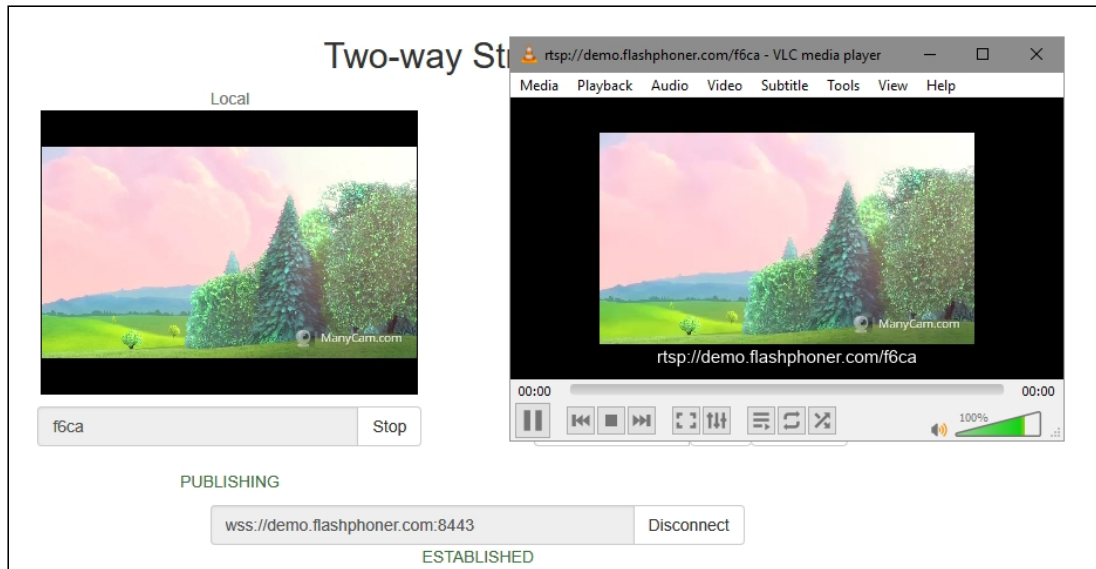


6. Run VLC, select the **Media - Open network stream** menu. Enter the URL of the WCS server and enter the identifier of the stream, in this example

`rtsp://demo.flashphoner.com/fc6a:`

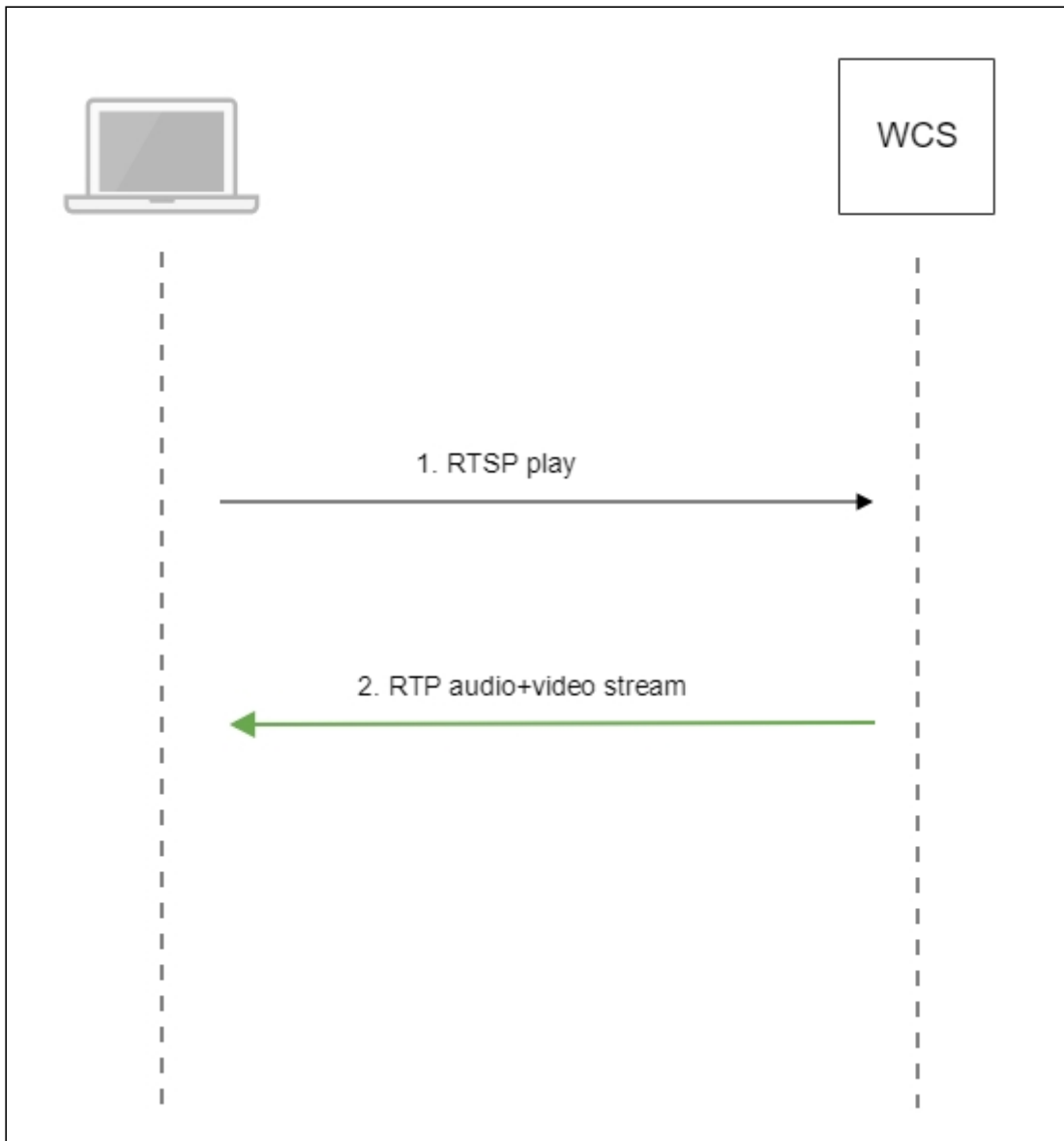


7. Click the **Play** button. The player starts playing the stream:



## Call flow

Below is the call flow when playing a stream via RTSP in a software player



1. The software player establishes a connection to the WCS server via RTSP.
2. The software player receives the media stream from WCS.

## RTSP server configuration

By default, TCP port 554 is used to listen RTSP connections. This value can be set with the following parameter in `flashphoner.properties` file

```
rtsp.port=554
```

Since build [5.2.801](#), WCS is running from `flashphoner` user for security reasons. Therefore RTSP server may not be launched because TCP ports in range 0-1000 are unavailable to non-root users. In this case RTSP port should be changed, for example

```
rtsp.port=5554
```

## RTSP playback authentication via REST hook

RTSP playback authentication via [REST hook](#) can be set up if necessary. To do this, the following parameter should be set in [flashphoner.properties](#) file:

```
rtsp_server_auth_enabled=true
```

When RTSP connection is established, `/playRTSP` query will be sent to backend server application `defaultApp`

```
URL:http://localhost:8081/apps/EchoApp/playRTSP
OBJECT:
{
  "nodeId" : "NTk1tLorQ001lGbPJufexrKceubGCR0k@192.168.1.5",
  "appKey" : "defaultApp",
  "sessionId" : "/192.168.1.100:59711/192.168.1.5:554",
  "mediaSessionId" : "29868390-73ee-4f49-ba92-78d717c53070-test-RTSP",
  "name" : "rtsp://p11.flashphoner.com:554/test",
  "mediaProvider" : "RTSP",
  "userAgent" : "LibVLC/3.0.4 (LIVE555 Streaming Media v2016.11.28)"
}
```

Such query will be sent on using every RTSP method except `OPTIONS`. If backend server responds `200 OK`, WCS server allows to execute RTSP method and to play RTSP stream. If backend server returns `403 Forbidden`, WCS server breaks the connection with RTSP client.

Thus, RTSP client can be authenticated by RTSP stream URL, User-Agent, client and server IP address and port.

## Custom access key and backend application usage for RTSP playback authentication

Since build [5.2.1008](#) it is possible to set custom authentication key (token) in RTSP URL, for example

```
rtsp://wcs:5554/streamName?aclAuth=1254789
```

The following REST hook `/playRTSP` content will be sent to `defaultApp` backend application

```
{
  "nodeId" : "XLepaP08Uyz9LqAjXHWnwuFxrEri0fCj@192.168.1.39",
  "appKey" : "testApp",
  "sessionId" : "/192.168.1.83:55195/192.168.1.39:5554",
  "mediaSessionId" : "71317dfc-0222-4acd-912e-57e67f2a272a-streamName-RTSP",
  "name" : "rtsp://wcs:5554/streamName?aclAuth=1254789",
}
```

```
...
"mediaProvider" : "RTSP",
"userAgent" : "LibVLC/3.0.8 (LIVE555 Streaming Media v2016.11.28)",
"custom" : {
  "aclAuth" : "1254789"
}
}
```

Authentication token name is set by the following parameter as like for HLS playback

```
client_acl_property_name=aclAuth
```

It is also possible to set custom backend application key

```
rtsp://wcs:5554/streamName?appKey=customAppKey&aclAuth=1254789
```

In this case REST hook `/playHLS` will be sent to backend application with defined key (`customAppKey` in the example above).

## Adjusting RTSP playback parameters

To adjust RTSP playback parameters, for example, to change audio or video codec, SDP setting file [rtsp\\_server.sdp](#) should be used. Note that this file should contain WCS server IP address.

## Playing H265 without transcoding

Since build [5.2.1577](#) it is possible to play [MPEG-TS](#) H265 stream via RTSP. To do this, H265 codec must be set in `rtsp_server.sdp` file:

```
v=0
o=- 1988962254 1988962254 IN IP4 0.0.0.0
c=IN IP4 0.0.0.0
t=0 0
a=sdplang:en
a=range:npt=now-
a=control:*
m=audio 0 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/48000/2
a=fmtp:96 profile-level-id=1;mode=AAC-
hbr;sizeLength=13;indexLength=3;indexDeltaLength=3
a=control:audio
a=recvonly
m=video 0 RTP/AVP 119
a=rtpmap:119 H265/90000
a=control:video
a=recvonly
```

### Warning

Streams published in H264, VP8, or MPV codecs may not be played as H265! Use this codec to play MPEG-TS H265 streams only

## Known issues

### 1. Frame loss and picture artefacts can occur when HD stream is played via RTSP

#### Symptoms

Some artefacts are observed, and player log contains lost frame reports when HD stream is played via RTSP.

#### Solution

Switch player to RTSP interleaved mode, for example, in VLCe settings tab Input/Codecs set radiobutton Live 555 stream transport to RTP over RTSP (TCP)

### 2. Freezes can occur when WebRTC stream is played via RTSP, if player receives no key frame

#### Symptoms

Freezes when WebRTC stream is played as RTSP in VLC player

#### Solution

Enable the following setting in [flashphoner.properties](#) file

```
periodic_fir_request=true
```

### 3. VLC on Windows can display audio parameters incorrectly

When stream is played as RTSP in VLC on Windows, audio samplerate and bitrate can be displayed incorrectly due to [VLC known bug](#).

