

# In a player via RTSP

- [Overview](#)
  - [RTSP-codecs](#)
  - [Operation flowchart](#)
- [Quick manual on testing](#)
  - [Publishing a video stream on the server and playing it via RTSP in a software player](#)
- [Call flow](#)
- [RTSP playback authentication via REST hook](#)
- [Adjusting RTSP playback parameters](#)
- [Known issues](#)

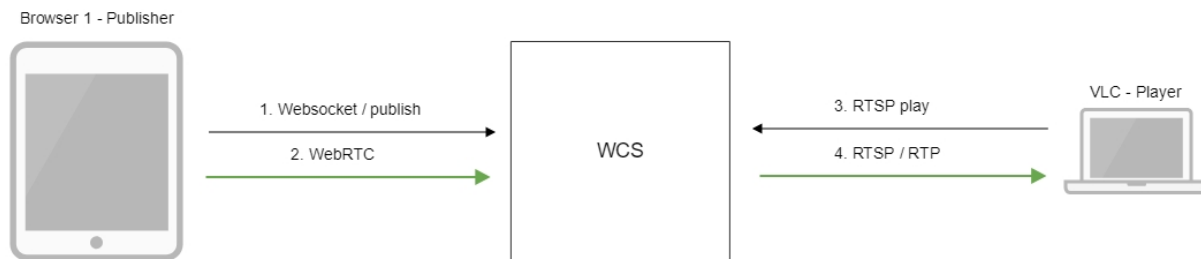
## Overview

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

## RTSP-codecs

- Video: H.264, VP8
- 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

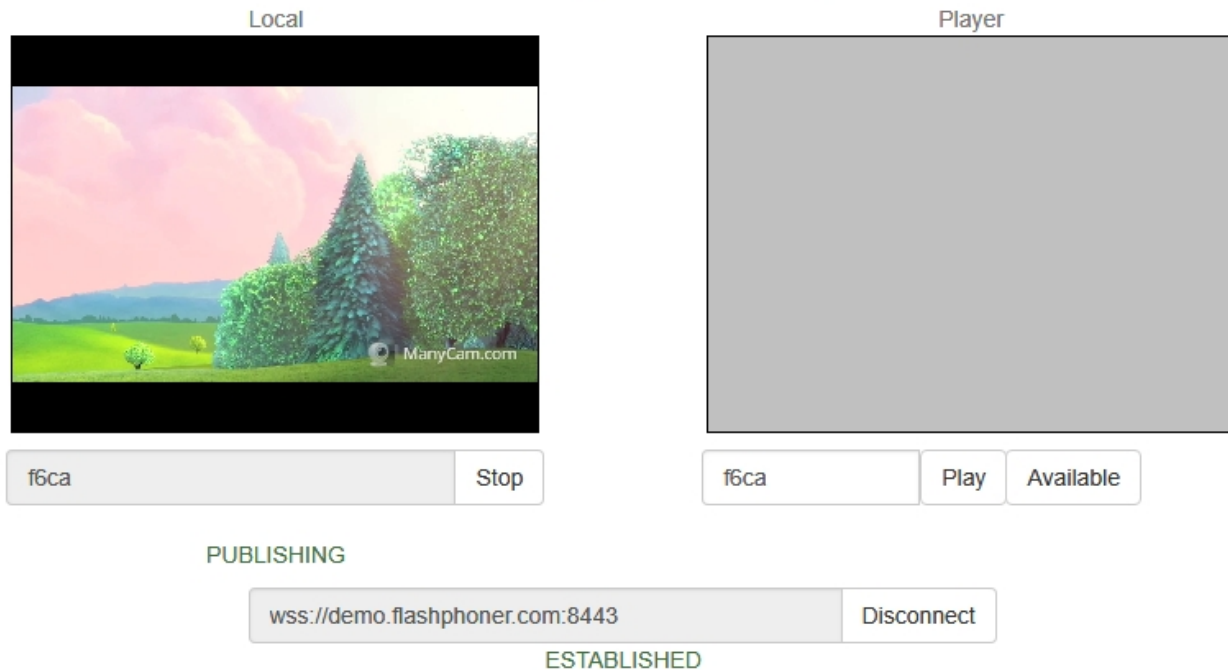
### Publishing a video stream on the server and playing it via RTSP in a software player

1. For the test we use:

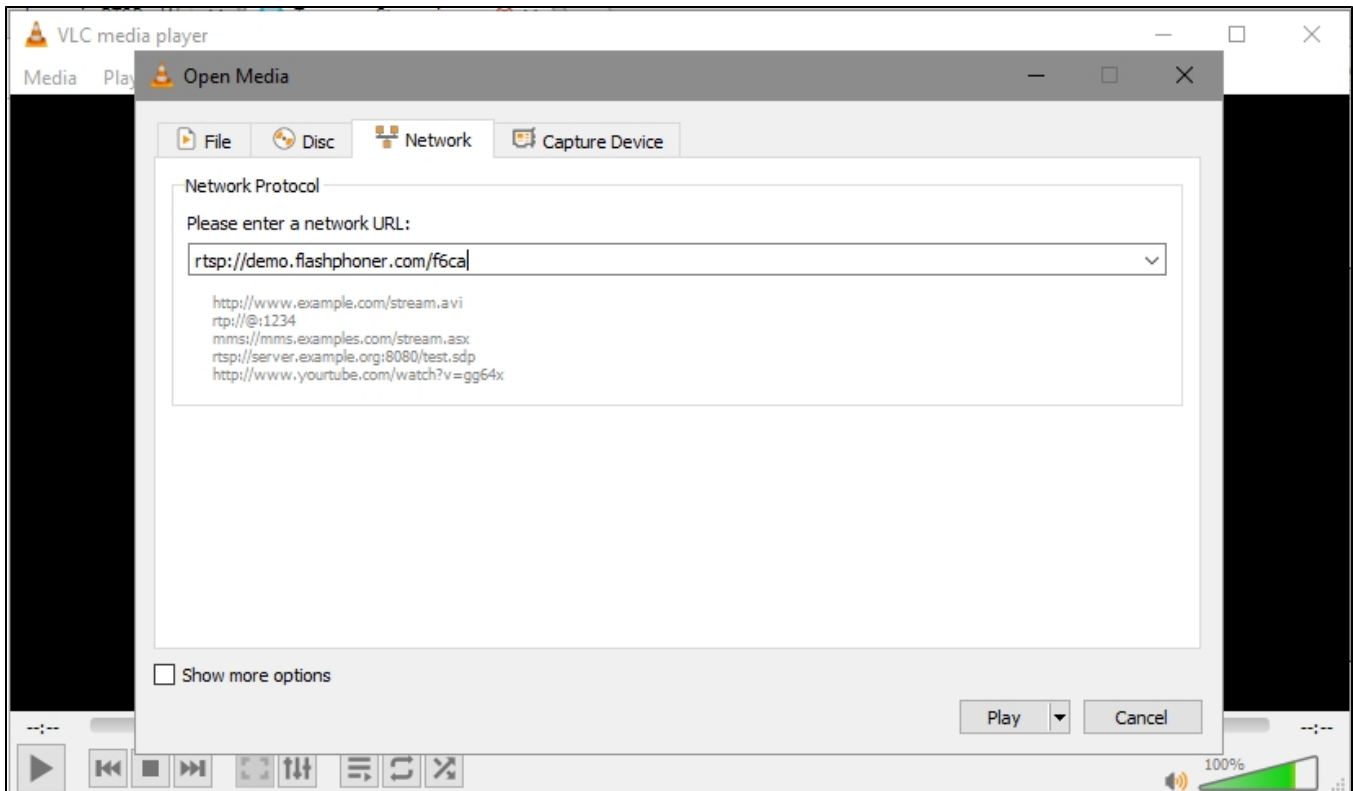
- the demo server at [demo.flashphoner.com](http://demo.flashphoner.com);
- the [Two Way Streaming](#) web application to publish the stream;
- VLC Player to play the stream.

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

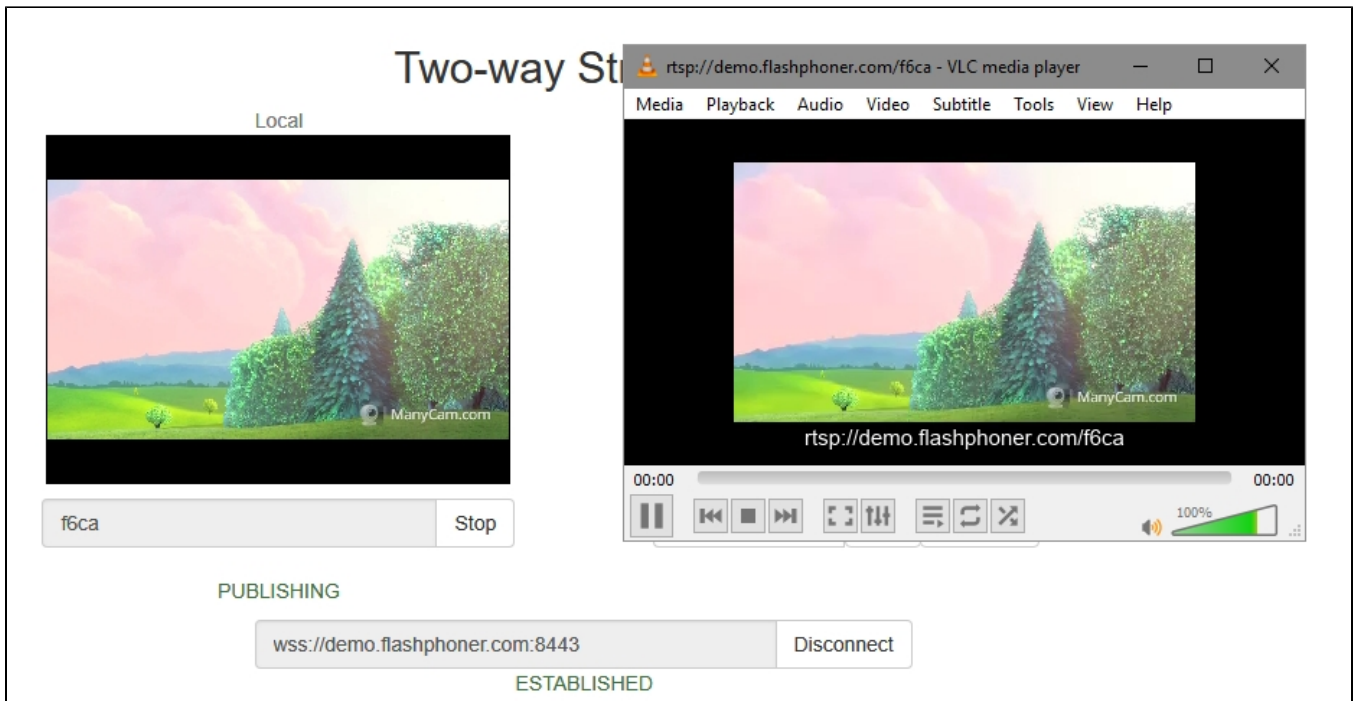
## Two-way Streaming



3. 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/f6ca`:

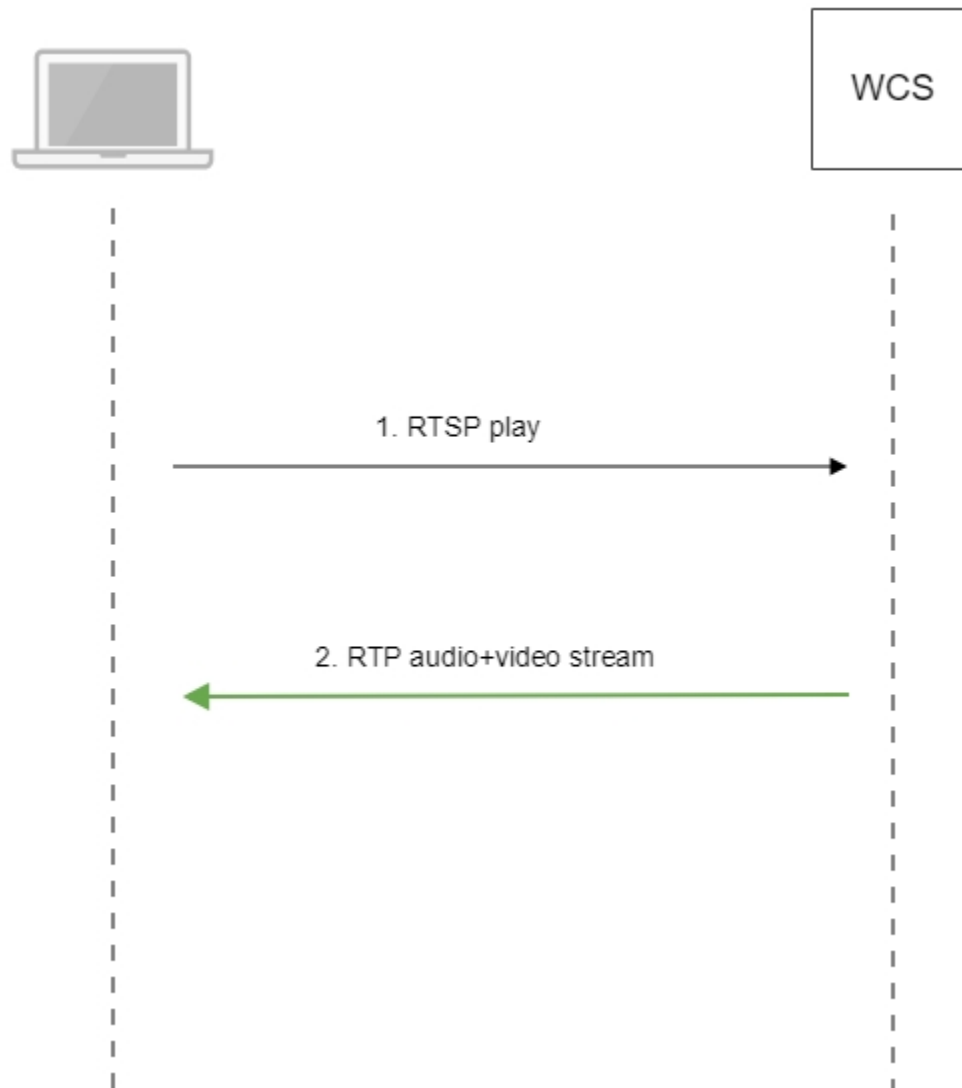


4. Click the "Play" button. The player starts playing the stream:



## 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 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

```
URL:http://localhost:8081/apps/EchoApp/playRTSP
OBJECT:
{
  "nodeId" : "NTkltLorQ001lGbPJufexrKceubGCR0k@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 excepting 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.

## 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.

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