In a player via RTMP

- Overview
  - RTMP-codecs
  - Operation flowchart
- Quick manual on testing
  - Publishing a video stream on the server and playing it via RTMP in a software player
- Call flow
- Parsing stream URL parameters
- Track order management in RTMP stream
- RTMP playback sound suppression

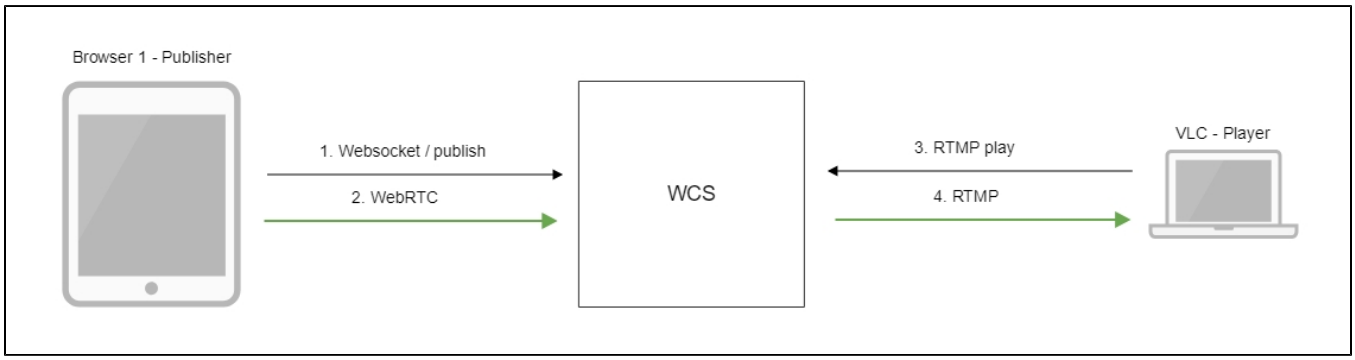
Overview

A stream published on the WCS server can be played via RTMP in a third-party player. In this case, WCS itself serves as anRTMP-source.

RTMP-codecs

- Video: H.264
- 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 RTMP
4. VLC Player receives the stream from the server and plays it

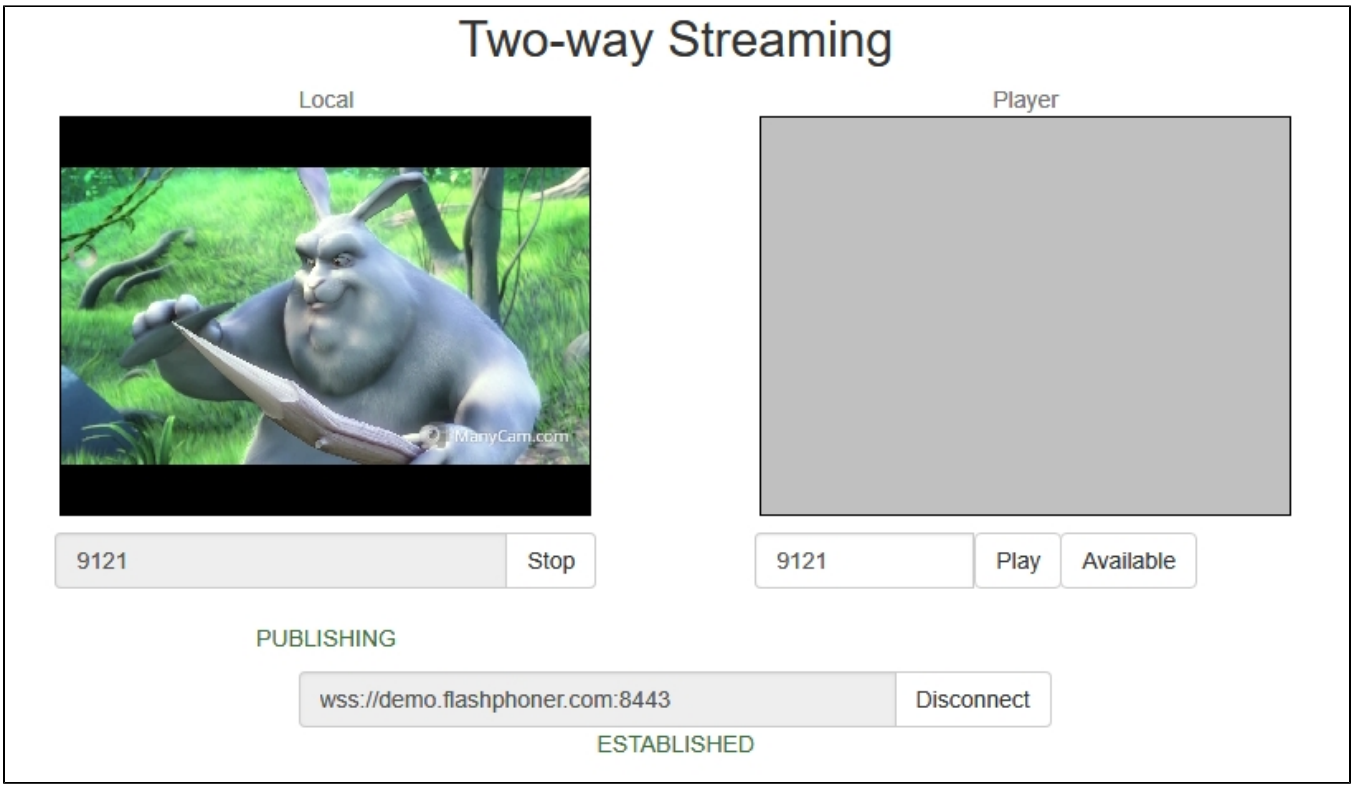
Quick manual on testing

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

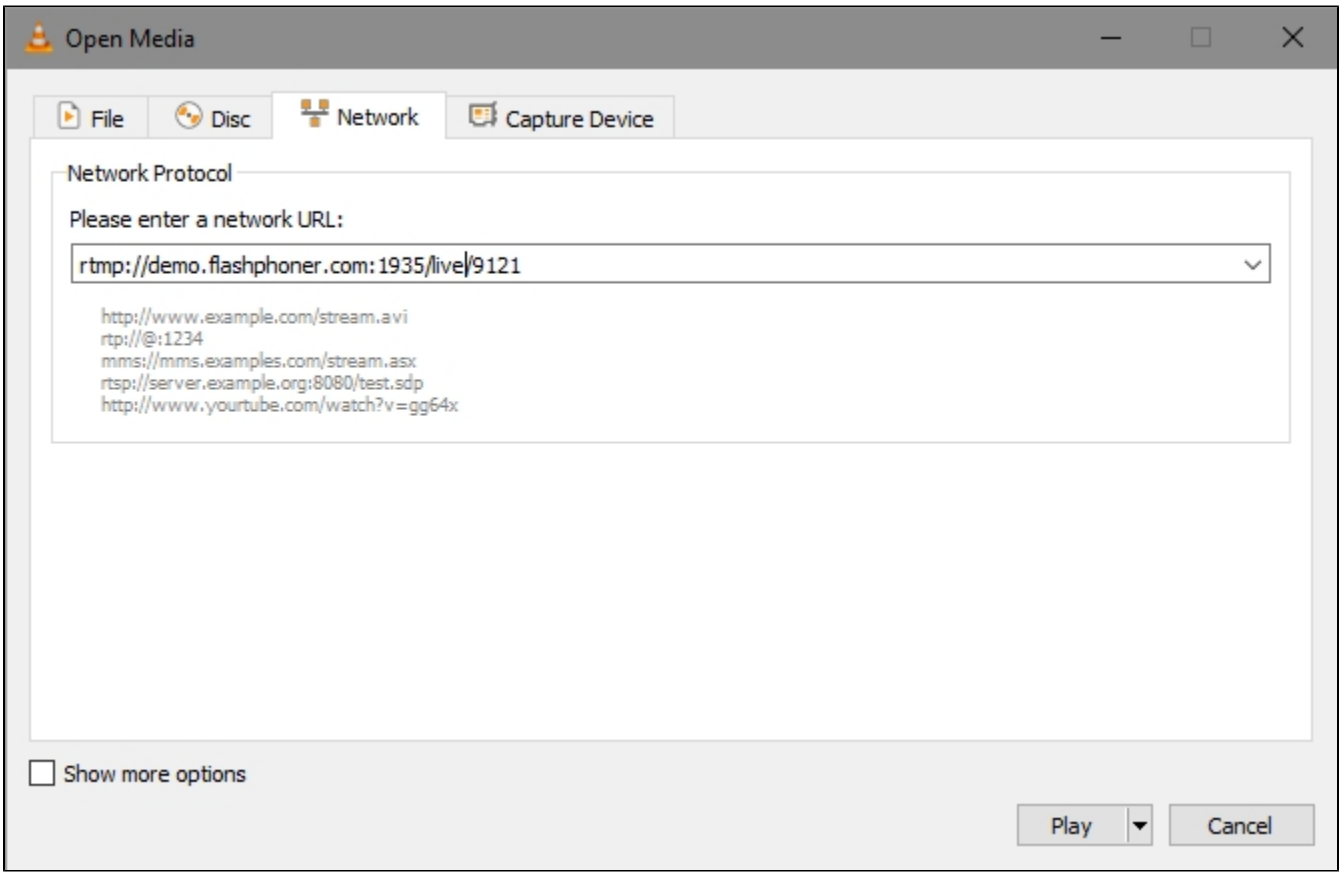
1. For the test we use:

- the demo server at demo.flashphoner.com;
- theTwo Way Streamingweb 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:



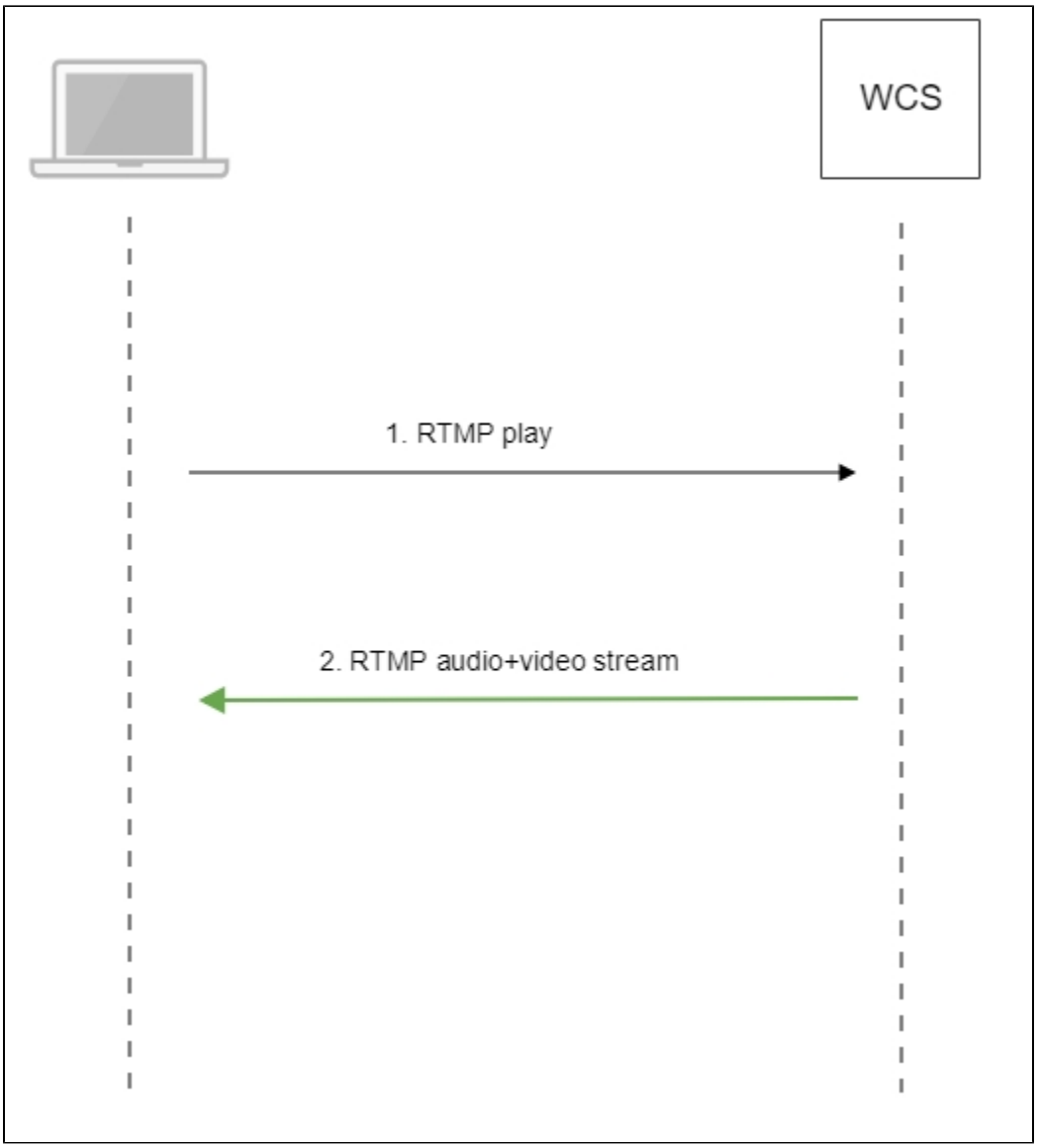
3. Run VLC, select the "Media - Open network stream" menu. Enter the URL of the WCS server and enter the identifier of the stream, in this exampe: rtmp://demo.flashphoner.com:1935/live/9121:



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

Call flow

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



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

Parsing stream URL parameters

When RTMP stream is published or played on WCS, RTMP connection and stream parameters may be set in stream URL like this:

```
rtmp://host:1935/live?connectParam1=val1&connectParam2=val2/streamName?streamParam1=val1&streamParam2=val2
```

Where

- host is WCS server hostname;
- connectParam1, connectParam2 are RTMP connection parameters;
- streamName is stream name on server;
- streamParam1, streamParam2 are stteam parameters.

WCS server passes the parameters to backend server in REST hook in `customfield`, for example:

Connection parameters

```
URL:http://localhost:8081/apps/EchoApp/connect
OBJECT:
{
"nodeId" : "Qb3rAjf3lzoy6PEl1WZkUhRG1DsTykgj@192.168.1.1",
"appKey" : "flashStreamingApp",
"sessionId" : "/127.0.0.1:5643/192.168.1.1:1935",
"useWsTunnel" : false,
"useWsTunnelPacketization2" : false,
"useBase64BinaryEncoding" : false,
"keepAlive" : false,
"custom" : {
"connectParam1" : "val1",
"connectParam2" : "val2"
},
"login" : "rQq83sodiCPY0pJXCxGO"
}
```

Publishing parameters

```
URL:http://localhost:8081/apps/EchoApp/publishStream
OBJECT:
{
"nodeId" : "Qb3rAjf3lzoy6PEl1WZkUhRG1DsTykgj@192.168.1.1",
"appKey" : "flashStreamingApp",
"sessionId" : "/127.0.0.1:5643/192.168.1.1:1935",
"mediaSessionId" : "627990f9-8fe5-4e92-bb2a-863cc4eb43de",
"name" : "stream1",
"published" : true,
"hasVideo" : false,
"hasAudio" : true,
"status" : "NEW",
"record" : true,
"width" : 0,
"height" : 0,
"bitrate" : 0,
"minBitrate" : 0,
"maxBitrate" : 0,
"quality" : 0,
"mediaProvider" : "Flash",
"custom" : {
"streamParam1" : "val1",
"streamParam2" : "val2"
}
}
```

Playback parameters

```
URL:http://localhost:8081/apps/EchoApp/playStream
OBJECT:
{
"nodeId" : "Qb3rAjf3lzoy6PEl1WZkUhRG1DsTykgj@192.168.1.1",
"appKey" : "flashStreamingApp",
"sessionId" : "/127.0.0.1:5643/192.168.1.1:1935",
"mediaSessionId" : "stream1/127.0.0.1:5643/192.168.1.1:1935",
"name" : "stream1",
"published" : false,
"hasVideo" : true,
"hasAudio" : true,
"status" : "NEW",
"record" : false,
"width" : 0,
"height" : 0,
"bitrate" : 0,
"minBitrate" : 0,
"maxBitrate" : 0,
"quality" : 0,
"mediaProvider" : "Flash",
"custom" : {
"streamParam1" : "val1",
"streamParam2" : "val2"
}
}
```

This feature can be used for example to authenticate client on backend server while publishing or playing RTMP-stream on WCS server.

Track order management in RTMP stream

Most players on various platforms suppose video track to be first in RTMP stream. To guarantee this order and to send videodata before audiodata, set the following parameter in flashphoner.propertiesfile:

```
rtmp_send_video_first=true
```

Note that if this setting is active, a stream containing audio track only can not be played as RTMP because audiodata will not be sent to client.

RTMP playback sound suppression

Sound may be disabled while stream published on server playback as RTMP. To do this, the following RTMP URL parameter should be passed:

```
rtmp://yourserver:1935/live?suppress_sound=true/streamName
```

In this case audio track will be replaced by silence.