

# В плеере по RTSP

- Описание
  - RTSP-кодеки
  - Схема работы
- Краткое руководство по тестированию
  - Трансляция видеопотока на сервер и воспроизведение его по RTSP в программном плеере
- Последовательность выполнения операций (Call flow)
- Настройка RTSP сервера
- Аутентификация воспроизведения RTSP при помощи REST hook
  - Использование собственного ключа доступа и собственного приложения на бэкенде для аутентификации
- Настройка параметров проигрывания RTSP
  - Проигрывание H265 без транскодинга
  - Поддержка динамического определения кодеков
- Поддержка interleaved режима
- Известные проблемы

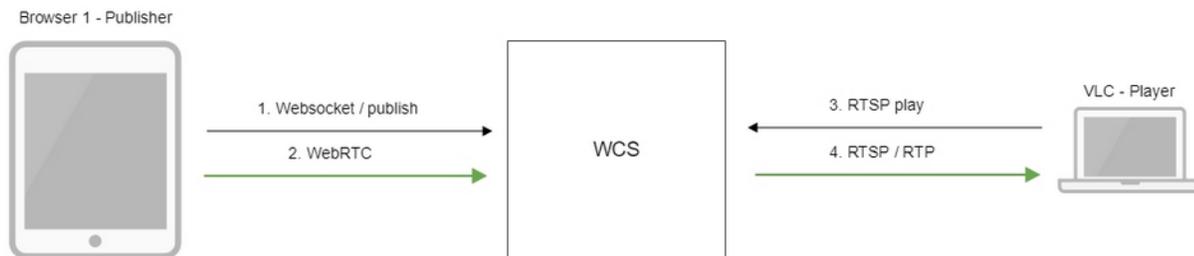
## Описание

Поток, опубликованный на WCS-сервере, можно воспроизвести по RTSP в стороннем плеере. В этом случае WCS сам выступает в качестве **RTSP-источника**.

## RTSP-кодеки

- Видео: H.264, VP8, H265 (начиная со сборки [5.2.1577](#))
- Аудио: AAC, G.711, Speex

## Схема работы



1. Браузер устанавливает соединение с сервером по Websocket
2. Браузер захватывает камеру и микрофон и отправляет на сервер WebRTC-поток
3. Проигрыватель VLC устанавливает соединение с сервером по RTSP
4. Проигрыватель VLC получает поток с сервера и воспроизводит его

## Краткое руководство по тестированию

### Трансляция видеопотока на сервер и воспроизведение его по RTSP в программном плеере

1. Для теста используем:

- демо-сервер [demo.flashphoner.com](http://demo.flashphoner.com);
- веб-приложение [Two Way Streaming](#) для публикации потока;
- проигрыватель VLC для воспроизведения потока.

2. Откройте веб-приложение Two Way Streaming. Нажмите Connect, затем Publish. Скопируйте идентификатор потока:

# Two-way Streaming

Local



528e

Stop

Player



528e

Play

Available

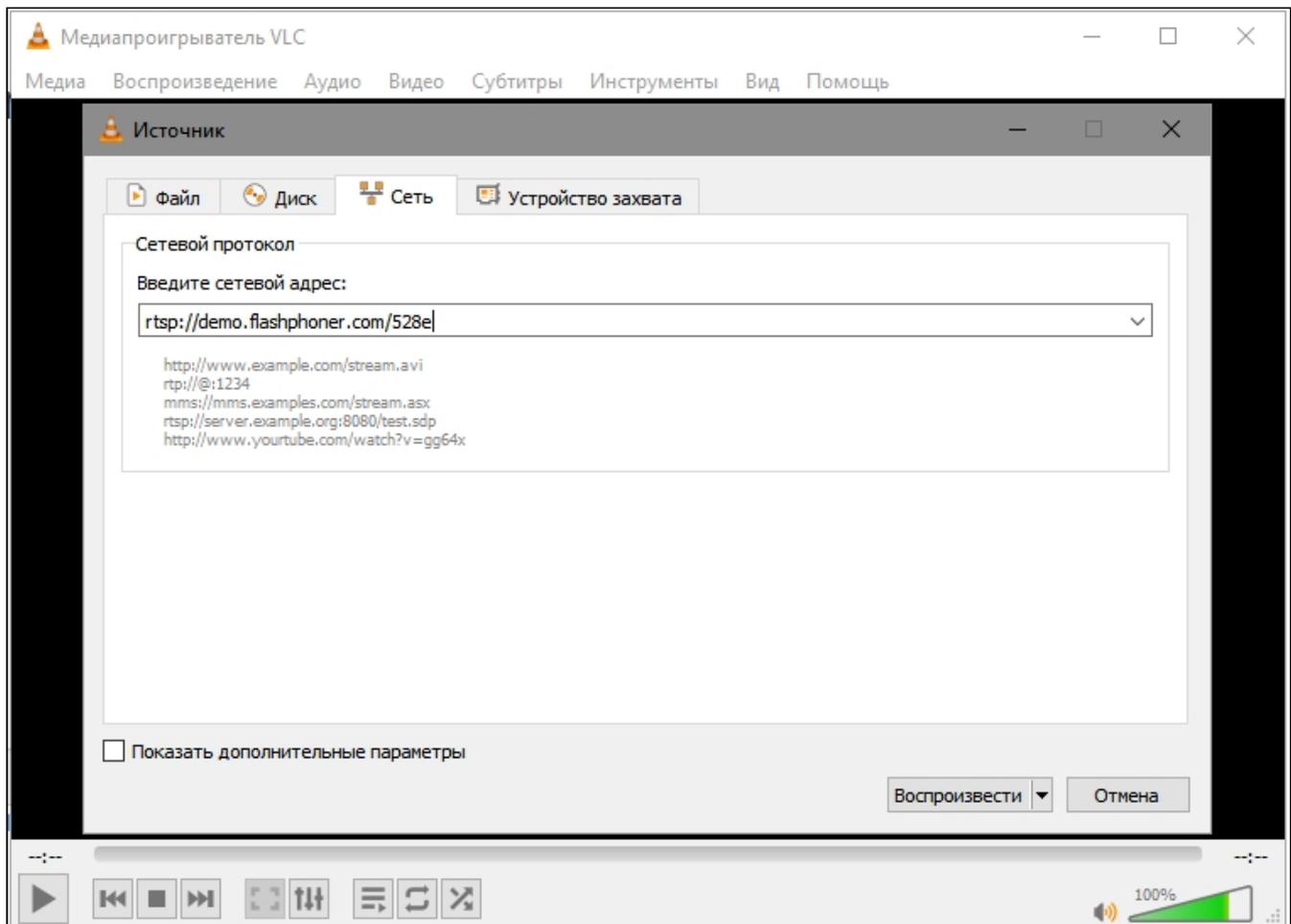
PUBLISHING

wss://demo.flashphoner.com:8443

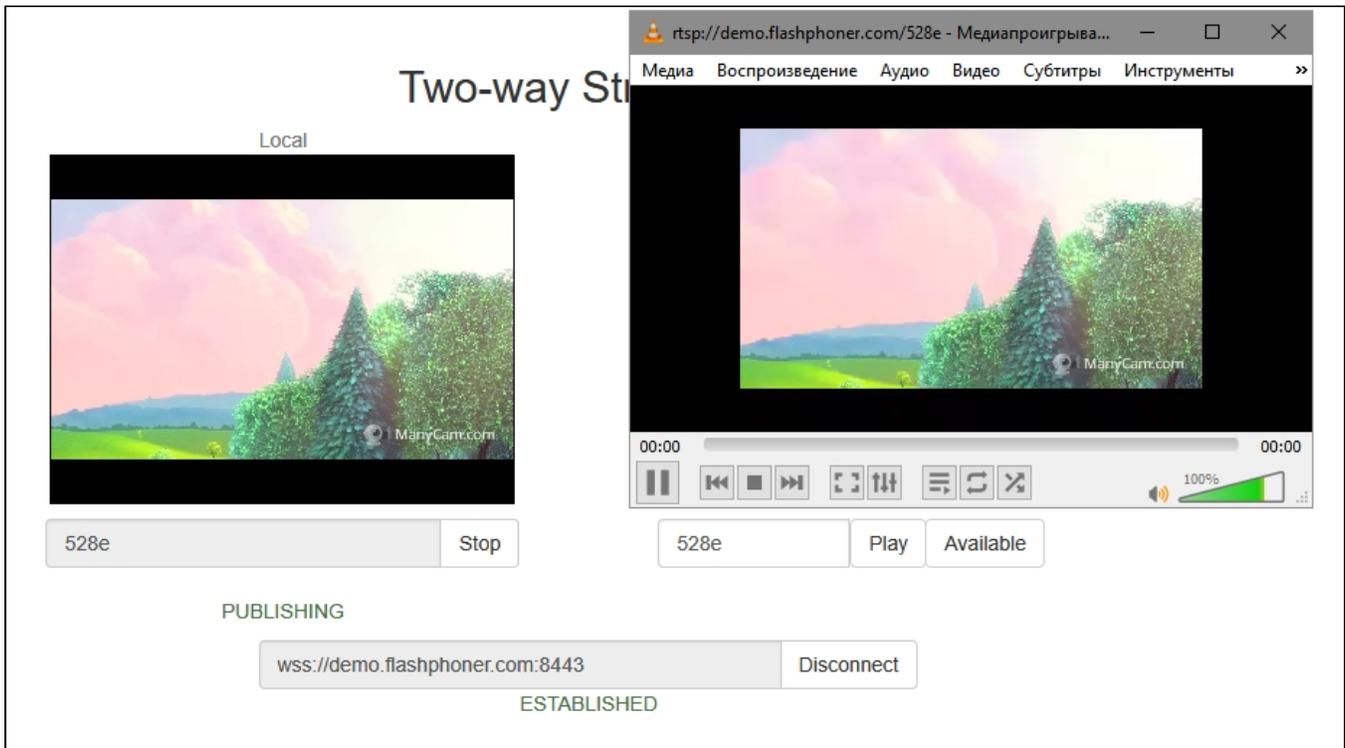
Disconnect

ESTABLISHED

3. Запустите VLC, выберите пункт меню "Медиа - Открыть URL". Введите URL WCS-сервера с указанием идентификатора потока, в данном примере  
rtsp://demo.flashphoner.com/528e:

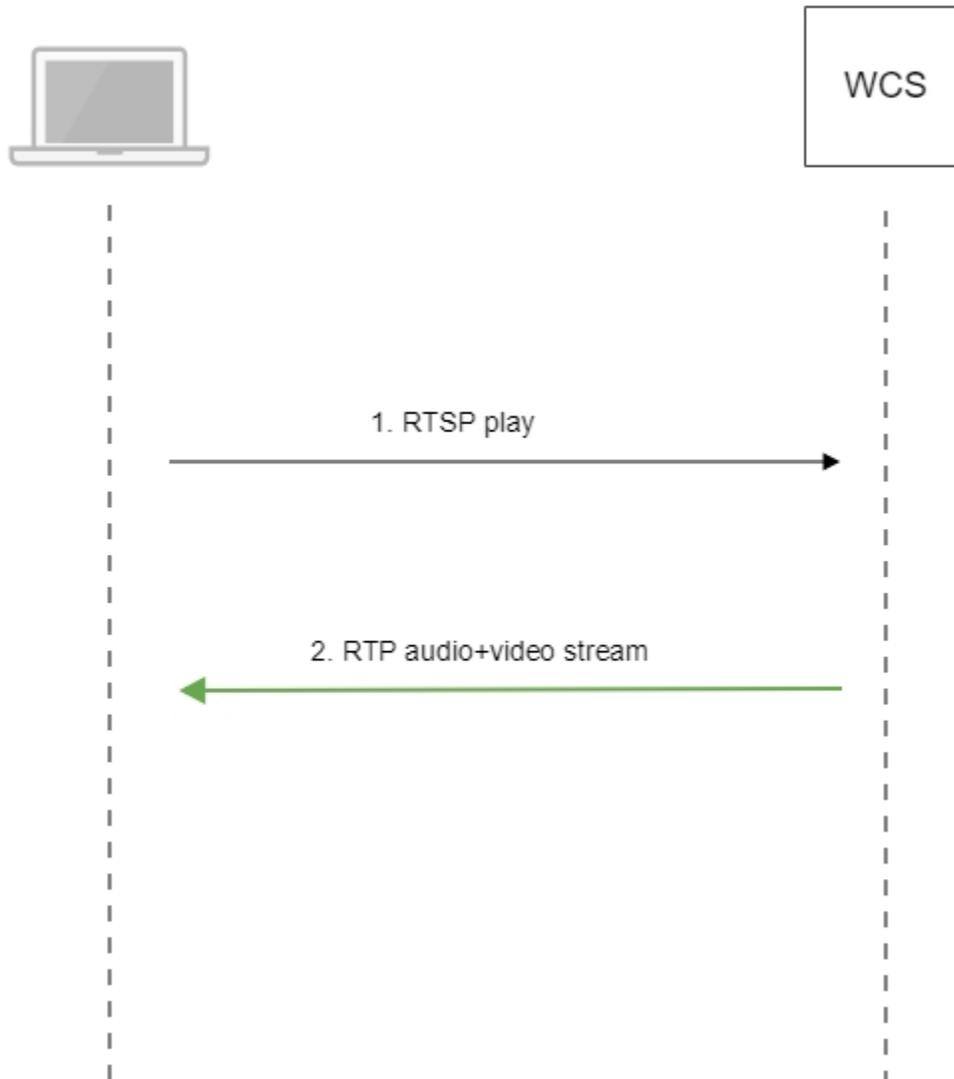


4. Нажмите кнопку "Воспроизвести". Проигрыватель начнет воспроизведение потока:



## Последовательность выполнения операций (Call flow)

Ниже описана последовательность вызовов при воспроизведении потока по RTSP в программном плеере.



1. Программный проигрыватель устанавливает соединение с WCS-сервером по RTSP
2. Программный проигрыватель получает от WCS медиапоток.

## Настройка RTSP сервера

По умолчанию, для обслуживания RTSP клиентов используется TCP порт 554. Это значение задается настройкой в файле [flashphoner.properties](#)

```
rtsp.port=554
```

Начиная со сборки [5.2.801](#), WCS запускается от пользователя flashphoner. В связи с этим RTSP сервер может не запуститься из-за того, что непривилегированным пользователям недоступны TCP порты в диапазоне 0-1000. В этом случае необходимо изменить RTSP порт, например

```
rtsp.port=5554
```

# Аутентификация воспроизведения RTSP при помощи REST hook

При необходимости, может быть настроена аутентификация воспроизведения RTSP при помощи [REST hook](#). Для этого, в файле [flashphoner.properties](#) необходимо указать следующую настройку:

```
rtsp_server_auth_enabled=true
```

При установке RTSP соединения, на бэкэнд сервер отправляется запрос /playRTSP в приложение defaultApp

```
URL:http://localhost:8081/apps/EchoApp/playRTSP
ОБЪЕКТ:
{
  "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://test.flashphoner.com:554/test",
  "mediaProvider" : "RTSP",
  "userAgent" : "LibVLC/3.0.4 (LIVE555 Streaming Media v2016.11.28)"
}
```

Такой запрос отсылается при использовании всех RTSP-методов, кроме OPTIONS. Если бэкэнд-сервер возвращает 200 OK, WCS сервер разрешает выполнение RTSP метода и воспроизведение RTSP потока. Если бэкэнд-сервер возвращает 403 Forbidden, WCS сервер разрывает RTSP-соединение с клиентом.

Таким образом, RTSP клиент может быть аутентифицирован по URL RTSP потока, User-Agent, IP адресу и порту клиента и сервера.

## Использование собственного ключа доступа и собственного приложения на бэкэнде для аутентификации

В сборке [5.2.1008](#) добавлена возможность указать собственный ключ доступа (токен) для аутентификации в RTSP URL, например

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

Запрос /playRTSP, отправленный в этом случае к бэкэнд-приложению defaultApp, будет выглядеть так

```
{
  "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"
  }
}
```

Имя параметра RTSP URL задается, как и для HLS, настройкой

```
client_acl_property_name=aclAuth
```

Также возможно указать собственный ключ бэкэнд приложения

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

В этом случае запрос /playRTSP будет отправлен в указанное приложение (customAppKey в примере выше).

## Настройка параметров проигрывания RTSP

Для настройки параметров воспроизведения RTSP, например, для изменения кодека звука или видео, используется файл SDP `rtsp_server.sdp`. Обратите внимание, что в файле SDP обязательно должен быть указан IP-адрес WCS-сервера.

### Проигрывание H265 без транскодинга

В сборке [5.2.1577](#) добавлена возможность проигрывания MPEG-TS H265 потока по RTSP. Для этого в файле `rtsp_server.sdp` должен быть указан кодек H265:

```
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
```



Потоки, опубликованные в кодеках H264, VP8 или MPV, невозможно проиграть как H265! Используйте этот способ только для проигрывания MPEG-TS H265 потоков

### Поддержка динамического определения кодеков

В сборке [5.2.1592](#) по умолчанию используется динамическая настройка кодеков для воспроизведения RTSP

```
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
m=video 0 RTP/AVP
```

В этом случае, если поток на сервере опубликован в кодеке H264, VP8 или H265, и RTSP-клиент поддерживает этот кодек, то по RTSP этот поток будет играть без транскодирования. Аналогичным образом обрабатывается и аудио.

### Поддержка interleaved режима

До сборки [5.2.1609](#), WCS поддерживал только interleaved режим, при котором и RTSP сигналинг, и RTP трафик идут по TCP, поэтому ряд плееров (в том числе VLC) с настройками по умолчанию не могли проигрывать RTSP с WCS. Начиная со сборки [5.2.1609](#), поддерживается и non-interleaved режим, при котором RTSP сигналинг идет по TCP, а RTP трафик по UDP. Отметим, что non-interleaved режим менее устойчив к потерям пакетов.

## Известные проблемы

1. При проигрывании потоков высокого разрешения по RTSP через UDP могут наблюдаться потери кадров и артефакты

Симптомы: при проигрывании HD потока по RTSP картинка содержит артефакты, в логге плеера видны потери кадров

Решение: переключить плеер в interleaved режим, например, для VLC установить в разделе настроек Input/Codecs переключатель Live 555 stream transport в положение RTP over RTSP (TCP)

2. При проигрывании WebRTC потока как RTSP могут наблюдаться фризы, если плеер не получает ключевой кадр

Симптомы: фризы при проигрывании WebRTC потока в VLC как RTSP

Решение: включить настройку в файле [flashphoner.properties](#)

```
periodic_fir_request=true
```

3. При проигрывании потока как RTSP в VLC под Windows могут некорректно отображаться параметры дискретизации звука, битрейта аудио из-за [буга VLC](#).