В браузере по WebRTC

Описание

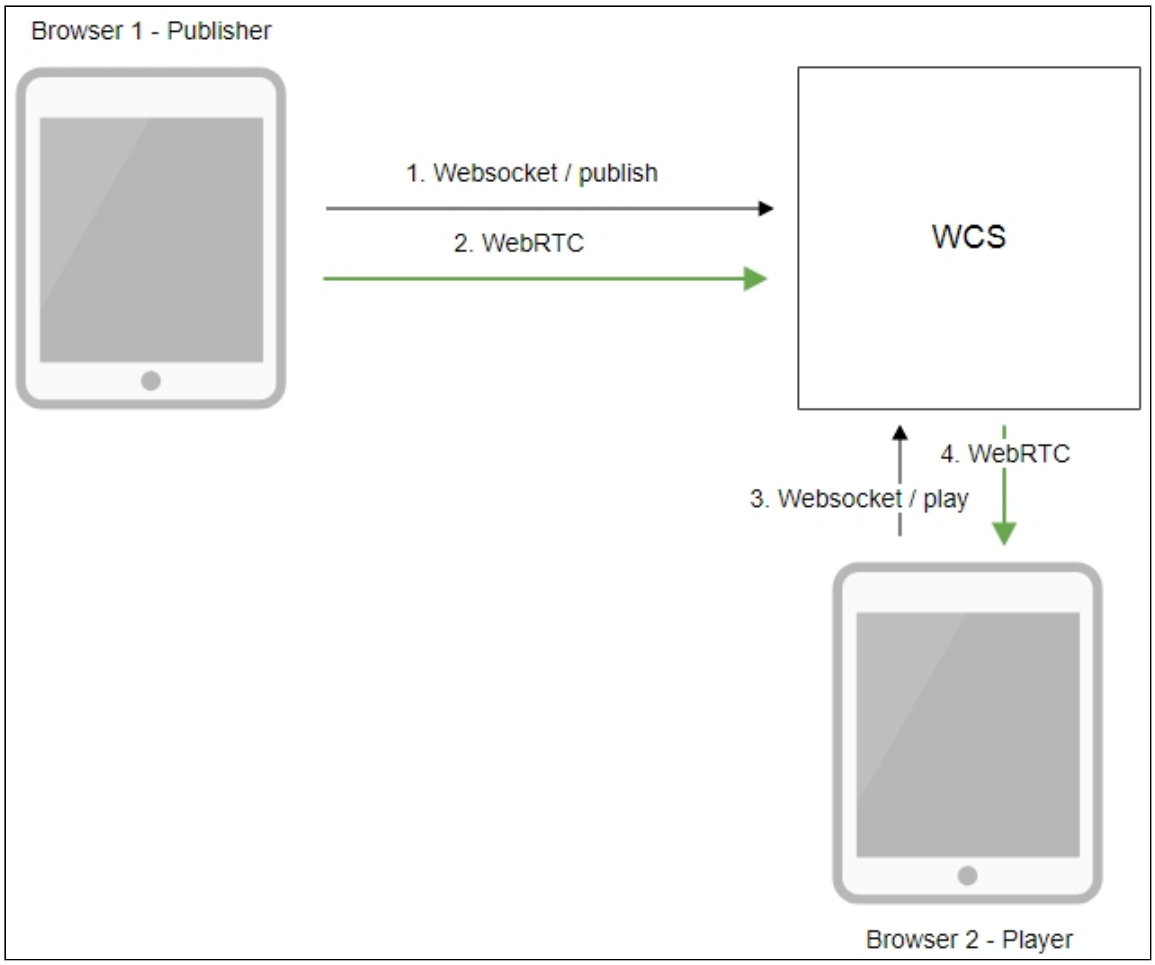
Поддерживаемые платформы и браузеры

	Chrome	**Firefox**	**Safari**	**Edge**
Windows	✅	✅	❌	✅
Mac OS	✅	✅	✅	✅
Android	✅	✅	❌	✅
iOS	✅	✅	✅	✅

Поддерживаемые кодеки

- Video: H.264, VP8
- Audio: Opus, PCMA, PCMU, G722, G729

Схема работы



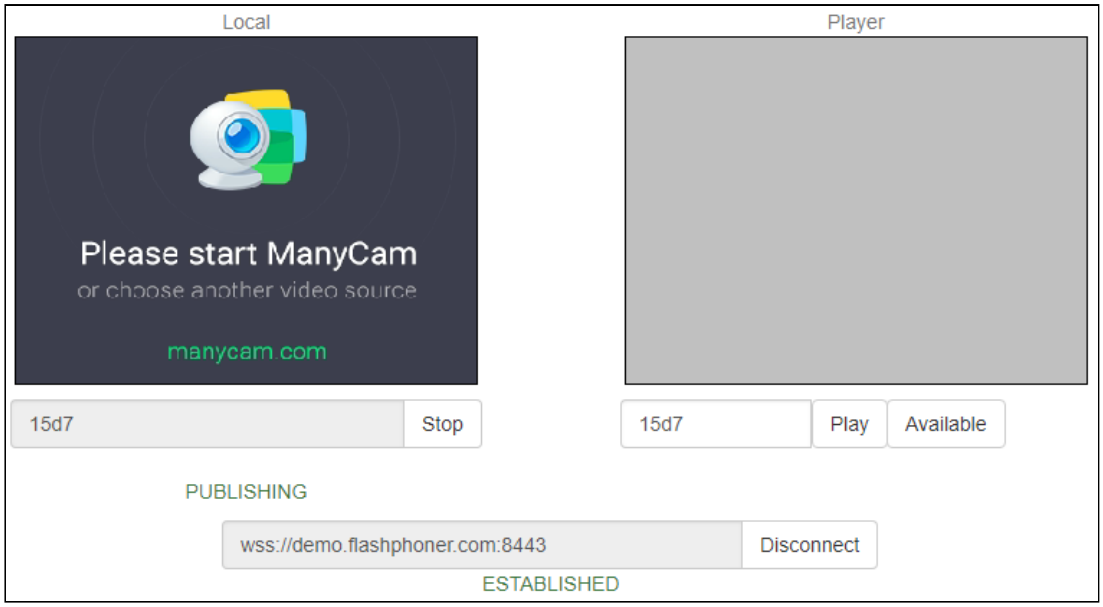
1. Браузер соединяется с сервером по протоколу Websocket и отправляет команду `publishStream`.
2. Браузер захватывает микрофон и камеру и отправляет WebRTC поток на сервер.
3. Второй браузер устанавливает соединение также по Websocket и отправляет команду `playStream`.
4. Второй браузер получает WebRTC поток и воспроизводит этот поток на странице.

Краткое руководство по тестированию

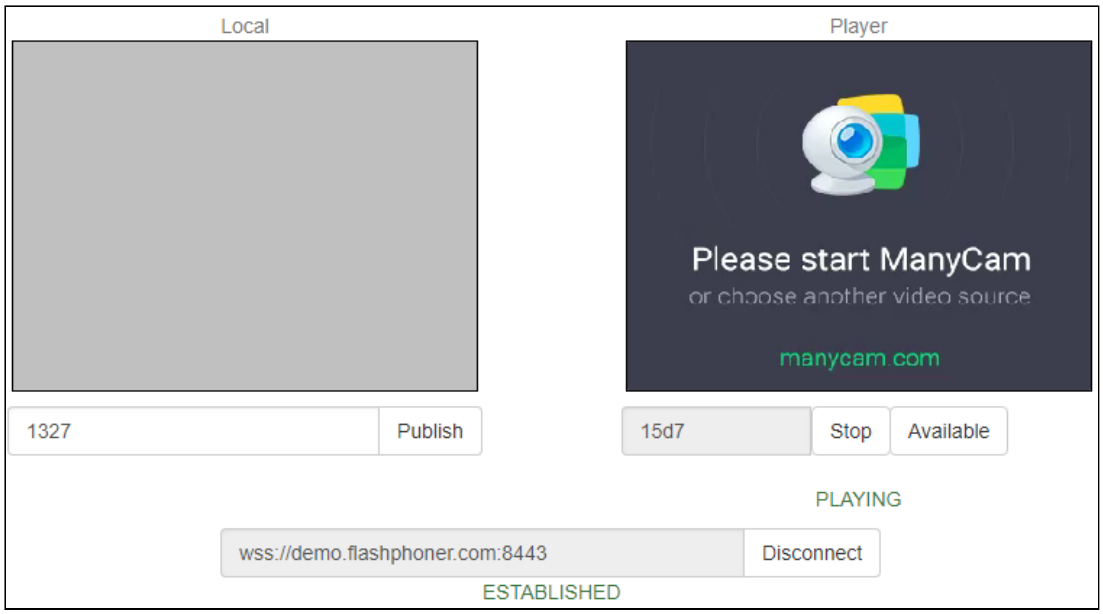
1. Для теста используем демо-сервер `demo.flashphoner.com` и веб-приложение Two Way Streaming `https://demo.flashphoner.com/client2/examples/demo/streaming/two_way_streaming/two_way_streaming.html`

2. Установите соединение с сервером по кнопке `Connect`

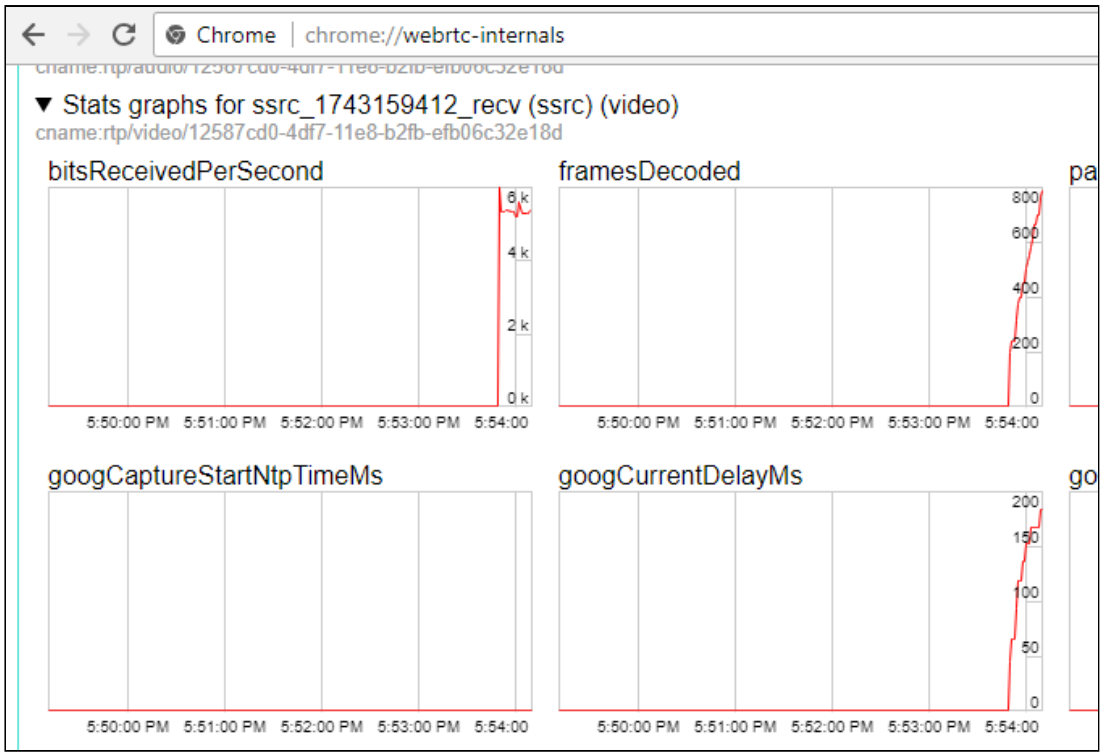
3. Нажмите `Publish`. Браузер захватывает камеру и отправляет поток на сервер



4. Откройте Two Way Streaming в отдельном окне, нажмите `Connect` и укажите идентификатор потока, затем нажмите `Play`



5. Графики воспроизведения `chrome://webrtc-internals`

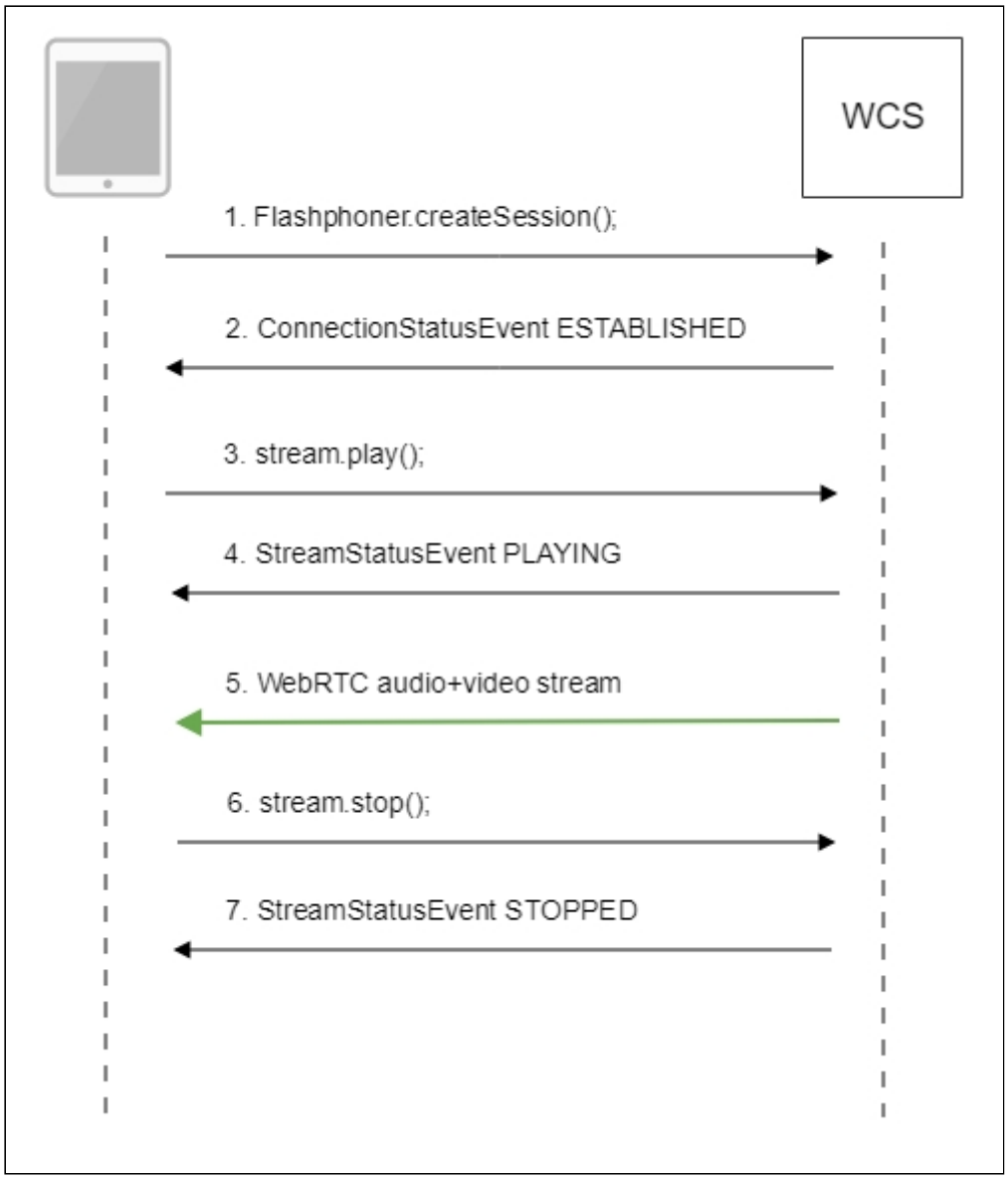


Последовательность выполнения операций

Ниже описана последовательность вызовов при использовании примера Two Way Streaming для воспроизведения потока

two_way_streaming.html

two_way_streaming.js



1. Установка соединения с сервером `Flashphoner.createSession()` code

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
  setStatus("#connectStatus", session.status());
  onConnected(session);
}).on(SESSION_STATUS.DISCONNECTED, function () {
  setStatus("#connectStatus", SESSION_STATUS.DISCONNECTED);
  onDisconnected();
}).on(SESSION_STATUS.FAILED, function () {
  setStatus("#connectStatus", SESSION_STATUS.FAILED);
  onDisconnected();
});
```

2. Получение от сервера события, подтверждающего успешное соединение `SESSION_STATUS.ESTABLISHED` code

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
  setStatus("#connectStatus", session.status());
  onConnected(session);
}).on(SESSION_STATUS.DISCONNECTED, function () {
  ...
}).on(SESSION_STATUS.FAILED, function () {
  ...
});
```

3. Воспроизведение потока
   `Stream.play()` code

```
session.createStream({
  name: streamName,
  display: remoteVideo
  ...
}).play();
```

4. Получение от сервера события, подтверждающего успешное воспроизведение потока
   `STREAM_STATUS.PLAYING` code

```
session.createStream({
  name: streamName,
  display: remoteVideo
}).on(STREAM_STATUS.PENDING, function(stream) {
  ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
  setStatus("#playStatus", stream.status());
  onPlaying(stream);
}).on(STREAM_STATUS.STOPPED, function () {
  ...
}).on(STREAM_STATUS.FAILED, function (stream) {
  ...
}).play();
```

5. Прием аудио-видео потока по WebRTC
6. Остановка воспроизведения потока
   `Stream.stop()` code

```
function onPlaying(stream) {
    $("#playBtn").text("Stop").off('click').click(function () {
        $(this).prop('disabled', true);
        stream.stop();
    }).prop('disabled', false);
    $("#playInfo").text("");
}
```

7. Получение от сервера события, подтверждающего остановку воспроизведения потока

`STREAM_STATUS.STOPPED` code

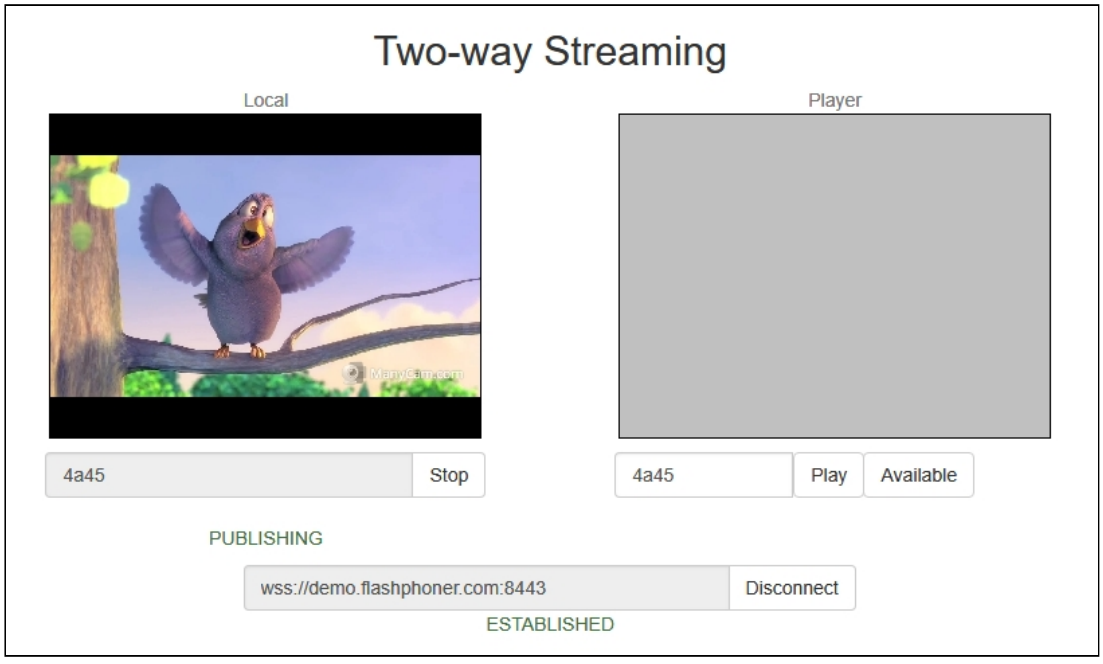
```
session.createStream({
  name: streamName,
  display: remoteVideo
}).on(STREAM_STATUS.PENDING, function(stream) {
  ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
  ...
}).on(STREAM_STATUS.STOPPED, function () {
  setStatus("#playStatus", STREAM_STATUS.STOPPED);
  onStopped();
}).on(STREAM_STATUS.FAILED, function (stream) {
  ...
}).play();
```

Воспроизведение двух и более потоков на одной странице

WCS предоставляет возможность воспроизведения двух и более потоков на одной странице. С точки зрения схемы работы и последовательности выполнения операций воспроизведение любого числа потоков не отличается от воспроизведения одного.

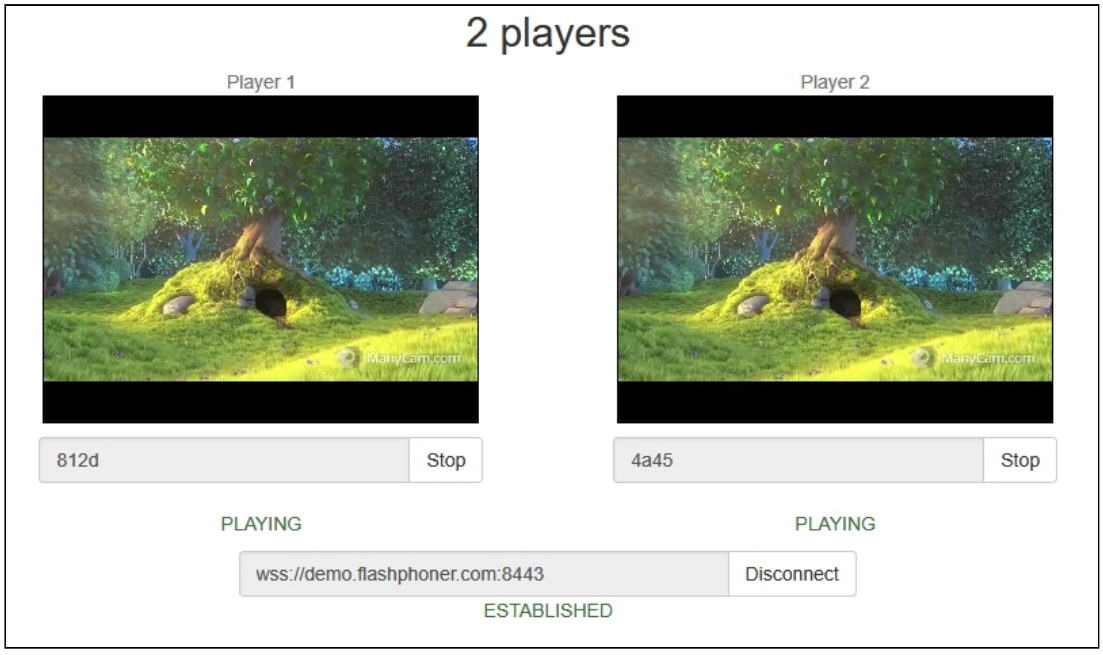
1. Для теста используем:
2. демо-сервер `demo.flashphoner.com`;
3. веб-приложение Two Way Streaming для публикации потоков
4. веб-приложение 2 Players для воспроизведения потоков
5. Откройте веб-приложение Two Way Streaming, нажмите `Connect`, затем `Publish`. Скопируйте идентификатор первого потока из окна `Local`:

6. В другой вкладке откройте веб-приложение Two Way Streaming, нажмите `Connect`, затем `Publish`. Скопируйте идентификатор второго потока из окна `Local`:



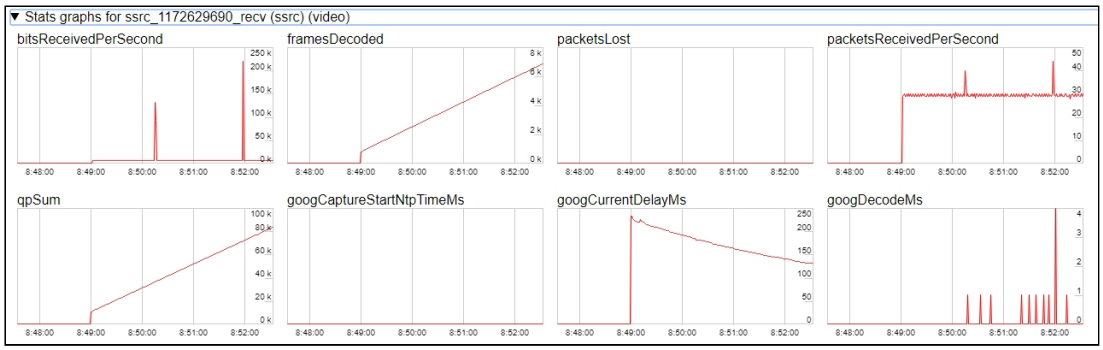
7. Откройте веб-приложение 2 Players, укажите идентификаторы первого (слева) и второго (справа) потоков:

8. Нажмите `Play` под правым и левым окнами плеера:



9. Графики `chrome://webrtc-internals` для первого потока:

10. Графики `chrome://webrtc-internals` для второго потока:



Максимальное количество потоков, которые можно играть на одной странице

Максимальное количество потоков, которое можно воспроизвести на одной странице с приемлемым качеством, зависит от следующих параметров:

- параметры одного потока (разрешение и битрейт)
- пропускная способность канала от сервера до клиента
- используемый транспорт (UDP или TCP)
- производительность клиентского устройства

Например, для потока 1920x1080 с битрейтом 2 Мбит/с, с использованием TCP транспорта для канала пропускной способностью 30-35 Мбит/с экспериментально получены следующие максимально возможные значения:

- ПК на базе Intel Core i5 8 gen и новее, от 8 Гб RAM: до 15 потоков аудио+видео, или 6 потоков аудио+видео и 14 потоков только с аудио
- Флагманское Android/iOS устройство 2018 года и новее (Samsung S серии, Apple iPhone Pro): до 15 потоков аудио+видео, или 6 потоков аудио+видео и 14 потоков только с аудио
- Устройство среднего и ниже класса или устаревшее Android/iOS устройство (Nokia 5, Apple iPhone 7): до 6 потоков аудио+видео, или только с аудио

Таким образом, для потока 1920x1080 с битрейтом 2 Мбит/с оптимальным будет проигрывание не более чем 6 потоков на одной странице, чтобы воспроизведение работало у любых клиентов.

Рассмотрим случай вебинара: один поток 1920x1080 с битрейтом 2 Мбит/с и несколько потоков 640x360 с битрейтом 500 кбит/с. В тех же условиях:

- ПК на базе Intel Core i5 8 gen и новее, от 8 Гб RAM: до 25 потоков аудио+видео, или 6 потоков аудио+видео и 25 потоков только с аудио

- Флагманское Android/iOS устройство 2018 года и новее (Samsung S серии, Apple iPhone Pro): до 20 потоков аудио+видео, или 6 потоков аудио+видео и 25 потоков только с аудио
- Устройство среднего и ниже класса или устаревшее Android/iOS устройство 2017 года и новее: до 10 потоков аудио+видео, или 6 потоков аудио+видео и 15 потоков только с аудио

Таким образом, для вебинара с трансляцией рабочего стола и нескольких веб камер оптимальным будет проигрывание не более чем 10 потоков на одной странице, чтобы воспроизведение работало у любых клиентов.

Воспроизведение WebRTC потока в стороннем плеере

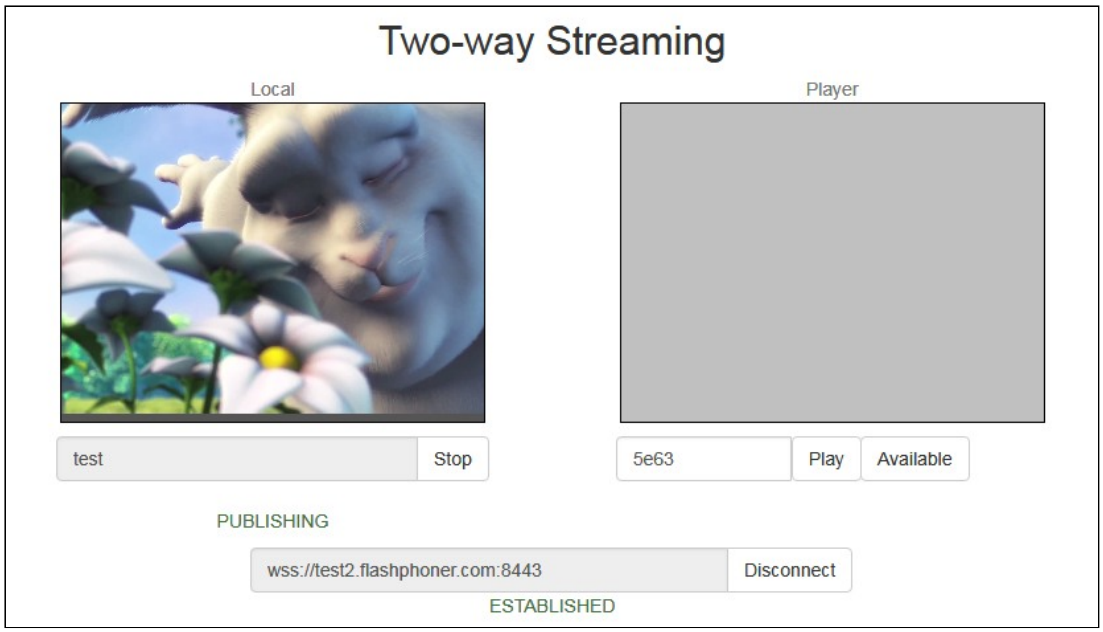
Поток, опубликованный на WCS сервере, можно проиграть по WebRTC в стороннем браузерном плеере, например, в VR плеере. Для этого элемент страницы `video`, в котором должен быть проигран поток, необходимо передать как параметр `remoteVideo` в функцию WebSDK `session.createStream()`

```
session.createStream({
    name: document.getElementById('playStream').value,
    display: display,
    remoteVideo: video
})
...
```

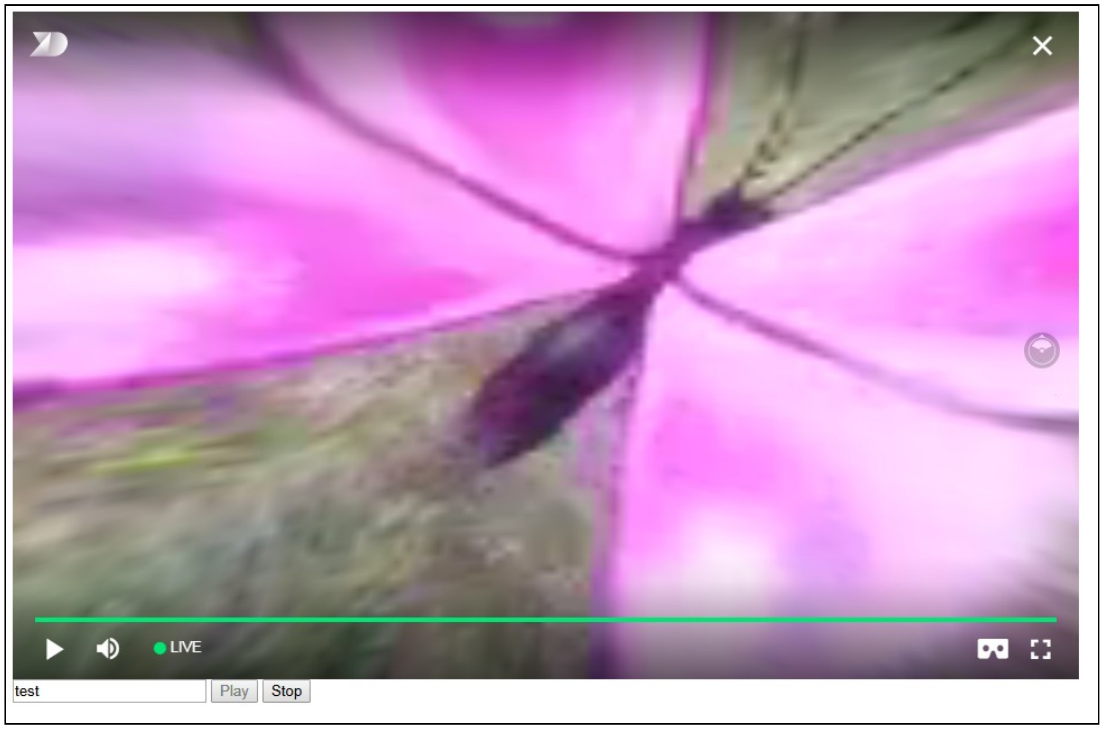
Тестирование

1. Для теста используем:
2. WCS сервер
3. Пример Two Way Streaming для публикации потока
4. Delight VR player для проигрывания потока

5. Пубо=ликация WebRTC потока на сервере



6. Воспроизведение WebRTC потока в VR плеере



Пример кода для использования стороннего плеера

1. Элемент страницы для проигрывания видео, поле ввода имени потока и кнопки для запуска и остановки проигрывания

```
<div style="width: 50%;" id="display">
   <dl8-live-video id="remoteVideo" format="STEREO_TERPON">
       <source>
```

```
    </dl8-live-video>
</div>
<input class="form-control" type="text" id="playStream"
placeholder="Stream Name">
<button id="playBtn" type="button" class="btn btn-default"
disabled>Play</button>
<button id="stopBtn" type="button" class="btn btn-default"
disabled>Stop</button>
```

2. Обработка готовности плеера к проигрыванию

```
document.addEventListener('x-dl8-evt-ready', function () {
    dl8video = document.getElementById('remoteVideo');
    $('#playBtn').prop('disabled', false).click(function() {
        playStream();
    });
});
```

3. Установка соединения с WCS сервером и создание объекта потока

```
var video = dl8video.contentElement;
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
   var session = Flashphoner.getSessions()[0];
   session.createStream({
       name: document.getElementById('playStream').value,
       display: display,
       remoteVideo: video
   }).on(STREAM_STATUS.PLAYING, function (stream) {
       ...
   }).play();
})
```

4. Запуск проигрывания в VR плеере и обработка нажатия кнопки `Stop`

```
...
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
    var session = Flashphoner.getSessions()[0];
    session.createStream({
          ...
       }).on(STREAM_STATUS.PLAYING, function (stream) {
       dl8video.start();
       $('#stopBtn').prop('disabled', false).click(function() {
           $('#playBtn').prop('disabled', false);
           $('#stopBtn').prop('disabled', true);
           stream.stop();
           dl8video.exit();
       });
    }).play();
})
```

Full custom player page code sample

Автозапуск воспроизведения

Примеры Player и Embed Player поддерживают автозапуск воспроизведения при помощи параметра

```
autoplay=true
```

например

```
https://hostname:8888/embed_player?
urlServer=wss://hostname:8443&streamName=stream1&autoplay=true&mediaProviders=We
```

Здесь

- `hostname` - имя WCS-сервера
- `stream1` - имя потока на сервере

Особенности автозапуска воспроизведения в браузерах

Chrome

В последних версиях браузера Chrome (71 и выше) была изменена политика автозапуска воспроизведения контента на веб-страницах. Теперь для запуска воспроизведения видео необходимо, чтобы пользователь совершил какое-либо действие, например, нажатие на кнопку.

Это изменение влияет на создание аудиоконтекста, который необходим, чтобы работала регулировка громкости. Согласно новой политике, создание аудиоконтекста также требует действия от пользователя.

В связи с этим, в Chrome 71, а также в других браузерах на базе Chromium, поддерживающих изменение политики автозапуска, видео при автозапуске может проигрываться без звука. Для того, чтобы включить звук, пользователь должен использовать подходящий элемент управления в окне Embed Player.

Firefox и MacOS Safari

Как и в Chrome, автозапуск воспроизведения работает без звука, для включения звука требуется действие пользователя.

iOS Safari

Автозапуск воспроизведения работает, начиная с iOS 12.2. При этом политика автозапуска требует, чтобы пользователь переместил регулятор громкости для воспроизведения звука.

В версиях iOS 12.2-12.3 звук может не начать воспроизводиться и при движении регулятора громкости. В таких случаях необходимо повторно запустить воспроизведение видео, не обновляя страницу.

При включенном Low Power Mode автозапуск воспроизведения в iOS Safari не работает.

Тонкая настройка воспроизведения звука в iOS Safari

В случае воспроизведения и последующей публикации видео на одной странице (например, видеочат) в iOS Safari уровень звука для проигрываемого потока может меняться. Избежать этого можно двумя способами:

1. Запрашивать доступ к медиаустройствам при создании сессии перед проигрыванием

```
Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED,
function (session) {
  ...
  if (Browser.isSafariWebRTC() && Browser.isiOS() &&
Flashphoner.getMediaProviders()[0] === "WebRTC") {
      Flashphoner.playFirstVideo(localVideo, true,
PRELOADER_URL).then(function () {
          Flashphoner.getMediaAccess(null, localVideo).then(function
(disp) {
          });
      });
  }
  ...
});
```

2. Через 1-1.5 секунды после получения статуса потока `PLAYING`, отключить и снова включить звук и/или видео

```
session.createStream({
  name: streamName,
  display: remoteVideo
}).on(STREAM_STATUS.PENDING, function (stream) {
  ...
}).on(STREAM_STATUS.PLAYING, function (stream) {
  setStatus("#playStatus", stream.status());
  onPlaying(stream);
  if (Browser.isSafariWebRTC() && Browser.isiOS() &&
Flashphoner.getMediaProviders()[0] === "WebRTC") {
      setTimeout(function () {
          stream.muteRemoteAudio();
```

```
            stream.unmuteRemoteAudio();
        }, 1500);
    }
    ...
}).play();
```

Воспроизведение стерео звука в браузере

Как и для публикации стерео звука в кодеке Opus, для проигрывания в браузере необходимо установить параметры кодека на стороне сервера

```
opus_formats = maxaveragebitrate=64000;stereo=1;sprop-stereo=1;
```

В этом случае браузер Firefox играет стерео звук без дополнительных настроек.

При проигрывании в браузере потока, захваченного из RTMP, RTSP или VOD источника, как правило, звук транскодируется в кодек Opus. По умолчанию, кодировщик Opus настроен на передачу речи и монофонического аудио. Для проигрывания стерео звука в браузере, необходимо повысить битрейт кодирования Opus до 60 кбит/с или выше

```
opus.encoder.bitrate=64000
```

Браузеры на основе Chrome

По умолчанию, из-за бага в движке браузер Chrome играет WebRTC поток со стерео звуком в кодеке Opus как моно. В связи с этим необходимы дополнительные настройки на стороне клиента, в зависимости от реализации.

С использованием Web SDK

В сборке Web SDK 0.5.28.2753.151 добавлена настройка для проигрывания стерео звука

```
constraints.audio.stereo=true
```

например

```
session.createStream({
  name: streamName,
  display: remoteVideo,
  constraints: {
      audio: {
          stereo: true
      }
  }
```

```
  ...
}).play();
```

С использованием Websocket API

Если в проекте используется только Websocket API, необходимо изменить параметры кодека Opus в исходящем (offer) SDP, непосредственно после его создания

```
var connection = new RTCPeerConnection(connectionConfig,
connectionConstraints);
...
connection.createOffer(constraints).then(function (offer) {
 offer.sdp = offer.sdp.replace('minptime=10', 'minptime=10;stereo=1;sprop-
stereo=1');
 connection.setLocalDescription(offer).then(function () {
     ...
 });
});
```

Дополнительная задержка при воспроизведении видеопотока

В некоторых случаев при воспроизведении видеопотока необходимо добавить заданную задержку относительно трансляции. В сборке WebSDK 0.5.28.2753.142, поставляемой вместе со сборкой WCS 5.2.708, с этой целью добавлена опция `playoutDelay`:

```
session.createStream({
  name: streamName,
  display: remoteVideo,
  playoutDelay: 10
}).on(STREAM_STATUS.PENDING, function (stream) {
  ...
}).play();
```

Задержка задается в секундах.

Данная опция поддерживается только в браузерах на основе Chromium, которые поддерживают атрибут

```
partial interface RTCRtpReceiver {
  attribute double? playoutDelayHint;
};
```

Задержка не применяется к аудио дорожкам в потоке, а также к аудио потокам без видео.

Известные проблемы

1. Возможный баг в браузере Safari на iOS приводит к фризам при воспроизведении WebRTC

Симптомы

Останавливается воспроизведение видео, звуковая дорожка при этом может продолжать играть, для восстановления требуется перезагрузка страницы либо перезапуск браузера.

Решение

a) включить транскодер на сервере, указав в файле flashphoner.properties

```
disable_streaming_proxy=true
```

b) при воспроизведении потока с iOS Safari явно указать ширину и высоту,например:

```
session.createStream({constraints:{audio:true,video:
{width:320,height:240}}}).play();
```

2. При публикации потока по RTMP и воспроизведении в браузере по WebRTC вместо аудиокодека Opus используется PCMU

Симптомы

В `chrome://webrtc-internals` отображается кодек PCMU

Решение

Отключить алгоритм избегания транскодинга (Avoid Transcoding Alhorithm) при помощи опции в файле flashphoner.properties

```
disable_rtc_avoid_transcoding_alg=true
```

3. Проигрывание аудио в RTMP потоке может останавливаться

При публикации потока при помощи Flash Streaming, воспроизведении этого потока в iOS Safari по WebRTC и одновременной публикации потока по WebRTC из Safari перестает воспроизводиться звук.

Симптомы

a) Публикация потока `stream1` из приложения Flash Streaming в браузере Chrome под Windows b) Воспроизведение потока `stream1` на iOS Safari в приложении Two Way Streaming. Звук и видео воспроизводятся нормально.
c) Публикация потока `stream2` из iOS Safari в приложении Two Way Streaming. Воспроизведение звука пропадает.
d) Остановка публикации в iOS Safari. Воспроизведение звука восстанавливается.

Решение

Отключить алгоритм избегания транскодинга (Avoid Transcoding Alhorithm) на сервере при помощи опции в файле flashphoner.properties

```
disable_rtc_avoid_transcoding_alg=true
```

4. Проигрывание RTMp потока в браузере по WebRTC может останавливаться по keep alive

При публикации потока по RTMP и отключении Keep Alive для всех протоколов воспроизведение в браузере по WebRTC останавливается по истечению WebSocket-таймаута

Симптомы

При публикации потока по RTMP воспроизведение в браузере по WebRTC останавливается без явного указания ошибки

Решение

Если Keep Alive отключен для всех протоколов при помощи настройки в файле flashphoner.properties

```
keep_alive.algorithm=NONE
```

Необходимо также отключить таймаут на чтение WebSocket настройкой

```
ws_read_socket_timeout=false
```

5. Кодек G722 не работает в браузере Edge

Симптомы

Поток со звуком G722 не воспроизводится в браузере Edge или воспроизводится без звука, с фризами

Решение

Использовать другой кодек или другой браузер. В случае, если использование другого браузера невозможно, исключить кодек G722 при помощи настройки

```
codecs_exclude_streaming=g722,telephone-event
```

6. Некоторые браузеры, основанные на Chromium, в зависимости от версии и ОС не поддерживают кодек H264

Симптомы

Не работает публикация, не работает воспроизведение частично (только звук) или полностью при трансляции WebRTC потока H264

Решение

Разрешить поддержку vp8 на стороне сервера

```
codecs=opus,...,h264,vp8,...
```

исключить H264 для трансляции или воспроизведения на стороне клиента

```
publishStream = session.createStream({
    ...
    stripCodecs: "h264,H264"
}).on(STREAM_STATUS.PUBLISHING, function (publishStream) {
    ...
});
publishStream.publish();
```

Attention

При трансляции H264 потока и воспроизведении его как VP8 на сервере

включается транскодинг.

7. Включение Flash в Chrome может приводить к ошибке Cross origin content

Если в настройках браузера Chrome 71 и выше для сайта разрешен Flash, при воспроизведении по WebRTC в консоль браузера может выводиться ошибка `Cross-origin content must have visible size large than 400 x 300 pixels, or it will be blocked`

Симптомы

При воспроизведении по WebRTC в консоль браузера Chrome выводится сообщение ёCross-origin content must have visible size large than 400 x 300 pixels, or it will be blockedё, при этом воспроизведение работает

Решение

Использовать WebSDK без поддержки Flash (по умолчанию поддержка отключена в последних сборках)

```
flashphoner-no-flash.js
```

8. При большом количестве подписчиков наблюдаются задержки в воспроизводимом потоке

Симптомы

При большом количестве подписчиков (более 200 для потока 720p) наблюдаются задержки и фризы видео, аудио продолжает играть

Решение

Включить многопоточную отсылку кадров подписчикам

```
streaming_distributor_video_proxy_pool_enabled=true
```

Эта настройка действует только на потоки, которые не транскодируются на данном сервере

9. При проигрывании потока в iOS Safari по умолчанию звук идет в голосовой динамик

Симптомы

При проигрывании WebRTC потока, например, при входе в чат-комнату с iOS устройства тихий звук

Решение

Отключить и снова включить звук при старте проигрывания потока, например

```
stream = session.createStream(options).on(STREAM_STATUS.PLAYING, function
(stream) {
    stream.muteRemoteAudio();
    stream.unmuteRemoteAudio();
}).play();
```

10. При проигрывании потока в iOS Safari резко возрастает нагрузка на сервер, при условии использования JDK 11

Симптомы

При подключении подписчика с iOS Safari резко возрастает нагрузка на процессор сервера

Решение

Обновить JDK до одной из рекомендованных версий: 8, 12, 14.

11. При воспроизведении двух и более потоков на одной странице вбраузере Chrome на некоторых устройствах Xiaomi с MIUI 12 картинка в первом потоке подергивается

Симптомы

При воспроизведении двух потоков на одной странице в примере 2 Players картинка первого потока подергивается, мелькает изображение второго потока

Решение

a) использовать на устройстве Xioami MIUI 11

b) использовать микшер для проигрывания двух и более потоков на одной странице