

Публикация и воспроизведение потока по WebRTC через TCP

- [Описание](#)
 - [Поддерживаемые платформы и браузеры](#)
 - [Схема работы](#)
- [Последовательность выполнения операций \(Call Flow\)](#)
- [Настройка](#)
 - [Управление размерами буферов на прием и передачу](#)
 - [Использование портов](#)
- [Краткое руководство по тестированию](#)
- [Известные проблемы](#)

Описание

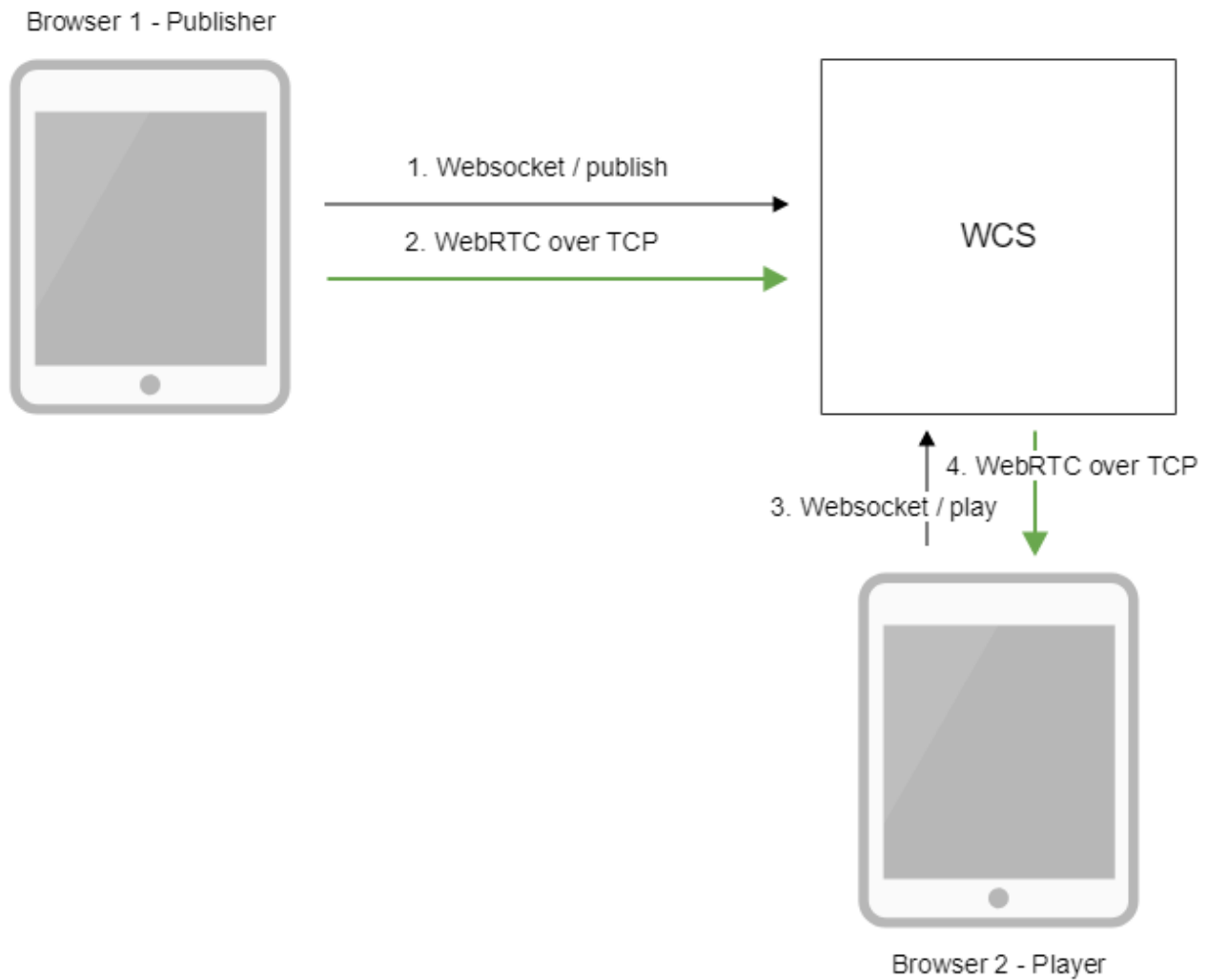
Как правило, для передачи WebRTC медиаданных на транспортном уровне используется UDP. С одной стороны, это позволяет снизить задержки при передаче данных. С другой стороны, в результате потерь пакетов, качество высокобитрейтных FullHD и 4K трансляций снижается даже на относительно хороших каналах.

Если необходимо обеспечить качество WebRTC-трансляций, WCS позволяет использовать TCP на транспортном уровне в соответствии с RFC [4571](#) и [6544](#).

Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari 11	Edge
Windows	+	+		+
Mac OS	+	+	+	
Android	+	+		
iOS	-	-	+	

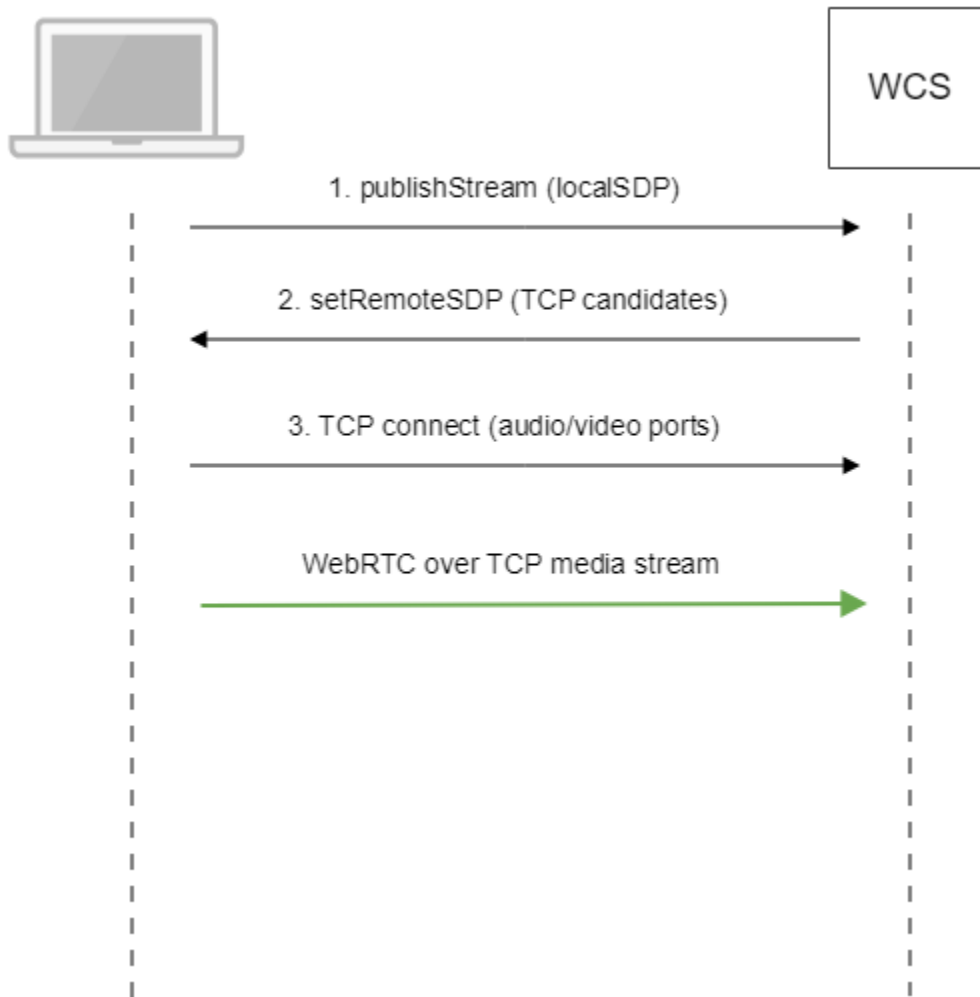
Схема работы



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

Последовательность выполнения операций (Call Flow)

Последовательность выполнения операций отличается от [публикации потока по WebRTC](#) в части установки WebRTC соединения:



1. Клиент отправляет серверу предложение локального SDP по Websocket.

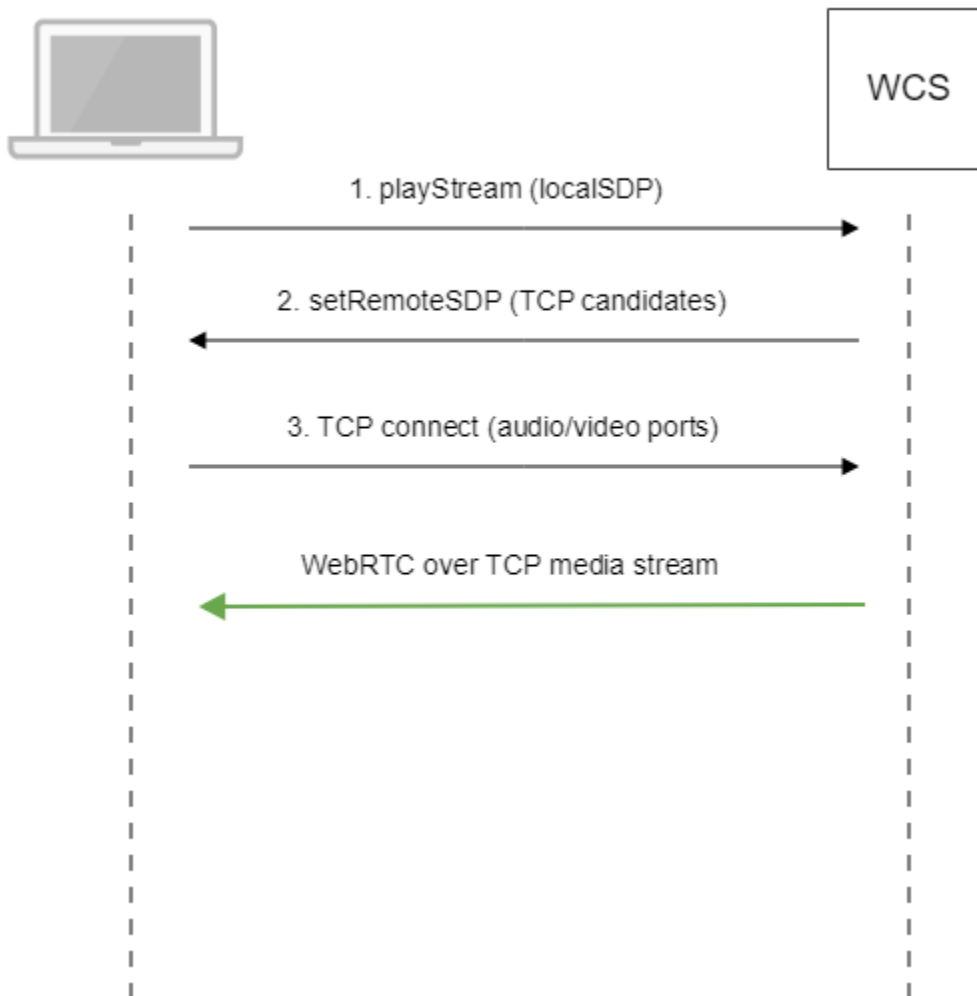
2. Клиент получает SDP от сервера. В SDP указываются TCP ICE-кандидаты:

```
v=0
o=Flashphoner 0 1545364895231 IN IP4 192.168.1.5
s=Flashphoner/1.0
c=IN IP4 192.168.1.5
t=0 0
m=audio 31038 RTP/SAVPF 111 8 9
c=IN IP4 192.168.1.5
...
a=candidate:1 1 tcp 2130706431 192.168.1.5 31038 typ host tcptype passive
a=candidate:1 2 tcp 2130706431 192.168.1.5 31038 typ host tcptype passive
a=end-of-candidates
...
m=video 31040 RTP/SAVPF 100 127 102 125 96
c=IN IP4 192.168.1.5
...
a=candidate:1 1 tcp 2130706431 192.168.1.5 31040 typ host tcptype passive
a=candidate:1 2 tcp 2130706431 192.168.1.5 31040 typ host tcptype passive
a=end-of-candidates
a=rtcp-mux
a=rtcp:31040 IN IP4 192.168.1.5
a=sendonly
a=ssrc:564293803 cname:rtp/video/1b951110-04d5-11e9-a8b5-19c6b1a7cdbb
```

Здесь 192.168.1.5 - IP адрес WCS сервера

3. Клиент устанавливает TCP соединение на порты для аудио и видео данных, указанные в SDP, и начинает передачу медиаданных.

Аналогично, последовательность выполнения операций при воспроизведении потока будет следующей:



1. Клиент отправляет серверу предложение локального SDP по Websocket.

2. Клиент получает SDP от сервера. В SDP указываются TCP ICE-кандидаты.

3. Клиент устанавливает TCP соединение на порты для аудио и видео данных, указанные в SDP, и начинает прием медиаданных.

Настройка

Использование WebRTC через TCP включается настройкой в файле [flashphoner.properties](#)

```
ice_tcp_transport=true
```

Управление размерами буферов на прием и передачу

Размеры буферов для приема и отправки данных настраиваются при помощи следующих параметров:

```
ice_tcp_send_buffer_size=1048576
ice_tcp_receive_buffer_size=1048576
```

По умолчанию, размеры буферов установлены в 1 Мб.

Использование портов

Для WebRTC через TCP используются TCP порты с номерами из диапазона, выделенного для WebRTC медиа портов

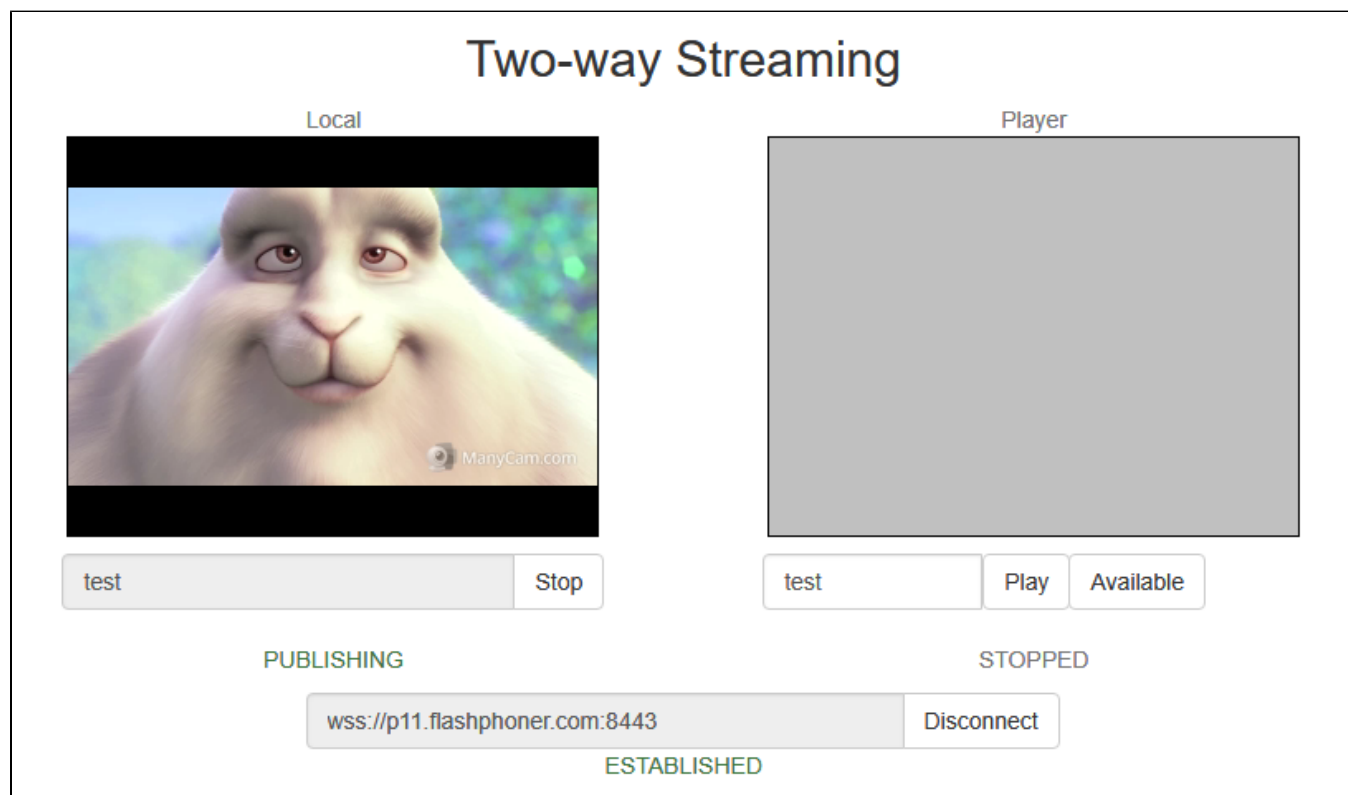
```
media_port_from    =31001
media_port_to      =32000
```

Краткое руководство по тестированию

1. Для теста используем:

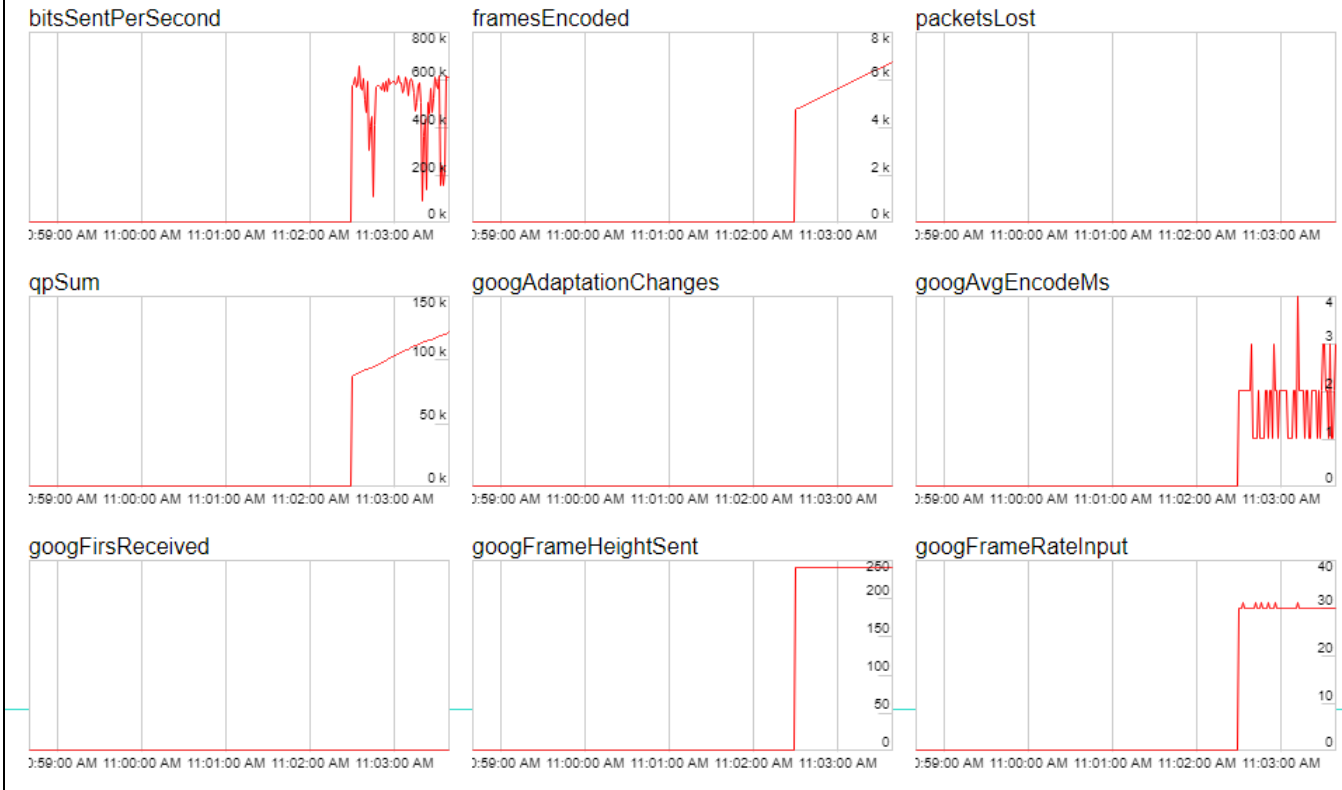
- WCS сервер
- веб-приложение [Two Way Streaming](#) для публикации и воспроизведения потока в браузере Chrome

2. Откройте веб приложение Two Way Streaming, нажмите Connect, введите имя потока test и нажмите Publish. начнется публикация потока



3. Чтобы убедиться, что поток отправляется на сервер, откройте `chrome://webrtc-internals`

▼ Stats graphs for ssrc_3239846956_send (ssrc) (video)



4. В окне Player введите имя потока test и нажмите Play. Начнется воспроизведение публикуемого потока.

Two-way Streaming

Local

ManyCam.com

test

Stop

PUBLISHING

Player

ManyCam.com

test

Stop

Available

PLAYING

wss://p11.flashphoner.com:8443

Disconnect

ESTABLISHED

5. Графики воспроизведения



6. Чтобы убедиться, что установлено TCP соединение, выполните на сервере команду

```
netstat -np | grep ESTABLISHED
```

Результатом выполнения будут следующие строки

```
# websocket session
tcp      0      0 192.168.1.5:8443      192.168.1.100:60289   ESTABLISHED 7459/java
# publishing stream
tcp      0      0 192.168.1.5:31030     192.168.1.100:60305   ESTABLISHED 7459/java
tcp      0      0 192.168.1.5:31032     192.168.1.100:60307   ESTABLISHED 7459/java
# playing stream
tcp      0    112 192.168.1.5:31038     192.168.1.100:60515   ESTABLISHED 7459/java
tcp      0    817 192.168.1.5:31040     192.168.1.100:60517   ESTABLISHED 7459/java
```

Здесь

- 192.168.1.5 - IP адрес WCS сервера
- 192.168.1.100 - IP адрес клиента

Известные проблемы

1. WebRTC соединение не устанавливается в некоторых браузерах (MS Edge на Windows, Chrome на Ubuntu), если на ПК используется дополнительный сетевой интерфейс (VPN)

Симптомы: публикация и воспроизведение WebRTC через TCP не работают

Решение: отключить все дополнительные сетевые интерфейсы, кроме имеющего доступ к WCS серверу.