

Публикация MPEG-TS RTP потока

- [Описание](#)
- [Поддержка кодеков](#)
- [Схема работы](#)
- [Тестирование](#)
- [Настройки](#)
 - [Остановка публикации при отсутствии медиаданных](#)
 - [Отключение подписчиков при остановке передачи данных от публикующего клиента](#)
- [REST API](#)
 - [REST-методы и статусы ответа](#)
 - [Параметры](#)
- [Публикация только аудио или только видео](#)
- [Публикация аудио с различными частотами дискретизации](#)
- [Возобновление публикации после остановки](#)
- [Ограничение адресов клиентов, с которых разрешена публикация](#)
- [Публикация H265](#)
- [Известные проблемы](#)

Описание

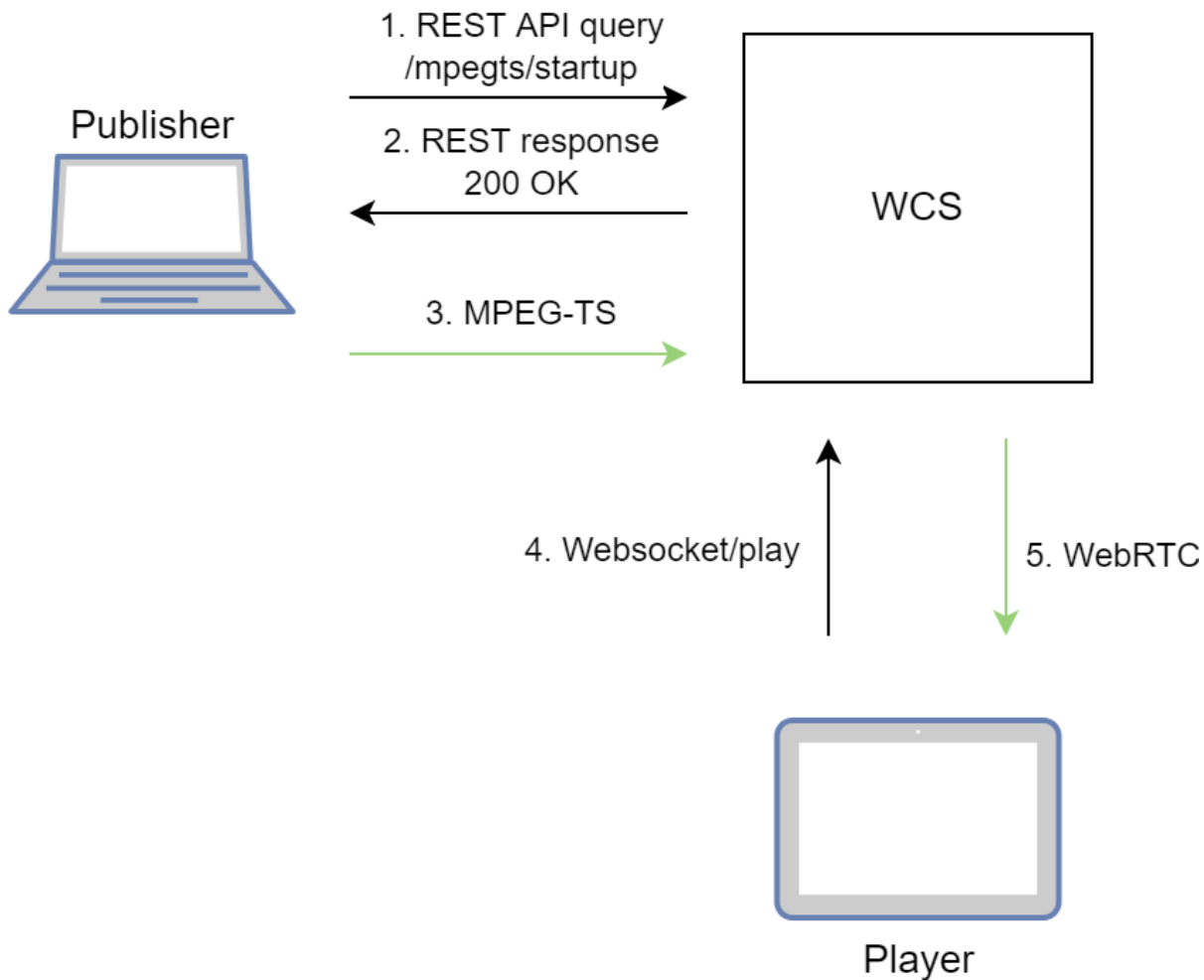
В сборке [5.2.1193](#) добавлена возможность публикации MPEG-TS RTP потока по UDP на WCS, а в сборке [5.2.1253](#) MPEG-TS поток может быть опубликован по SRT. Данный способ может быть удобен для публикации H264+AAC потока из программного или аппаратного кодировщика, поддерживающего MPEG-TS. В сборке [5.2.1577](#) добавлена возможность публикации H265+AAC потока.

Протокол [SRT](#) является более надежным по сравнению с UDP, поэтому по возможности рекомендуется использовать SRT для публикации MPEG-TS.

Поддержка кодеков

- H264
- H265, начиная со сборки [5.2.1577](#)
- AAC

Схема работы



1. Публикующий клиент отправляет REST API запрос /mpegts/startup
2. Публикующий клиент получает ответ 200 OK с URI для публикации потока
3. Поток публикуется на WCS по указанному URI
4. Браузер устанавливает соединение по Websocket и отправляет команду play.
5. Браузер получает WebRTC поток и воспроизводит этот поток на странице.

Тестирование

1. Для теста используем:

- WCS сервер
- ffmpeg для публикации MPEG-TS потока
- веб-приложение [Player](#) в браузере Chrome для воспроизведения потока

2. Отправляем запрос /mpegts/startup с указанием имени потока test

SRT:

```
curl -H "Content-Type: application/json" -X POST http://test1.flashphoner.com:8081/rest-api/mpegts/startup -d '{"localStreamName":"test","transport":"srt"}'
```

UDP:

```
curl -H "Content-Type: application/json" -X POST http://test1.flashphoner.com:8081/rest-api/mpegts/startup -d '{"localStreamName":"test","transport":"udp"}'
```

Здесь test1.flashphoner.com - адрес WCS сервера

3. Получаем от сервера ответ 200 OK

SRT:

```
{
  "localMediaSessionId": "32ec1a8e-7df4-4484-9a95-e7eddc45c508",
  "localStreamName": "test",
  "uri": "srt://test1.flashphoner.com:31014",
  "status": "CONNECTED",
  "hasAudio": false,
  "hasVideo": false,
  "record": false,
  "transport": "SRT",
  "cdn": false,
  "timeout": 90000,
  "maxTimestampDiff": 1,
  "allowedList": []
}
```

UDP:

```
{
  "localMediaSessionId": "32ec1a8e-7df4-4484-9a95-e7eddc45c508",
  "localStreamName": "test",
  "uri": "udp://test1.flashphoner.com:31014",
  "status": "CONNECTED",
  "hasAudio": false,
  "hasVideo": false,
  "record": false,
  "transport": "UDP",
  "cdn": false,
  "timeout": 90000,
  "maxTimestampDiff": 1,
  "allowedList": []
}
```

4. Публикуем MPEG-TS поток по указанному URI

SRT:

```
ffmpeg -re -i bunny360p.mp4 -c:v libx264 -c:a aac -b:a 160k -bsf:v h264_mp4toannexb -keyint_min 60 -profile:v baseline -preset veryfast -f mpegts "srt://test1.flashphoner.com:31014"
```

UDP:

```
ffmpeg -re -i bunny360p.mp4 -c:v libx264 -c:a aac -b:a 160k -bsf:v h264_mp4toannexb -keyint_min 60 -profile:v baseline -preset veryfast -f mpegts "udp://test1.flashphoner.com:31014?pkt_size=1316"
```

```

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'bunny360p.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf58.12.100
  Duration: 00:09:56.46, start: 0.000000, bitrate: 631 kb/s
  Stream #0:0[0x1](eng): Video: h264 (High) (avc1 / 0x31637661), yuv420p(progressive), 640x360, 499 kb/s, 24 fps, 24 tbr, 12288 tbn (default)
    Metadata:
      handler_name    : VideoHandler
      vendor_id       : [0][0][0][0]
  Stream #0:1[0x2](eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 128 kb/s (default)
    Metadata:
      handler_name    : SoundHandler
      vendor_id       : [0][0][0][0]
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> h264 (libx264))
  Stream #0:1 -> #0:1 (aac (native) -> aac (native))
Press [q] to stop, [?] for help
[libx264 @ 00000249853ac540] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[libx264 @ 00000249853ac540] profile Constrained Baseline, level 3.0, 4:2:0, 8-bit
Output #0, mpegts, to 'udp://95.191.130.39:31006?pkt_size=1316':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf59.16.100
  Stream #0:0(eng): Video: h264, yuv420p(progressive), 640x360, q=2-31, 24 fps, 90k tbn (default)
    Metadata:
      handler_name    : VideoHandler
      vendor_id       : [0][0][0][0]
      encoder         : Lavc59.18.100 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
  Stream #0:1(eng): Audio: aac (LC), 48000 Hz, stereo, fltp, 160 kb/s (default)
    Metadata:
      handler_name    : SoundHandler
      vendor_id       : [0][0][0][0]
      encoder         : Lavc59.18.100 aac
frame= 5553 fps= 24 q=28.0 size=   18251kB time=00:03:51.65 bitrate= 645.4kbits/s speed=   1x

```

5. Открываем веб-приложение Player. Укажите в поле "Stream" имя потока test и нажмите кнопку "Start". Начнется трансляция опубликованного потока

Player



WCS URL

wss://test1.flashphoner.com:8443

Stream

test

Volume



Full Screen



PLAYING

Stop

Настройки

Остановка публикации при отсутствии медиаданных

По умолчанию, публикация MPEG-TS потока будет остановлена на стороне сервера, если сервер не получает медиаданных в течение 90 секунд. Это время задается настройкой в миллисекундах

```
mpegts_stream_timeout=90000
```

Отключение подписчиков при остановке передачи данных от публикующего клиента

Если публикующий клиент по какой-то причине остановил передачу медиаданных, а затем возобновил (например, перезапустил ffmpeg), нарушается последовательность временных меток кадров потока. Такой поток не может быть корректно воспроизведен по WebRTC. В связи с этим, если зафиксировано нарушение последовательности временных меток для публикуемого MPEG TS потока, все подписчики принудительно отключаются, и должны заново подключиться к нему. Максимально допустимое изменение двух соседних временных меток задается настройкой в секундах

```
mpegts_max_pts_diff=1
```

REST API

REST-запрос должен быть HTTP/HTTPS POST запросом в таком виде:

- HTTP: <http://test.flashphoner.com:8081/rest-api/mpegts/startup>
- HTTPS: <https://test.flashphoner.com:8444/rest-api/mpegts/startup>

Здесь:

- test.flashphoner.com - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /mpegts/startup - используемый REST-метод

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/mpegts/startup	<pre>{ "localStreamName": "test", "transport": "srt", "hasAudio": true, "hasVideo": true}</pre>	<pre>{ "localMediaSessionId": "32ec1a8e-7df4-4484-9a95-e7eddc45c508", "localStreamName": "test", "uri": "srt://192.168.1.39:31014", "status": "CONNECTED", "hasAudio": false, "hasVideo": false, "record": false, "transport": "SRT", "cdn": false, "timeout": 90000, "maxTimestampDiff": 1, "allowedList": []}</pre>	200 - OK 409 - Conflict 500 - Internal error	Начать публикацию MPEG-TS потока
/mpegts/find	<pre>{ "localStreamName": "test", "uri": "srt://192.168.1.39:31014"}</pre>	<pre>[{ "localMediaSessionId": "32ec1a8e-7df4-4484-9a95-e7eddc45c508", "localStreamName": "test", "uri": "srt://192.168.1.39:31014", "status": "CONNECTED", "hasAudio": true, "hasVideo": true, "record": false, "timeout": 90000, "maxTimestampDiff": 90000 }]</pre>	200 – потоки найдены 404 – потоки не найдены 500 - Internal error	Найти MPEG-TS поток по заданным критериям

/mpegts /find_all		<pre>[{ "localMediaSessionId": "32ec1a8e-7df4-4484-9a95-e7eddc45c508", "localStreamName": "test", "uri": "srt://192.168.1.39:31014", "status": "PROCESSED_LOCAL", "hasAudio": false, "hasVideo": false, "record": false, "transport": "SRT", "cdn": false, "timeout": 90000, "maxTimestampDiff": 1, "allowedList": [] }]</pre>	200 – потоки найдены 404 – потоки не найдены 500 - Internal error	Найти все опубликованные MPEG-TS потоки
/mpegts /terminate	<pre>{ "localStreamName": "test" }</pre>		200 - поток завершен 404 - поток не найден 500 - Internal error	Завершить MPEG-TS поток

Параметры

Имя параметра	Описание	Пример
localStreamName	Имя, которое будет присвоено опубликованному потоку	test
transport	Используемый транспорт	srt
uri	URI для публикации потока	udp://192.168.1.39:31014
localMediaSessionId	Идентификатор медиасессии потока	32ec1a8e-7df4-4484-9a95-e7eddc45c508
status	Статус потока	CONNECTED
hasAudio	Поток содержит аудио	true
hasVideo	Поток содержит видео	true
record	Поток записывается	false
timeout	Максимальное время ожидания медиаданных, мс	90000
maxTimestampDiff	Максимально допустимое изменение метки времени, с	1
allowedList	Список адресов, с который разрешена публикация потока	["192.168.1.0/24"]

Публикация только аудио или только видео

Начиная со сборки [5.2.1253](#), можно начать публикацию только видео или только аудио потока, указав соответствующий параметр REST API запроса /mpegts/startup

- поток только с видео

```
{
  "localStreamName": "mpegts-video-only",
  "transport": "srt",
  "hasAudio": false
}
```

- поток только с аудио

```
{
  "localStreamName": "mpegts-audio-only",
  "transport": "srt",
  "hasVideo": false
}
```

Публикация аудио с различными частотами дискретизации

По умолчанию, для публикации MPEG-TS используются следующие параметры видео и аудио

```
v=0
o=- 1988962254 1988962254 IN IP4 0.0.0.0
c=IN IP4 0.0.0.0
t=0 0
a=sdplang:en
m=audio 1 RTP/AVP 102
a=rtpmap:102 mpeg4-generic/44100/2
a=sendonly
m=video 1 RTP/AVP 119
a=rtpmap:119 H264/90000
a=sendonly
```

Видео должно быть опубликовано в кодеке H264 с частотой дискретизации 90000 Гц, аудио должно быть опубликовано в кодеке AAC с частотой дискретизации 44100 Гц и двумя каналами.

При необходимости, можно указать поддержку нескольких частот дискретизации аудио, либо поддержку одноканального звука. Для этого необходимо:

1. Создать в каталоге /usr/local/FlashphonerWebCallServer/conf файл mpegts_agent.sdp

```
sudo touch /usr/local/FlashphonerWebCallServer/conf/mpegts_agent.sdp
```

2. Добавить в файл необходимое описание параметров SDP

```
sudo nano /usr/local/FlashphonerWebCallServer/conf/mpegts_agent.sdp
```

например

```
v=0
o=- 1988962254 1988962254 IN IP4 0.0.0.0
c=IN IP4 0.0.0.0
t=0 0
a=sdplang:en
m=audio 1 RTP/AVP 102 103 104
a=rtpmap:102 mpeg4-generic/44100/2
a=rtpmap:103 mpeg4-generic/48000/2
a=rtpmap:104 mpeg4-generic/32000/1
a=sendonly
m=video 1 RTP/AVP 119
a=rtpmap:119 H264/90000
a=sendonly
```

3. Установить нужные права и перезапустить WCS, чтобы применить изменения

```
sudo nano /usr/local/FlashphonerWebCallServer/bin/webcallserver set-permissions
sudo systemctl restart webcallserver
```

Возобновление публикации после остановки

Под каждую публикацию MPEG-TS выделяется отдельный UDP порт, который ждет входящего соединения (для SRT) и трафика от клиента. В целях безопасности, начиная со сборки [5.2.1299](#), если клиент остановил публикацию, поток на сервере останавливается, и повторно к тому же самому порту подключиться нельзя. Зрители в этом случае получают событие `STREAM_STATUS.FAILED`. Чтобы возобновить публикацию, должен быть использован новый REST API запрос для создания на сервере нового потока, при необходимости с тем же именем.

Ограничение адресов клиентов, с которых разрешена публикация

В сборке [5.2.1314](#) добавлена возможность задать список адресов, с которых разрешена публикация MPEG-TS по UDP, указав соответствующий параметр REST API запроса `/mpegts/startup`

```
{
  "localStreamName": "mpegts-stream",
  "transport": "udp",
  "allowedList": [
    "192.168.0.100",
    "172.16.0.1/24"
  ]
}
```

В сборке [5.2.1485](#) аналогичным образом можно ограничить список адресов, с которых разрешена публикация MPEG-TS по SRT:

```
{
  "localStreamName": "mpegts-stream",
  "transport": "srt",
  "allowedList": [
    "192.168.0.100",
    "172.16.0.1/24"
  ]
}
```

В списке могут быть как точные адреса, так и маски адресов. Если такой список содержится в запросе, то опубликовать поток можно будет только с клиентов, чьи адреса соответствуют списку.

Публикация H265

В сборке [5.2.1577](#) добавлена возможность публикации MPEG-TS потока H265+AAC. Для этого в файле `mpegts_agent.sdp` должен быть указан видео 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
m=audio 1 RTP/AVP 102
a=rtpmap:102 mpeg4-generic/48000/2
a=sendonly
m=video 1 RTP/AVP 119
a=rtpmap:119 H265/90000
a=sendonly
```

Начиная со сборки [5.2.1598](#), WCS по умолчанию поддерживает публикацию MPEG-TS потока в H264 и H265 без изменений настроек SDP.

Также H265 должен быть добавлен в список поддерживаемых кодеков

```
codecs=opus,a law,u law,g729,speex16,g722,mpeg4-generic,telephone-event,h264,vp8,flv,mpv,h265
```

и в списки исключений

```
codecs_exclude_sip=mpeg4-generic,flv,mpv,h265
codecs_exclude_sip_rtmp=opus,g729,g722,mpeg4-generic,vp8,mpv,h265
codecs_exclude_sfu=a law,u law,g729,speex16,g722,mpeg4-generic,telephone-event,flv,mpv,h265
```

Публикация H265 при помощи ffmpeg

```
ffmpeg -re -i source.mp4 -c:v libx265 -c:a aac -ar 48000 -ac 2 -b:a 160k -bsf:v hevc_mp4toannexb -keyint_min 120 -profile:v main -preset veryfast -x265-params crf=23:bframes=0 -f mpegts "srt://test.flashphoner.com:31014"
```



При проигрывании H265 потока любым способом на сервере включается транскодинг из H265 в H264 или VP8!

Известные проблемы

1. Если публикация MPEG-TS потока по UDP была остановлена на стороне сервера по REST API `/mpegts/terminate`, публикующий кодировщик продолжает отправлять медиаданные

Симптомы: при остановке публикации MPEG-TS потока на сервере ffmpeg продолжает отправлять данные по UDP

Решение: для UDP это ожидаемое поведение, поскольку самим протоколом не предусмотрены никакие оповещения отправляющей стороны о том, что порт, принимающий данные, уже закрыт. Используйте протокол SRT, в котором данный случай обрабатывается корректно, и публикующий клиент останавливается.