

# В браузере по WebRTC ABR

## Описание

В сборке WCS 5.2.1504 добавлена возможность проигрывания потока с сервера по WebRTC в браузере в нескольких качествах видео, наподобие HLS ABR. При этом браузер автоматически начинает проигрывать первое доступное качество, а клиент может переключаться между ними, при необходимости.

Начиная со сборки WCS 5.2.1663 и SFU SDK 2.0.231 поддерживается автоматическое переключение между доступными качествами ABR при изменении состояния канала. Выбор качества основывается на WebRTC статистике.

## Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari	Edge
Windows	✓	✓	✗	✓
Mac OS	✓	✓	✓	✓
Android	✓	✓	✗	✓
iOS	✓ (iOS 14.4)	✓ (iOS 14.4)	✓	✗

## Поддерживаемые способы публикации

По WebRTC ABR могут проигрываться потоки, опубликованные на WCS всеми доступными способами:

- WebRTC
- RTMP
- RTSP
- MPEG-TS по UDP или SRT
- Stream mixer

## Поддерживаемые кодеки

Для проигрывания WebRTC ABR любой поток на сервере будет транскодирован в H264+Opus.

## Принципы реализации

Для того, чтобы реализовать WebRTC ABR, используются [функции SFU с поддержкой Simulcast](#). При этом видео дорожка потока, опубликованного на сервере, транскодируется в несколько H264 дорожек с различными параметрами, и эти дорожки отправляются клиенту как отдельные качества. Аудио дорожка транскодируется в Opus и также отправляется клиенту. Поскольку основная сущность SFU - комната, на сервере создается комната с именем, соответствующим имени потока.

## Настройка

### Настройка подсистемы SFU

Для проигрывания потока, опубликованного на WCS, при помощи SFU Simulcast, необходимо:

- ограничить кодеки, используемые на стороне SFU, до H264 + Opus
- задать возможные профили кодирования H264
- включить мост между ядром WCS и подсистемой SFU

```
codecs_exclude_sfu=alaw, ulaw, g729, speex16, g722, mpeg4-generic, telephone-  
event, flv, mpv, vp8, h265  
profiles=42e01f, 640028  
wcs_sfu_bridge_enabled=true
```

После этого WCS необходимо перезапустить.

### Настройка профилей качества

Профили качества настраиваются в файле

`/usr/local/FlashphonerWebCallServer/conf/wcs_sfu_bridge_profiles.yml`. По умолчанию, указаны следующие профили:

```
profiles:  
  s :  
    width : 320  
    height : 240  
    bitrate : 500  
    gop : 60  
    fps : 30  
  m :  
    width : 640  
    height : 480  
    bitrate : 800  
    gop : 60
```

```
fps : 30
h :
width : 960
height : 720
bitrate : 1300
gop : 60
fps : 30
```

Поддерживаются следующие параметры профилей:

- `height` - высота картинки (обязательный параметр)
- `width` - ширина картинки
- `bitrate` - битрейт кодирования, в кбит/с
- `gop` - размер группы кадров
- `fps` - количество кадров в секунду

Параметр `gop` фактически задает периодичность кодирования ключевых кадров. В примере выше, при 30 кадрах в секунду, ключевой кадр будет формироваться каждые две секунды.

Имена профилей могут быть произвольными и используются для выбора качества на стороне клиента. В примере выше профили можно назвать `240p`, `480p` и `720p` соответственно.

Настройки профилей качества применяются после перезапуска WCS.

### Использование одинаковых профилей качества видео для HLS ABR и WebRTC ABR

Начиная со сборки [5.2.1665](#), если описания профилей видео совпадают для HLS ABR и WebRTC ABR, используются одинаковые кодировщики. Например, для WebRTC ABR

```
profiles:
  240p:
    height: 240
    bitrate: 500
    codec: h264
    gop: 60
    fps: 30

  480p:
    height: 480
    bitrate: 1000
    codec: h264
    gop: 60
    fps: 30

  720p:
    height: 720
    bitrate: 1500
    codec: h264
```

```
gop: 60
fps: 30
```

## и HLS ABR

```
profiles:
-240p:
  audio:
    codec: mpeg4-generic
    rate: 48000
    channels: 2
    groupId: audio
  video:
    height: 240
    bitrate: 500
    codec: h264
    gop: 60
    fps: 30
    audioGroupId: audio

-480p:
  audio:
    codec: mpeg4-generic
    rate: 48000
    channels: 2
    groupId: audio
  video:
    width: 0
    height: 480
    bitrate: 1000
    codec: h264
    gop: 60
    fps: 30
    audioGroupId: audio

-720p:
  audio:
    codec: mpeg4-generic
    rate: 48000
    channels: 2
    groupId: audio
  video:
    width: 0
    height: 720
    bitrate: 1500
    codec: h264
    gop: 60
    fps: 30
    audioGroupId: audio
```

будет создано 3 кодировщика видео

```
-----Native Resources-----
native_resources=139921985831216, NENC:H264/OPENH264, 495;139921847247232, mpeg4-
generic, 1852672;139922451207808, RESAMPLER:48000/48000, 0;139922048558376, FFbdecoderNative:H264/FFMPEG, 1489587;139922451204656, opus, -13542;139921983186880, NENC:H264/OPENH264, 495;139921983168640, NENC:H264/OPENH264, 495
native_resources.audio_codecs=2
native_resources.audio_resamplers=1
native_resources.video_transcoders=0
native_resources.video_decoders=1
native_resources.video_encoders=3
native_resources.writers=0
```

Должны совпадать следующие параметры профиля видео, если они указаны:

- height
- width (если указана и не равна 0)
- codec
- bitrate
- fps
- gop
- profile
- level
- codeclmpl

### Сортировка профилей качества видео

Начиная со сборки [5.2.1663](#), профили качества видео сортируются в том порядке, в каком они указаны в файле

`/usr/local/FlashphonerWebCallServer/conf/wcs_sfu_bridge_profiles.yml`. Например, при настройке

```
profiles:
  240:
    height: 240
    bitrate: 500
    codec: h264
    gop: 60
    fps: 30

  480:
    height: 480
    bitrate: 1000
    codec: h264
    gop: 60
    fps: 30

  720:
    height: 720
    bitrate: 1500
    codec: h264
    gop: 60
    fps: 30

  1080:
    height: 1080
    bitrate: 3000
    codec: h264
    gop: 60
    fps: 30
```

клиент получит список профилей в порядке

```
240, 480, 720, 1080
```

Если в настройке встречаются два профиля с одинаковыми названиями, возникает неопределенное поведение. В этом случае сервер будет использовать только последний из профилей с одинаковыми именами.

## Транскодирование максимального качества только при наличии В-фреймов в исходном потоке

Для того, чтобы снизить нагрузку на сервер при кодировании видео, в сборке [5.2.1840](#) добавлена возможность транскодировать максимальное ABR качество (которое обычно соответствует оригинальному разрешению потока) только при наличии В-фреймов в потоке. Эта возможность включается настройкой

```
h264_b_frames_force_transcoding=true
```

При этом сервер проверяет наличие В-фреймов в исходном потоке, анализируя заданное количество фреймов (по умолчанию 10)

```
frame_cnt_to_determine_their_type=10
```

Если в потоке есть В-фреймы, максимальное ABR качество будет транскодироваться, и будет доступно плееру.

Если в потоке нет В-фреймов, максимальное ABR качество не будет транскодироваться. В этом случае оригинальное качество необходимо запросить отдельно с клиента.

## Краткое руководство по тестированию

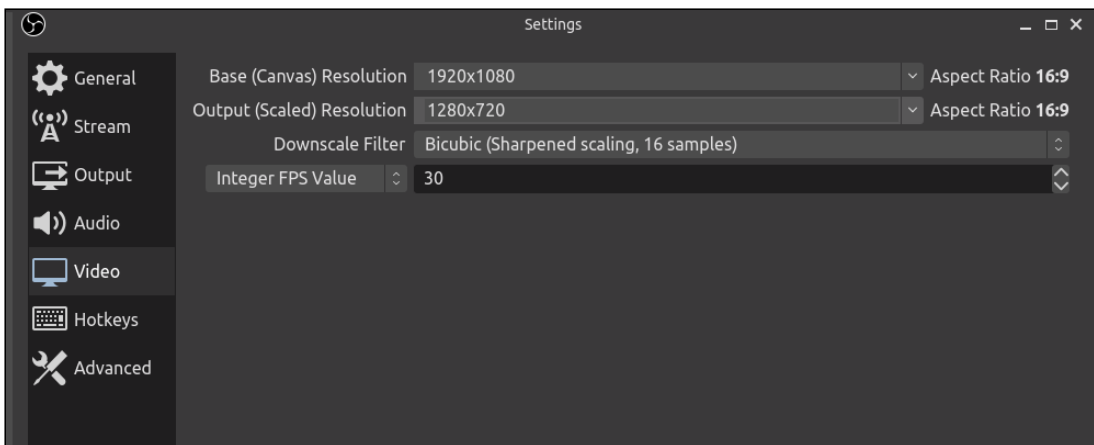
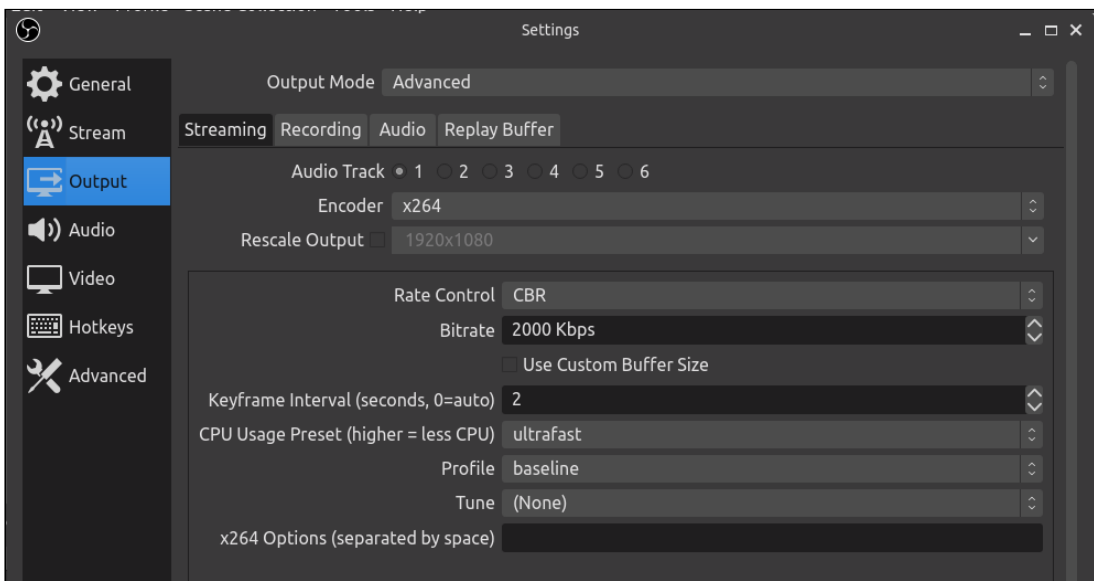
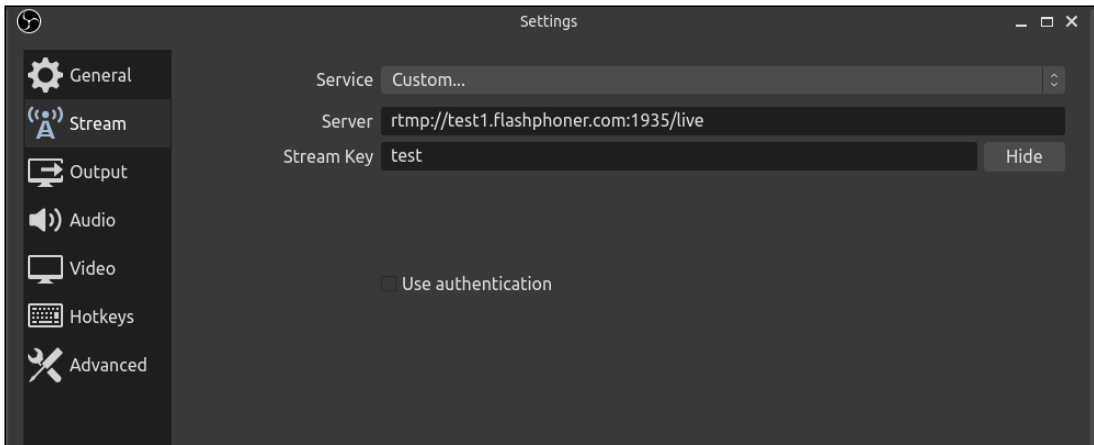
1. Для теста используем:

- WCS сервер с [настройками WebRTC ABR](#)
- OBS для публикации потока на сервере
- пример WebRTC ABR Player

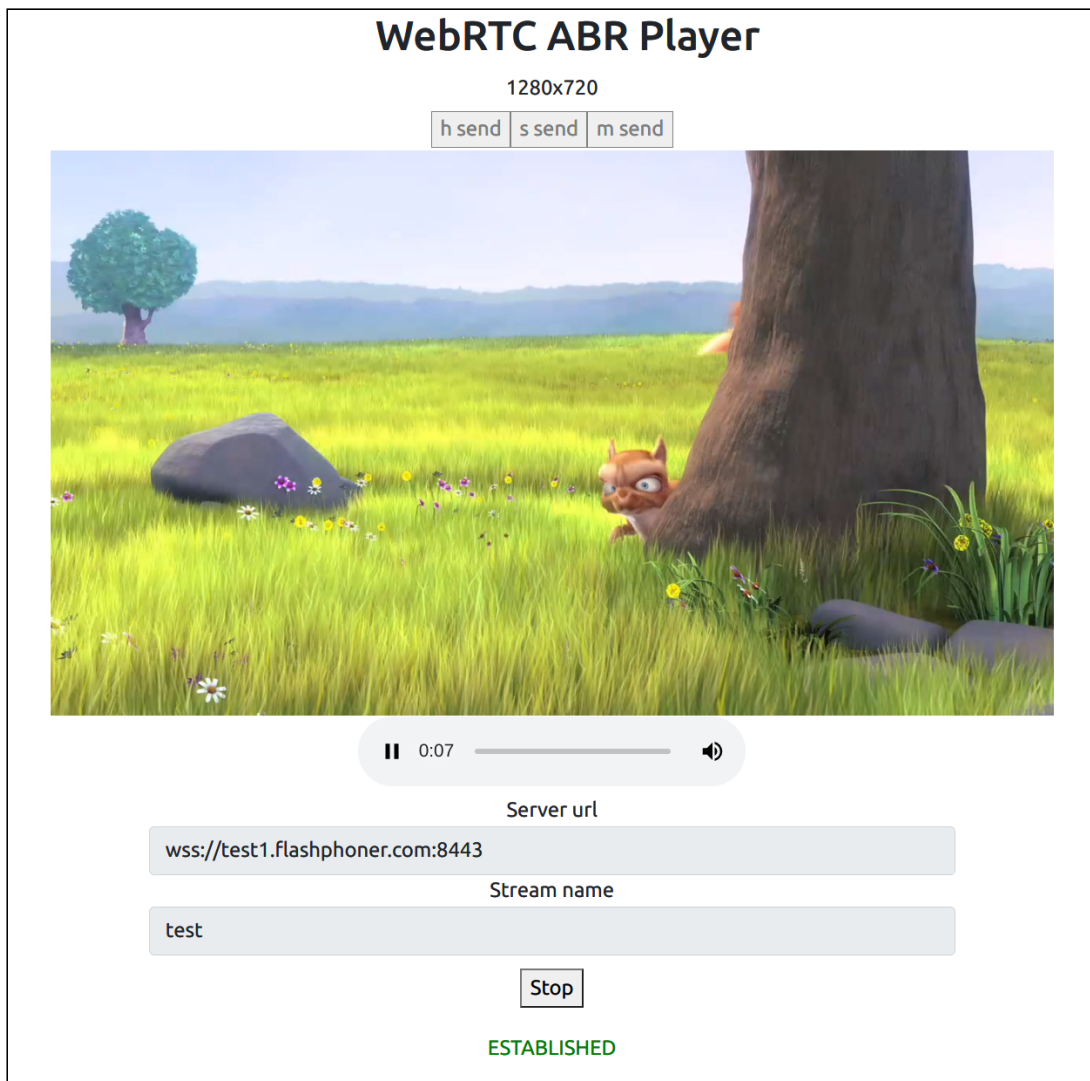
```
https://test1.flashphoner.com:8444/client2/sfu/webrtc-abr-  
player/player.html
```

 для проигрывания потока

2. Опубликуем RTMP поток `test` 720p 30 fps с битрейтом 2000 кбит/с



3. Откроем пример WebRTC ABR Player, указав в поле `Stream name` имя потока `test`, и нажмем `Play`. Начнется проигрывание потока



## Известные проблемы

1. Не все качества могут быть доступны зрителю, в зависимости от пропускной способности канала. При этом в примере WebRTC ABR Player кнопки тех качеств, которые недоступны, отображаются красным.
2. WebRTC ABR дает повышенную нагрузку на CPU сервера, т.к. на каждый опубликованный на сервере поток создается несколько кодировщиков, в зависимости от количества профилей.
3. В iOS Safari звук в примере WebRTC ABR Player по умолчанию выключен, на аудио элементе отображается перечеркнутый динамик, т.к. для включения звука iOS Safari требует действия пользователя.