

Управление битрейтом при захвате WebRTC потока в браузере

- [Описание](#)
 - [Поддерживаемые платформы и браузеры](#)
- [Настройки](#)
 - [REMB](#)
 - [Как это работает](#)
 - [TWCC](#)
- [Как форсировать повышение битрейта](#)
 - [Форсирование повышения битрейта в Chrome настройками сервера](#)
- [Использование](#)
- [Известные проблемы](#)

Описание

Для получения оптимального качества картинки с учетом пропускной способности канала при захвате WebRTC потока в браузере необходимо управлять битрейтом видео. Для этого в WCS предусмотрена возможность ограничения минимального и максимального битрейта видео в публикуемом потоке. Битрейт аудио не регулируется.

В сборках сервера до [5.2.1825](#) для управления битрейтом используется [REMB](#). Начиная со сборки [5.2.1825](#) по умолчанию включен [TWCC](#).

Поддерживаемые платформы и браузеры

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

Настройки

REMB

Начиная со сборки [5.2.1825](#) поддержка REMB включается настройкой

```
webrtc_cc2_twcc=false
```

Для ограничения битрейта предназначены следующие настройки WCS:

	На стороне браузера (JavaScript)	На стороне сервера (flashphoner.properties)
Ограничение минимального битрейта	constraints.video.minBitrate	webrtc_cc_min_bitrate
Ограничение максимального битрейта	constraints.video.maxBitrate	webrtc_cc_max_bitrate

При этом, на стороне браузера битрейт задается в килобитах в секунду, например

```
constraints.video.maxBitrate=600
```

а на стороне сервера в битах в секунду

```
webrtc_cc_max_bitrate=600000
```

Если заданы настройки с обеих сторон, то настройки браузера имеют приоритет над настройками сервера.

Если настройки браузера не заданы, применяются настройки сервера.

Если не заданы ни те, ни другие настройки, применяются значения по умолчанию

```
webrtc_cc_min_bitrate=30000
webrtc_cc_max_bitrate=10000000
```

Эти настройки работают в основных современных браузерах и задают границы управления битрейтом при помощи [REMB](#).

Как это работает

Если задан `maxBitrate`, WCS сервер, при достижении указанной планки, будет отправлять браузеру REMB-команду снизить битрейт. Если задан `minBitrate`, WCS сервер, при достижении указанной планки, перестанет отправлять браузеру REMB-команды снижения битрейта. Таким образом, настройки обозначают три диапазона, в каждом из которых WCS сервер управляет битрейтом:

№	Диапазон	Управление
1	[0, minBitrate]	WCS сервер прекращает управление битрейтом и не отправляет REMB сообщений
2	[minBitrate, maxBitrate]	WCS сервер осуществляет активное управление битрейтом: в зависимости от джиттера и равномерности входящего трафика, WCS принимает решение об отправке REMB команд для снижения битрейта. Если с каналом все хорошо, WCS ничего не делает и битрейт не снижается.
3	[maxBitrate, ...]	В этом диапазоне WCS сервер постоянно посылает команды на снижение битрейта до maxBitrate

TWCC

Начиная со сборки [5.2.1825](#), TWCC включен по умолчанию

```
webrtc_cc2_twcc=true
```

Ограничение битрейта для TWCC можно настроить только на стороне браузера, в кбит/с

```
constraints.video.minBitrate=500
constraints.video.maxBitrate=1000
```

Настройки битрейта для TWCC работают в Android SDK [1.1.0.62](#), iOS SDK [2.6.122](#) и WebSDK [2.0.239](#).

Как форсировать повышение битрейта

В настоящее время, форсировать повышение битрейта можно [только в браузере на основе Chromium](#), указав в SDP при помощи [замены параметров](#) `x-google-max-bitrate` и `x-google-min-bitrate`.

Через настройки на стороне клиента и сервера форсировать увеличение битрейта невозможно, можно только управлять его понижением. При этом, настройки, специфичные для Chrome, если они выставлены, будут иметь приоритет, т.е. настройки `constraints` и настройки сервера будут игнорироваться. Отметим, что настройка Chrome по умолчанию, определенная опытным путем, на сегодняшний день составляет

```
x-google-max-bitrate=2500
```

В последних сборках Chrome при форсировании битрейта на стороне браузера необходимо указывать опцию `videoContentHint: "motion"`, поскольку при других положениях данной опции Chrome сбрасывает битрейт ради того, чтобы удержать разрешение публикации

```
session.createStream({
  name: streamName,
  display: localVideo,
  ...
  videoContentHint: "motion"
}).publish();
```

Форсирование повышения битрейта в Chrome настройками сервера

Настройки для замены параметров SDP в браузере Chrome могут быть выставлены на стороне сервера при помощи параметров

```
webrtc_sdp_min_bitrate_bps=3000000  
webrtc_sdp_max_bitrate_bps=7000000
```

Эти параметры задаются в битах в секунду. В данном случае настройки аналогичны следующим

```
x-google-max-bitrate=7000;x-google-min-bitrate=3000
```

Эти настройки предназначены для Chromium-based браузеров. Также применяются при использовании Safari 12 и выше.

В Firefox форсирование битрейта не работает.

Использование

Удержание битрейта в определенных границах может быть полезным, например, при публикации видео для клиентов, использующих браузер Safari. Данный браузер чувствителен к резким изменениям битрейта, при этом ухудшается качество картинки вплоть до фризов и зависаний браузера. Поэтому при вещании для Safari рекомендуется максимально стабилизировать битрейт, задавая узкие пределы изменения, например

```
constraints.video.minBitrate=600  
constraints.video.maxBitrate=600
```

В этом случае картинка в Safari будет сохранять приемлемое качество, в зависимости от полосы пропускания и состояния канала.

Форсирование повышения битрейта необходимо при публикации HD и 4K потоков. В этом случае рекомендуется использовать для публикации браузер Chrome.

Известные проблемы

1. В некоторых версиях (например, Chrome 75) браузер при публикации WebRTC H264 потока прижимает битрейт к нижней границе x-google-min-bitrate

Симптомы: при настройке

```
x-google-min-bitrate=3000;x-google-max-bitrate=7000
```

битрейт публикации удерживается около 3000 кбит/с, при этом пропускная способность канала находится на уровне 20 Мбит/с

Решение: увеличить границы битрейта вдвое, например

```
webrtc_sdp_min_bitrate_bps=6000000  
webrtc_sdp_max_bitrate_bps=14000000
```