

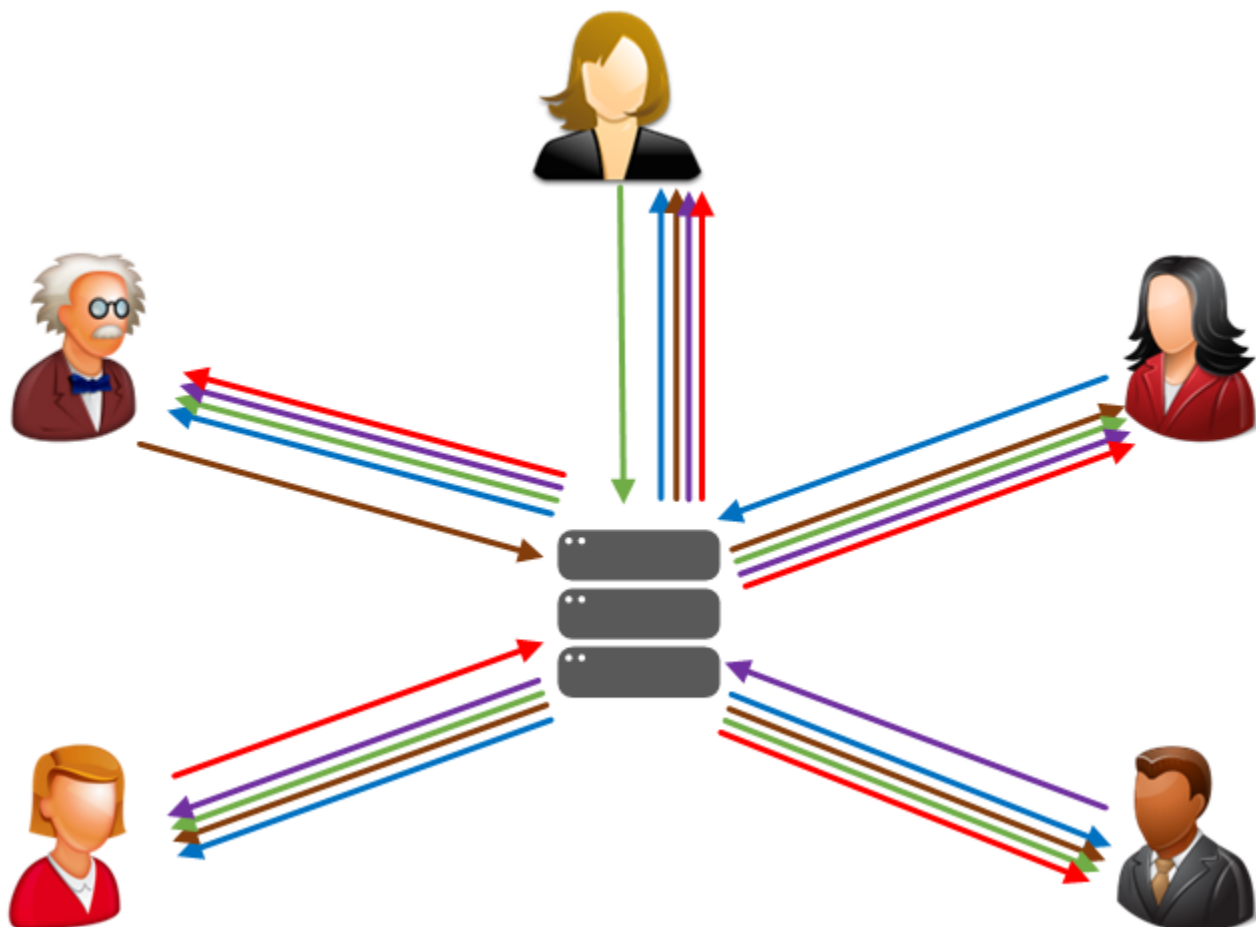
Функции SFU с поддержкой Simulcast

- Описание
 - Поддерживаемые платформы и браузеры
 - Поддерживаемые кодеки
- Принципы реализации
 - Конфигурация комнаты
 - Публикация потоков в комнате
- Настройка сервера
 - Публикация H264
- Краткое руководство по тестированию
- Мониторинг потоков в комнате
 - REST вызовы
 - REST-методы и статусы ответа
 - Параметры
- Действия над SFU потоками на сервере
 - Ограничения
- Поддержка TURN
- Известные проблемы

Описание

В сборке 5.2.1056 добавлена поддержка WebRTC Selective Forwarding Unit (SFU) с возможностью публикации и проигрывания любого количества потоков в одном WebRTC соединении (Simulcast). Основные области применения:

- публикация потока в нескольких качествах (например, 720p, 480p, 360p) с возможностью автоматического и ручного переключения качества
- организация видео и аудио чат-комнат



Поддерживаемые платформы и браузеры

	Chrome	Firefox	Safari 11	Chromium Edge
Windows	+	-		+
Mac OS	+	-	+	
Android	+	-		+
iOS	+ (iOS 14.4)	-	+	

Поддерживаемые кодеки

WebRTC video:

- H264
- VP8

WebRTC audio:

- Opus

Принципы реализации

Поскольку одним из случаев применения SFU являются видео и аудио конференции, на стороне сервера введена новая сущность - комната. Установив соединение с сервером, пользователь подключается к комнате и может публиковать свои потоки, а также проигрывать все потоки в данной комнате. За пределами комнаты потоки, опубликованные в ней, недоступны.

Конфигурация комнаты

Пример объекта, задающего конфигурацию комнаты:

```
"room": {
  "url": "wss://wcs:8443",
  "name": "ROOM1",
  "pin": "1234",
  "nickName": "User1"
}
```

Здесь

- url - WebSocket URL WCS-сервера
- name - уникальное имя комнаты
- pin - пин-код
- nickName - имя пользователя в комнате

Публикация потоков в комнате

Пользователь может добавлять и удалять видео и аудио потоки. При добавлении может быть указан набор качеств, и поток будет опубликован в этих качествах. Каждое качество имеет следующие характеристики:

- максимальный битрейт
- коэффициент масштабирования по отношению к разрешению оригинального потока (в сторону уменьшения)

При проигрывании потока пользователю доступны все качества, либо те из них, которые умещаются в канал подписчика. Например, при публикации потока 720p с качествами 720p 900 kbps, 360p 500 kbps и 180p 200 kbps, подписчик может получать только 360p и 180p, если его канала до сервера недостаточно для получения 720p.

Пример настройки публикации медиапотоков в комнате

```

"media": {
  "audio": {
    "tracks": [{
      "source": "mic",
      "channels": 1
    }]
  },
  "video": {
    "tracks": [{
      "source": "camera",
      "width": 1280,
      "height": 720,
      "codec": "H264",
      "encodings": [
        { "rid": "h", "active": true, "maxBitrate": 900000 },
        { "rid": "m", "active": true, "maxBitrate": 300000, "scaleResolutionDownBy": 2 }
      ]
    }]
  }
}

```

Здесь

- audio - конфигурация аудиодорожек
- video - конфигурация видеодорожек
- source - источник публикации: camera, screen, mic
- channels - количество каналов аудио
- width, height - исходная ширина и высота картинки видео
- codec - кодек видео: H264 или VP8
- encodings - набор качеств, с которыми поток будет опубликован

Параметры качества указываются в соответствии с описанием [RTCRtpEncodingParameters](#).

Настройка сервера

Публикация H264

По умолчанию, даже если в параметрах публикации задан кодек H264, в комнате будут публиковаться потоки VP8. Чтобы публиковать H264, необходимо:

- исключить все кодеки, кроме H264, чтобы убрать их из SDP
- ограничить минимальный битрейт публикации
- ограничить используемые профили кодирования H264

```

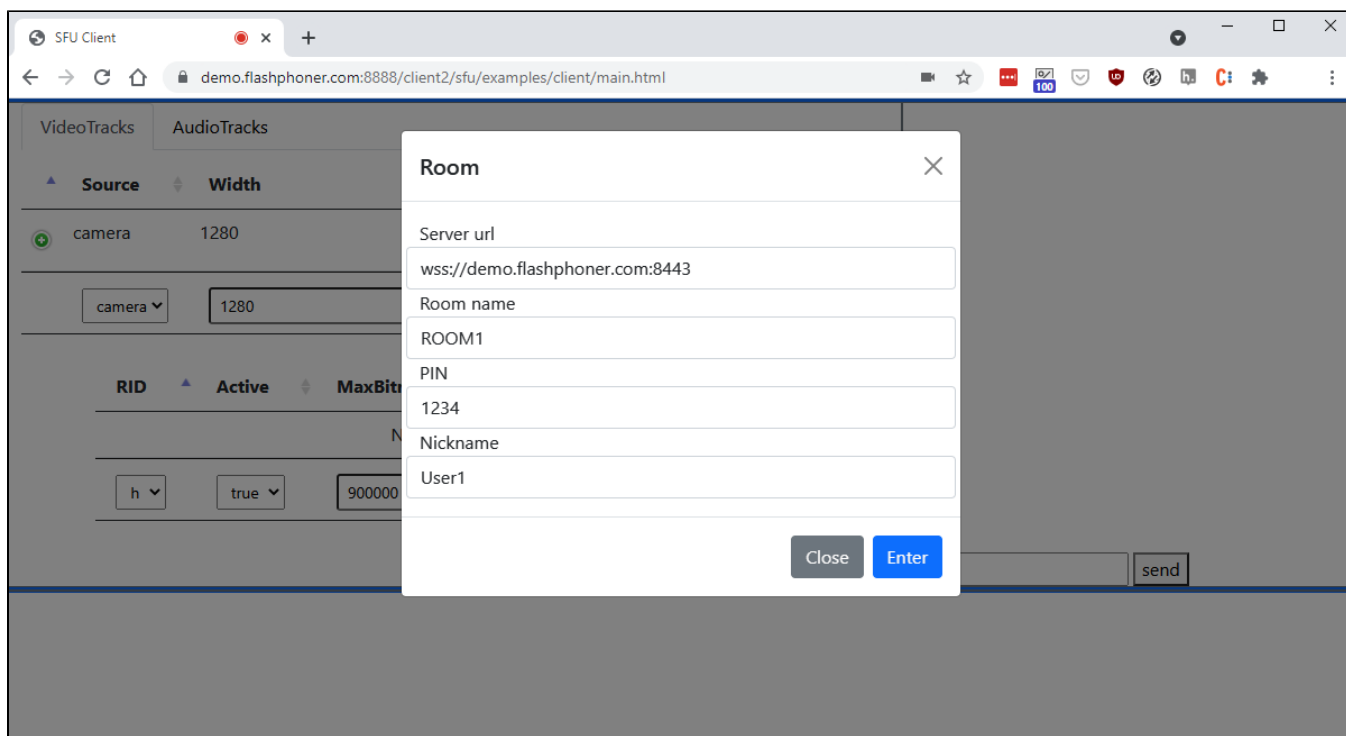
codecs_exclude_sfu=alaw,ulaw,g729,speex16,g722,mpeg4-generic,telephone-event,flv,mpv,vp8,h265
webrtc_cc_min_bitrate=1000000
profiles=42e01f,640028

```

Отметим, что публикация и проигрывание большого числа VP8 потоков в нескольких качествах потребляет ресурсы клиентского ПК. Если ресурсов не хватает, следует предпочесть H264, поскольку большинство браузеров поддерживают аппаратное ускорение для данного кодека.

Краткое руководство по тестированию

1. Откройте пример SFU client в браузере, например <https://demo.flashphoner.com:8888/client2/sfu/client/main.html>, введите URL сервера, имя комнаты, пин-код и имя пользователя, нажмите Enter



2. Поток пользователя User1 публикуется в комнате ROOM1

VideoTracks

AudioTracks

Source	Width	Height	Codec	Action
camera	1280	720	H264	Delete

camera

1280

720

H264

Add

RID	Active	MaxBitrate	ResolutionScale	Action
No data available in table				

h

true

900000

1

Add

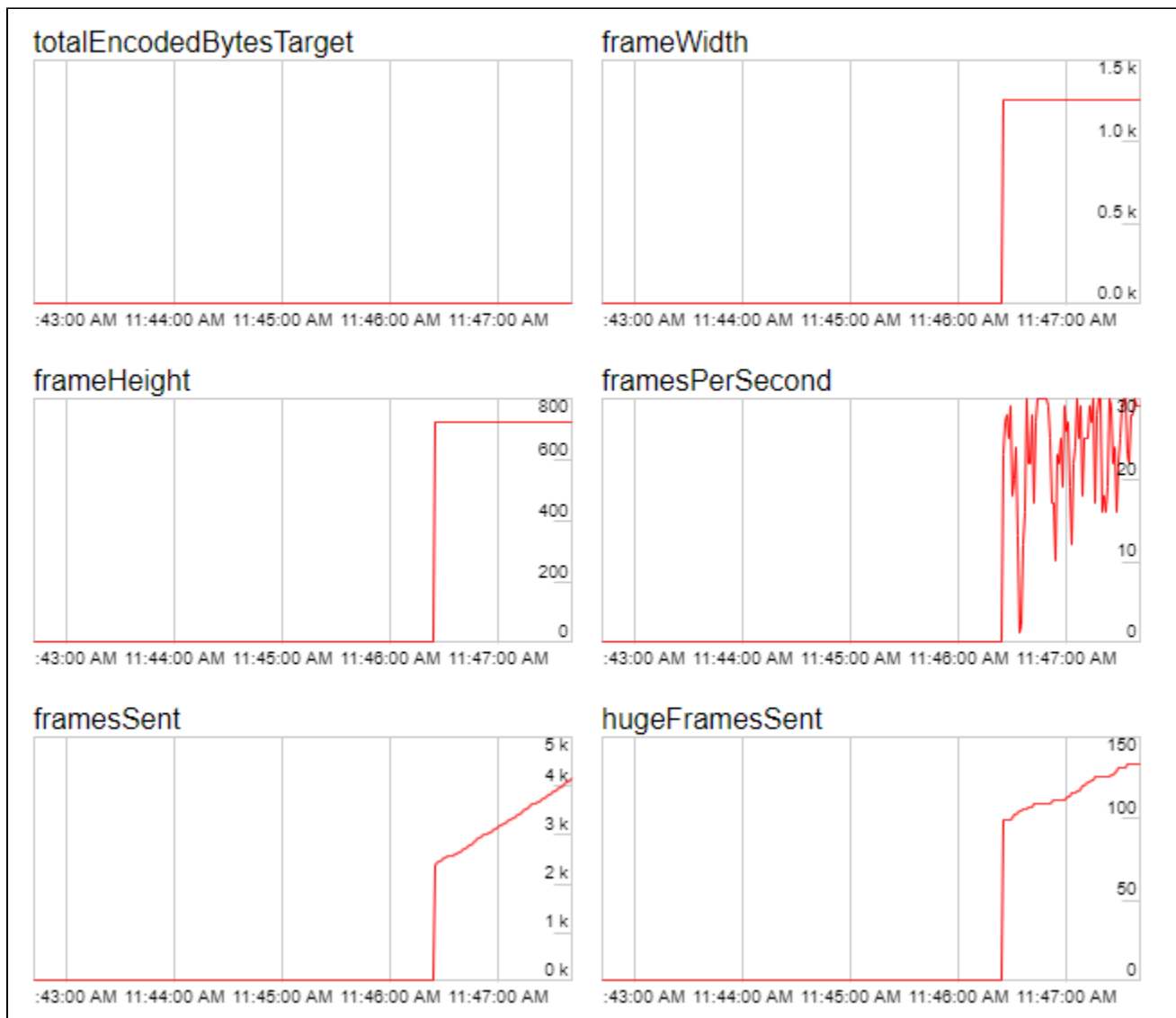
11:44:14 User1:
JOINED

send

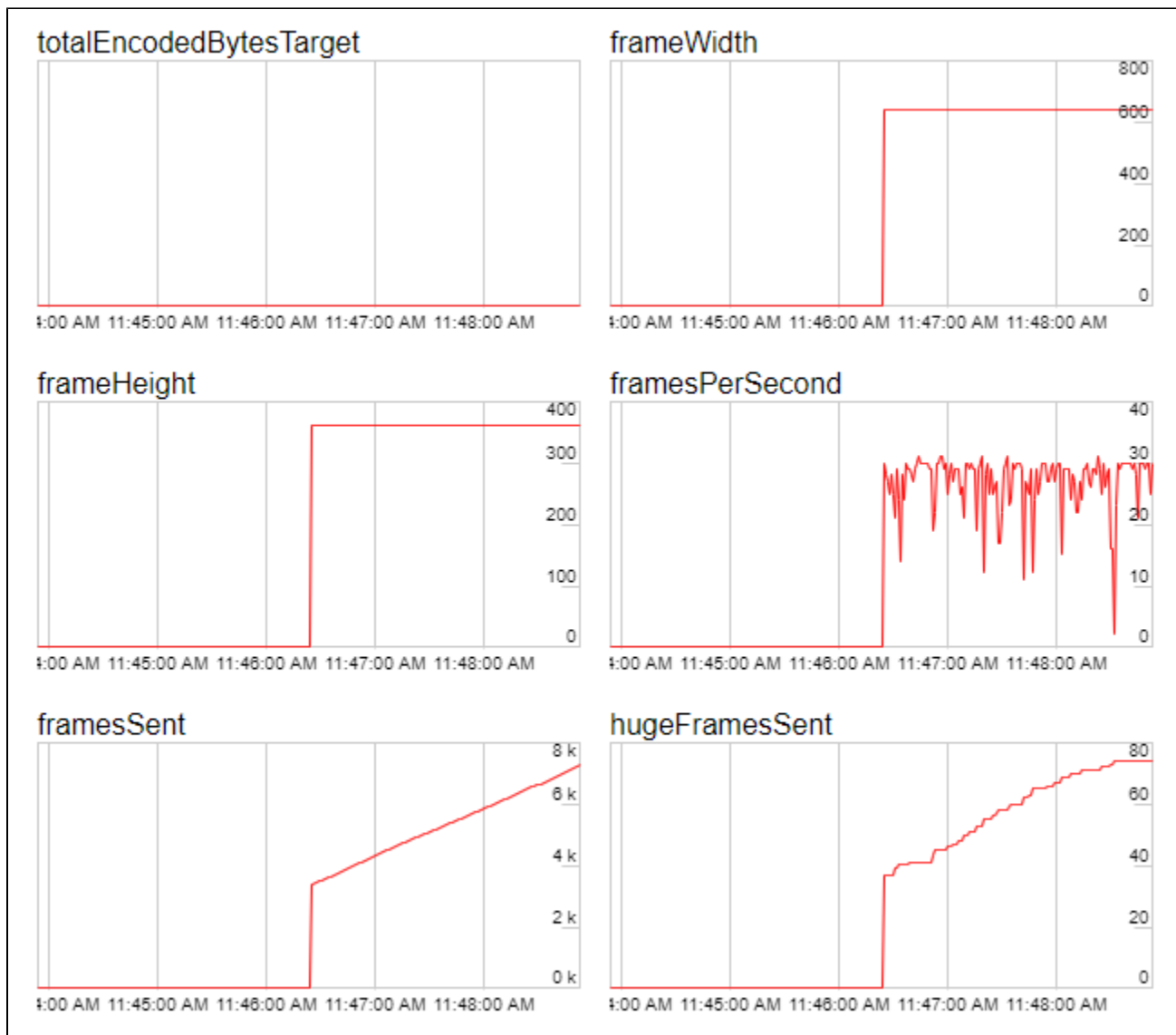
Name: local 1280x720

Audio state: false

Статистика отправки качества 720p



Статистика отправки качества 360p



3. Откройте страницу примера в другом браузере или в другом окне браузера, введите URL сервера и параметры комнаты такие же, как на шаге 3, имя пользователя User2

Room
×

Server url

wss://test1.flashphoner.com:8443

Room name

ROOM1

PIN

1234

Nickname

User2

Close

Enter

4. Поток пользователя User2 проигрывается у окне User1

VideoTracks

AudioTracks

Source	Width	Height	Codec	Action
camera	1280	720	H264	Delete

camera

1280

720

H264

Add

RID	Active	MaxBitrate	ResolutionScale	Action
No data available in table				

h

true

900000

1

Add

11:44:14 User1:
JOINED
11:52:52 User2:
JOINED

send

Name: local 1280x720

Audio state: false

Name: User2 640x360

h send l m send

TID0 TID1 TID2

Мониторинг потоков в комнате

Для мониторинга потоков в комнате используется REST API

REST вызовы

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

- HTTP:http://wcs:8081/rest-api/sfu/stats
- HTTPS:https://wcs:8444/rest-api/sfu/stats

Здесь:

- wcs - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /sfu/stats - используемый REST-метод

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/sfu/stats	<pre>{ "roomName": "ROOM1" }</pre>	<pre>{ "participants": [{ "nickName": "User1", "outgoingTracks": [{ "id": "9de9107c-ce5f-4d6b-b7d6-ea233d691d09", "codec": "opus", "bitrate": 0, "sampleRate": 48000, "channels": 2, "alive": true, "type": "AUDIO" }, { "id": "237dcef9-c66d-4c72-bd43-0c91aaea3b7e", "composite": true, "tracks": { "h send": { "id": "237dcef9-c66d-4c72-bd43-0c91aaea3b7e", "codec": "H264", "width": 1280, "height": 720, "fps": 30, "bitrate": 157976, "alive": true, "type": "VIDEO" }, "m send": { "id": "237dcef9-c66d-4c72-bd43-0c91aaea3b7e", "codec": "H264", "width": 640, "height": 360, "fps": 30, "bitrate": 263952, "alive": true, "type": "VIDEO" } } }], "incomingTracks": { "3c2dcd1c-7acd-4b90-8871-331be80cade0": "h send" } }, { "nickName": "User2",</pre>	200 - OK 404 - Not found 500 - Internal error	Показать текущую статистику комнаты

		<pre>"outgoingTracks": [{ "id": "3c2dcd1c-7acd-4b90-8871-331be80cade0", "composite": true, "tracks": { "h send": { "id": "3c2dcd1c-7acd-4b90-8871-331be80cade0", "codec": "H264", "width": 1280, "height": 720, "fps": 30, "bitrate": 238688, "alive": true, "type": "VIDEO" }, "m send": { "id": "3c2dcd1c-7acd-4b90-8871-331be80cade0", "codec": "H264", "width": 640, "height": 360, "fps": 30, "bitrate": 265368, "alive": true, "type": "VIDEO" } } }], "incomingTracks": { "9de9107c-ce5f-4d6b-b7d6-ea233d691d09": null, "237dcef9-c66d-4c72-bd43-0c91aaea3b7e": "h send" } }]</pre>		
--	--	---	--	--

Параметры

Имя параметра	Описание	Пример
romName	Имя комнаты	ROOM1
participants	Список участников комнаты	[]
nickName	Имя пользователя	User1
outgoingTracks	Список потоков, публикуемых пользователем	[]
incomingTracks	Список потоков, проигрываемых пользователем	{}
id	Идентификатор медиасессии	9de9107c-ce5f-4d6b-b7d6-ea233d691d09
codec	Видео или аудио кодек	H264
width	Ширина картинки видео	1280
heigth	Высота картинки видео	720
fps	FPS видео	30
bitrate	Битрейт видео или аудио, бит/с	265368
sampleRate	Частота дискретизации аудио, Гц	48000
channels	Количество каналов аудио	2

alive	Поток активен	true
type	Тип потока	VIDEO
composite	Поток включает несколько дорожек	true
tracks	Список дорожек в композитном потоке	{}

Действия над SFU потоками на сервере

В сборке [5.2.1068](#) добавлена возможность отображения SFU потоков как обычных WebRTC потоков на сервере. Эта возможность включается настройкой

```
sfu_bridge_enabled=true
```

При этом для каждого участника будет создан поток с именем {room}-{participant}-VIDEO для видео потока и {room}-{participant}-AUDIO для аудио. Эти потоки видны в статистике

```
-----Stream Stats-----  
...  
streams_viewers=ROOM1-User1-AUDIO/0;ROOM1-User1-VIDEO/0  
streams_synchronization=ROOM1-User1-AUDIO/0;ROOM1-User1-VIDEO/0
```

могут быть проиграны с сервера

Player



WCS URL

wss://test1.flashphoner.com:8443

Stream

ROOM1-User1-VIDEO

Volume



Full Screen



PLAYING

Stop

записаны [по REST API](#) или [добавлены в микшер](#).

При публикации экрана создается поток с именем {room}-{participant}-VIDEO-screen, например

```
-----Stream Stats-----  
...  
streams_viewers=ROOM1-User1-AUDIO/0;ROOM1-User1-VIDEO-screen/0;ROOM1-User1-VIDEO/0  
streams_synchronization=ROOM1-User1-AUDIO/0;ROOM1-User1-VIDEO-screen/0;ROOM1-User1-VIDEO/0
```

Если поток опубликован в нескольких качествах, на WCS будет доступен поток в максимальном качестве, которое публикуется в настоящее время, например 720p. Если это качество перестает публиковаться (например, ухудшается канал участника), будет произведено автоматическое переключение на следующее доступное качество, например 360p.

Ограничения

Если участник публикует более одного потока с камеры, на WCS будет доступен только один поток.

Поддержка TURN

Поскольку для публикации и проигрывания аудио и видео дорожек в браузере используется стандартный объект `RTCPeerConnection`, работа через TURN сервер настраивается при создании этого объекта. Например, по умолчанию потоки в примере SFU Two Way Streaming публикуются напрямую:

[code](#)

```
pc = new RTCPeerConnection();
...
```

Если необходимо использовать TURN сервер, например, встроенный TURN сервер WCS, код должен быть изменен следующим образом:

```
let connectionConfig = {
  iceServers: [
    {
      urls: 'turn:wcs:3478?transport=tcp',
      credential: 'coM77EMrV7Cwhyay',
      username: 'flashphoner'
    }
  ],
  iceTransportPolicy: "relay"
};
pc = new RTCPeerConnection(connectionConfig);
...
```

Здесь:

- `wcs` - адрес WCS сервера;
- `flashphoner` - имя пользователя встроенного TURN сервера по умолчанию;
- `coM77EMrV7Cwhyay` - пароль встроенного TURN сервера по умолчанию

В этом случае медиа трафик пойдет через TURN сервер, встроенный в WCS. Эту возможность можно использовать также для обертывания WebRTC трафика в TCP при плохом качестве каналов у клиента, т.к. WCS не поддерживает TCP транспорт для SFU потоков.

Известные проблемы

1. Публикация потока, захваченного с какого-либо из окон на экране в нескольких качествах, и сворачивание этого окна приводит к крашу вкладки браузера Chrome

Симптомы: при сворачивании окна, которое захватывается в данный момент, вкладка Chrome, с которой производится публикация, крашится

Решение: открыт [баг в Chromium](#), до исправления этого бага (исправлен в версии Chrome 98.0.4736.0) публиковать поток, захваченный с какого-либо окна на экране, в одном качестве (без Simulcast)