На другой RTMP сервер

Описание

Web Call Server по запросу конвертирует WebRTC аудио и видео поток в RTMP и отправляет на указанный RTMP-сервер. Таким образом может быть создана трансляция с веб-страницы наFacebook,YouTube Live,Wowza,Azure Media Servicesи другие сервисы, транслирующие живое видео.

Ретрансляция RTMP-потока может быть организована как при помощиREST-вызовов, так и средствами JavaScript API.

Поддерживаемые платформы и браузеры

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

Поддерживаемые кодеки

- Видео: H.264
- Аудио: AAC, G.711, Speex 16

Аутентификация на RTMP-сервере

Поддерживается, имя и пароль необходимо указывать в URL сервера, напримерrtmp://name:password@server:1935/live

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.

3. REST-клиент из браузера отправляет запрос /push/startup.
4. WCS-сервер публикует RTMP поток на указанный в запросе URL RTMP-сервера.
5. WCS-сервер передает RTMP поток.

REST-вызовы

Ретрансляция видеопотока на другой RTMP-сервер производится при помощи REST-вызовов

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

- HTTP:http://streaming.flashphoner.com:8081/rest-api/push/startup
- HTTPS:https://streaming.flashphoner.com:8444/rest-api/push/startup

Здесь:

- streaming.flashphoner.com- адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444- стандартный HTTPS порт
- rest-api- обязательный префикс
- /push/startup- используемый REST-вызов

REST-вызовы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример ответа	Статусы ответа	Описание

/push /startup	<pre>{
	"streamName": "name",
	"rtmpUrl": "rtmp://localhost:1935/live",
	"rtmpTransponderFullUrl": false
	"options": {}
	}</pre>
		<pre>{
		"mediaSessionId": "eume87rjk3df1i9u14elffga6t",
		"streamName": "rtmp_name",
		"rtmpUrl": "rtmp://localhost:1935/live",
		"width": 320,
		"height": 240,
		"muted": false,
		"soundEnabled": false,
		"options": {}
		}</pre>
			409 - Conflict
			
			500 - Internal error
				Создает транспондер, который подписывается на указанный поток и отправляет медиа трафик на указанный rtmpUrl.
				
				Имя потока, указанное в запросе, может быть именем уже публикуемого потока или именем зарезервированным при создании SIP-звонка (для отправки медиа трафика, полученного со стороны SIP).
				
				Если транспондер для такого потока и rtmpUrl уже существует, возвращает 409 Conflict.

/push /find	<pre>{
	"streamName": "name",
	"rtmpUrl": "rtmp://localhost:1935/live",
	}</pre>
		<pre>[{
		"mediaSessionId": "eume87rjk3df1i9u14elffga6t",
		"streamName": "rtmp_name",
		"rtmpUrl": "rtmp://localhost:1935/live",
		"width": 320,
		"height": 240,
		"muted": false,
		"soundEnabled": false,
		"options": {}
		}]</pre>
			404 - Transponder not found
			
			500 - Internal error
				Найти транспондеры по фильтру

/push /find_all		<pre>[{ "mediaSessionId" : "eume87rjk3df1i9u14elffga6t", "streamName": "rtmp_name", "rtmpUrl": "rtmp://localhost:1935/live", "width": 320, "height": 240, "muted": false, "soundEnabled": false, "options": {} }]</pre>	404 - Not found any transponder 500 - Internal error	Найти все транспондеры
/push /terminate	<pre>{ "mediaSessionId" : "eume87rjk3df1i9u14elffga6t" }</pre>		404 - Not found transponder 500 - Internal error	Завершить работу транспондера
/push /mute	<pre>{ "mediaSessionId" : "eume87rjk3df1i9u14elffga6t" }</pre>	void	404 - Not found transponder 500 - Internal error	Выключить аудио
/push /unmute	<pre>{ "mediaSessionId" : "eume87rjk3df1i9u14elffga6t" }</pre>	void	404 - Not found transponder 500 - Internal error	Включить аудио
/push /sound_on	<pre>{ "mediaSessionId" : "eume87rjk3df1i9u14elffga6t" "soundFile": "test.wav" "loop": true }</pre>	void	404 - Not found transponder 404 - No such file 500 - Internal error	Вставить аудио из RIFF WAV файла из директории /usr/local /FlashphonerWebCallServer/media/ на WCS-сервере

/push/sound_off	<pre>{
	"mediaSessionId":
	"eume87rjk3df1i9u14elffga6t"
	}</pre>
		void	404 - Not found transponder
			
			500 - Internal error
				Завершить вставку аудио из файла

Параметры

Имя параметра	Описание	Пример
streamName	Имя ретранслируемого потока	streamName
rtmpUrl	URL сервера, на который производится ретрансляция	rtmp://localhost:1935/live
rtmpFlashVersion	Версия Flash RTMP-клиента	LNX 76.219.189.0
options	Опции транспондера	{"action": "mute"}
mediaSessionId	Уникальный идентификатор транспондера	eume87rjk3df1i9u14elffga6t
width	Ширина изображения	320
height	Высота изображения	240
keyFrameInterval	Интервал ключевых кадров видео	60
fps	Частота кадров видео	30
muted	Звук приглушен	true
soundEnabled	Звук отключен	true
soundFile	Файл звуковой дорожки	test.wav
loop	Повтор	false
rtmpTransponderFullUrl	Брать имя для публикации на RTMP сервере из RTMP URL	false

Начиная со сборки5.2.785, добавлены следующие параметры: rtmpFlashVersion, keyFrameInterval и fps.

Параметроptionsможет быть использован, чтобы выключить аудио или вставить аудио из файла при создании транспондера.

Например,

```
"options": {"action": "mute"}
"options": {"action": "sound_on", "soundFile": "sound.wav", "loop": true}
```

Транскодинг при ретрансляции потока

Начиная со сборки5.2.560, если в параметрах запроса/push/startup не указывать ширину и высоту картинки

```
{
 "streamName": "name",
 "rtmpUrl": "rtmp://localhost:1935/live"
}
```

либо указать их равными 0

```
{
 "streamName": "name",
 "rtmpUrl": "rtmp://localhost:1935/live",
 "width": 0,
 "height": 0
}
```

транскодинг для ретранслируемого потока не включается.

Если задать высоту картинки явным образом (например, если сервер не принимает потоки меньше чем 720p)

```
{
 "streamName": "name",
 "rtmpUrl": "rtmp://localhost:1935/live",
 "width": 1280,
 "height": 720
}
```

поток при ретрансляции будет транскодирован и отправлен на целевой сервер в указанном разрешении.

Заданная ширина картинки применяется, только если отключено сохранение соотношения сторон, и задана и высота картинки. Если передать только параметр width без height, то он не применяется, и транскодинг для ретранслируемого потока не включается.

Начиная со сборки5.2.785, добавлены другие два параметра, с которыми включается транскодинг: keyFrameInterval и fps. Транскодинг для ретранслируемого потока включается, если задан любой из них или высота картинки.

Указание имени потока для публикации на RTMP сервере

По умолчанию, поток будет опубликован на RTMP сервере с тем же именем, под которым он опубликован на WCS, и префиксом `rtmp_`, например `rtmp_test`.Это поведение меняется настройками

```
rtmp_transponder_full_url=true
rtmp_transponder_stream_name_prefix=
```

Однако, эти настройки применяются ко всем ретрансляциям, и требуют перезапуска сервера. Поэтому в сборке5.2.860добавлен параметр запроса /push/startup, позволяющий указать полный RTMP URL, включая имя потока на RTMP сервере

```
POST /rest-api/push/startup HTTP/1.1
Host: localhost:8081
Content-Type: application/json

{
 "streamName":"stream1",
 "rtmpUrl":"rtmp://rtmp.flashphoner.com:1935/live/test",
 "rtmpTransponderFullUrl":true
}
```

В этом случае поток будет опубликован на RTMP сервере под именем, указаннымв RTMP URL, даже при настройках WCS по умолчанию

Отправка REST-запроса к WCS-серверу

Для отправки REST-запроса к WCS-серверу необходимо использоватьREST-клиент.

JavaScript API

При помощи WebSDK поток может быть ретранслирован на RTMP-сервер при создании, по аналогии с функциейSIP as stream. Пример использования данного метода приведен в веб-приложении WebRTC as RTMP.

webrtc-as-rtmp-republishing.html

webrtc-as-rtmp-republishing.js

1. При создании потока методу session.createStream() передается параметр rtmpUrl с указанием URL RTMP-сервера, принимающего трансляцию. Имя потока указывается в соответствии с правилами RTMP-сервера.

код:

```
function startStreaming(session) {
    var streamName = field("streamName");
    var rtmpUrl = field("rtmpUrl");
    session.createStream({
        name: streamName,
        display: localVideo,
        cacheLocalResources: true,
        receiveVideo: false,
        receiveAudio: false,
        rtmpUrl: rtmpUrl
        ...
    }).publish();
}
```

Ретрансляция потока начинается сразу после его успешной публикации на WCS-сервере

Настройка сервера

При создании RTMP-транспондера, WCS автоматически добавляет к имени ретранслируемого потока префикс в соответствии с настройкой в файлеflashphoner.properties:

```
rtmp_transponder_stream_name_prefix=rtmp_
```

Если сервер, на который ретранслируется поток, предъявляет определенные требования к имени (Facebook,YouTube), данная строка должна быть закомментирована.

Настройка

```
rtmp_transponder_full_url=true
```

включает возможность передачи параметров запроса серверу, на который ретранслируется поток.

Указать сетевой интерфейс для трансляции RTMP можно при помощи параметра

```
rtmp_publisher_ip=127.0.0.1
```

В данном случае RTMP будет публиковаться только на localhost

Передача параметров в URL сервера

Существует возможность передать параметры запроса серверу, на который ретранслируется поток, указав их в URL сервера, например

```
rtmp://myrtmpserver.com:1935/app_name/?user=user1&pass=pass1
```

или, если публикация предполагается в отдельный экземпляр приложения на RTMP-сервере

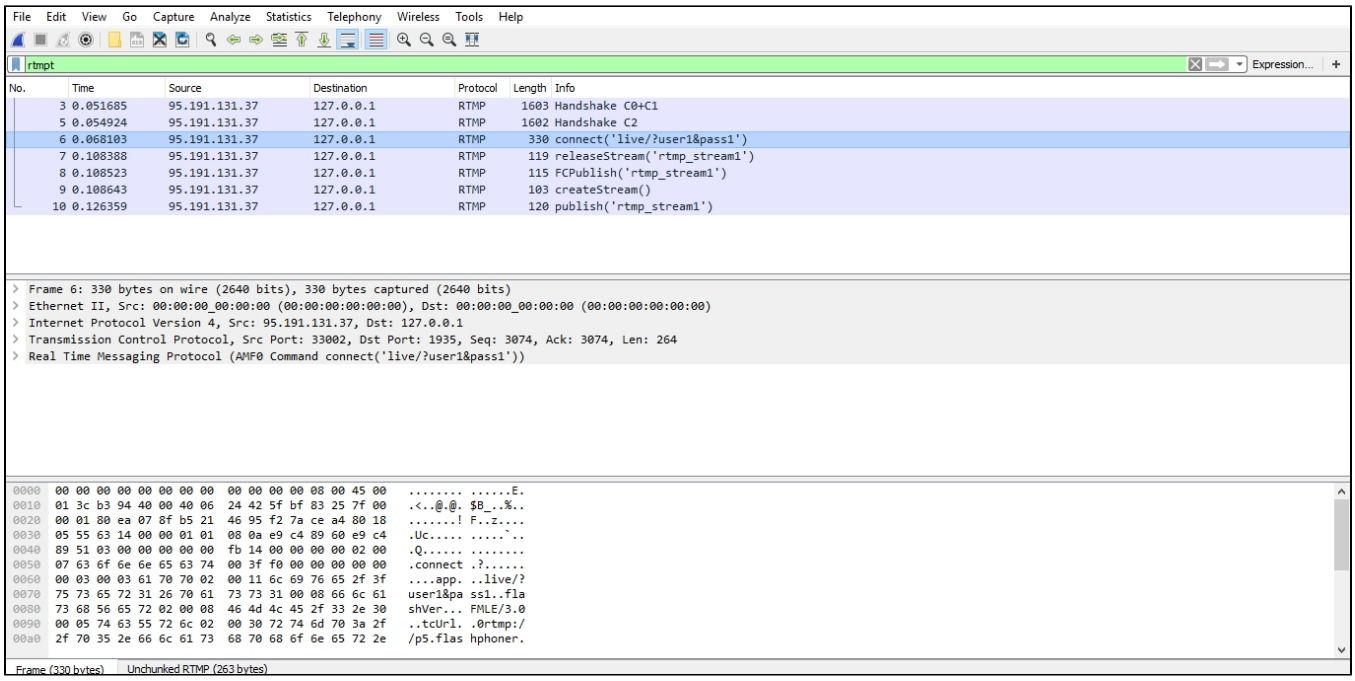
```
rtmp://myrtmpserver.com:1935/app_name/app_instance/?user=user1&pass=pass1
```

Здесь

- myrtmpserver.com - имя RTMP-сервера
- app_name - имя приложения на RTMP-сервере
- app_instance - имя экземпляра приложения наRTMP-сервере

Имя потока указывается в параметре REST-запроса /push/startup 'streamName' или в соответствующей опции при создании потока.

Пример установки RTMP-соединения с передачей параметров запроса



Передача имени потока в URL

В некоторых случаях имя потока при публикации необходимо передать в URL. Для этого необходимо указать настройку в файлеflashphoner.properties

```
rtmp_transponder_full_url=true
```

Тогда, для публикации в параметреREST-запроса /push/startup 'rtmpUrl' или в соответствующей опции при создании потока указывается URL вида

```
rtmp://myrtmpserver.com:1935/app_name/stream_name
```

или, для публикации в другой экземпляр приложения

```
rtmp://myrtmpserver.com:1935/app_name/app_instance/stream_name
```

В этом случаепараметр REST-запроса /push/startup 'streamName' или соответствующая опция при создании потока игнорируется.

Автоматическая ретрансляция на указанный сервер (не для рабочего окружения)

WCS-сервер может автоматически ретранслировать все публикуемые на нем потоки на заданный RTMP-сервер. Для того, чтобы активировать эту возможность, необходимо в файлеflashphoner.propertiesуказать следующие настройки:

```
rtmp_push_auto_start=true
rtmp_push_auto_start_url=rtmp://rtmp.server.com:1935/
```

Здесь rtmp.server.com - имя RTMP-сервера, на который должны ретранслироваться все потоки с WCS.

Этот функционал может использоваться для отладки, но не предназначен для рабочего окружения.

Автоматическое восстановление соединения при закрытии канала

Во время публикации RTMP-потока на другой RTMP-сервер, соединение может быть прервано и канал закрыт по различным причинам (перезапуск принимающего сервера, сетевые проблемы и т.п.). В этом случае может быть настроено автоматическое восстановление соединения и повторная публикация RTMP-потока при помощи параметрав файлеflashphoner.properties

```
rtmp_push_restore=true
```

Необходимо также настроить количество попыток восстановить соединение и интервал между попытками:

```
rtmp_push_restore_attempts=3
rtmp_push_restore_interval_ms=5000
```

В данном случае будет предпринято 3 попытки повторного подключения к RTMP серверу с интервалом 5 секунд. После этого попытки восстановить соединение прекращаются.

Буферизация исходящего RTMP-потока

В сборке5.2.700добавлена возможность буферизации исходящего RTMP потока. Это увеличивает задержку трансляции, но позволяет получить более плавное воспроизведение потока на сервере, куда поток ретранслируется. Буферизация включается при помощи параметра

```
rtmp_out_buffer_enabled=true
```

Могут быть настроены следующие параметры буферизации

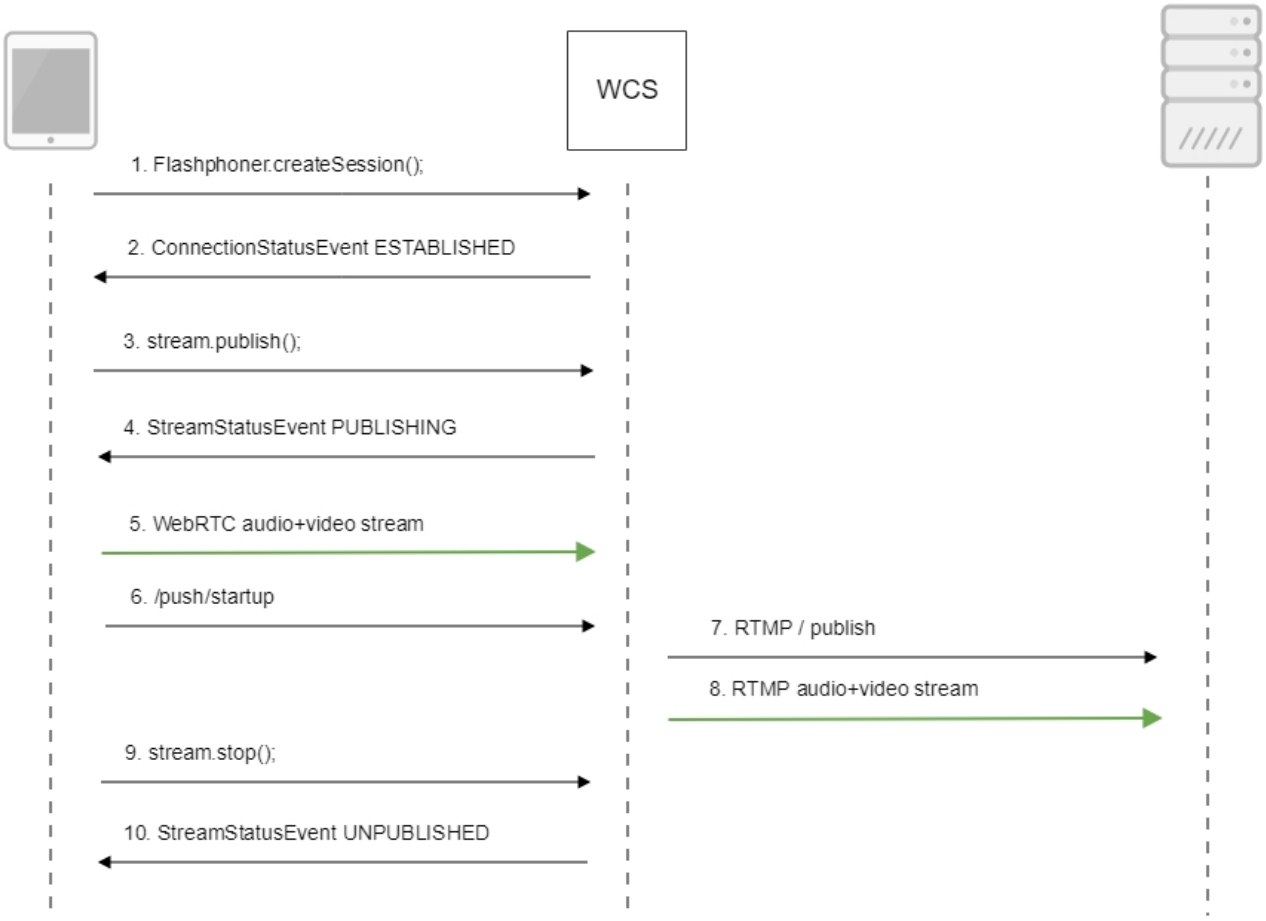
Параметр	Значение по умолчанию	Описание
rtmp_out_buffer_start_size	300	Размер буфера при запуске буферизации для потока, мс
rtmp_out_buffer_initial_size	2000	Первоначальный размер буфера, распределяемый при инициализации, мс
rtmp_out_buffer_polling_time	50	Периодичность проверки буфера, мс
rtmp_out_buffer_max_bufferings_allowed	-1	Количество буферизаций для потока, по умолчанию не ограничено

Последовательность выполнения операций (Call Flow)

Ниже описана последовательность вызовов при использовании примера Two Way Streaming для публикации потока и REST-клиента для отправки запроса /push/startup

two_way_streaming.html

two_way_streaming.js



1. Установка соединения с сервером.

Flashphoner.createSession();code

```
    Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function (session) {
        setStatus("#connectStatus", session.status());
        onConnected(session);
    }).on(SESSION_STATUS.DISCONNECTED, function () {
        setStatus("#connectStatus", SESSION_STATUS.DISCONNECTED);
        onDisconnected();
    }).on(SESSION_STATUS.FAILED, function () {
        setStatus("#connectStatus", SESSION_STATUS.FAILED);
        onDisconnected();
    });
```

2. Получение от сервера события, подтверждающего успешное соединение.

ConnectionStatusEvent ESTABLISHEDcode

```
    Flashphoner.createSession({urlServer: url}).on(SESSION_STATUS.ESTABLISHED, function (session) {
        setStatus("#connectStatus", session.status());
        onConnected(session);
    }).on(SESSION_STATUS.DISCONNECTED, function () {
        ...
    }).on(SESSION_STATUS.FAILED, function () {
        ...
    });
```

3. Публикация потока.

stream.publish();code

```
   session.createStream({
        name: streamName,
        display: localVideo,
        cacheLocalResources: true,
        receiveVideo: false,
        receiveAudio: false
        ...
    }).publish();
```

4. Получение от сервера события, подтверждающего успешную публикацию потока.

StreamStatusEvent, статус PUBLISHINGcode

```
   session.createStream({
        name: streamName,
        display: localVideo,
        cacheLocalResources: true,
        receiveVideo: false,
        receiveAudio: false
    }).on(STREAM_STATUS.PUBLISHING, function (stream) {
        setStatus("#publishStatus", STREAM_STATUS.PUBLISHING);
        onPublishing(stream);
    }).on(STREAM_STATUS.UNPUBLISHED, function () {
        ...
    }).on(STREAM_STATUS.FAILED, function () {
        ...
    }).publish();
```

5. Отправка аудио-видео потока по WebRTC

6. Отправка запроса /push/startup

```
http://demo.flashphoner.com:9091/rest-api/push/startup
{
 "streamName": "testStream",
 "rtmpUrl": "rtmp://demo.flashphoner.com:1935/live/testStream"
}
```

7. Установка соединения по RTMP с указанным сервером, публикация потока

8. Отправка аудио-видео потока по RTMP

9. Остановка публикации потока.

stream.stop();code

```
function onPublishing(stream) {
    $("#publishBtn").text("Stop").off('click').click(function () {
        $(this).prop('disabled', true);
        stream.stop();
    }).prop('disabled', false);
    $("#publishInfo").text("");
}
```

10. Получение от сервера события, подтверждающего остановку публикации потока.

StreamStatusEvent, статус UNPUBLISHEDcode

```
session.createStream({
     name: streamName,
     display: localVideo,
     cacheLocalResources: true,
     receiveVideo: false,
     receiveAudio: false
 }).on(STREAM_STATUS.PUBLISHING, function (stream) {
     ...
 }).on(STREAM_STATUS.UNPUBLISHED, function () {
     setStatus("#publishStatus", STREAM_STATUS.UNPUBLISHED);
     onUnpublished();
 }).on(STREAM_STATUS.FAILED, function () {
     ...
 }).publish();
```

Известные проблемы

1. При републикации потока на другой RTMP-сервер и проигрывании потока с этого сервера в плеереJWPlayer. соотношение сторон картинки может быть искажено

Симптомы: соотношение сторон при проигрывании потока с RTMP-сервера отличается от опубликованного потока

Решение: включить отправку метаданных при републикации

```
rtmp_transponder_send_metadata=true
```

2. Републикация может не работать, если RTMP-сервер, на который ретранслируется поток, требует определенную версию Flash

Симптомы: не проходит RTMP handshake, канал закрывается с ошибкой (RTMP error) в логе WCS-сервера

Решение: задать версию Flash RTMP-клиента, используя либо настройку rtmp_flash_ver_subscriber в flashphoner.properties, либо параметр rtmpFlashVersion в REST-вызове для ретрансляции

Например, для републикации на Periscope:

```
rtmp_flash_ver_subscriber = LNX 76.219.189.0
```

3. Сервер, на который ретранслируется поток, может требовать определенные параметры потока: битрейт, частоту кадров и интервал ключевых кадров видео

Симптомы: например, Periscope показывает предупреждения о несоответствии рекомендуемым парамерами

Решение: задать требуемые ограничения для исходного потока (например, для битрейта аудио) и параметры в REST-вызове для ретрансляции (keyFrameInterval и fps)

4. При ретрансляции FullHD, 2K, 4K потоков с большими размерами кадров, пакеты могут не помещаться в буфер сокета на отправку, из-за чего в некоторых плеерах могут наблюдаться артефакты

Симптомы: при проигрывании ретранслированного потока на хорошем канале эпизодически появляются артефакты

Решение: включить буферизацию RTMP пакетов при отправке настройкой

```
rtmp.server_buffer_enabled=true
```