

# CDN 2.1

- Описание
- Группировка узлов CDN
- Выделение узлов под задачи транскодинга
  - Профили транскодинга
    - Параметры профиля
    - Управление параметрами кодирования
    - Тонкая настройка кодирования
    - Поддержка нескольких каналов звука
  - Управление профилями транскодинга при помощи REST API
    - REST-методы и статусы ответа
    - Параметры
    - Изменение используемых профилей
  - В каких случаях поток транскодируется на Edge сервере
  - Сохранение соотношения сторон видео при транскодировании по профилю
  - Настройка транскодеров для работы в условиях больших нагрузок
    - Приостановка декодирования входящих потоков
    - Использование библиотеки jemalloc для распределения памяти
    - Снижение порога многопоточного кодирования
    - Уменьшение количества выделений памяти при декодировании
      - Освобождение памяти, выделенной под неиспользуемые буфера
  - Транскодирование по двум и более профилям с одинаковыми параметрами видео
  - Предотвращение транскодирования к более высоким разрешениям
    - Ограничения
- Выбор маршрута для воспроизведения потока
  - Пример распределения потоков в CDN с использованием Transcoder узлов
  - Группировка узлов и транскодинг
  - Получение информации о маршрутах с помощью REST API
    - REST-методы и статусы ответа
    - Параметры
  - Примеры разбора маршрутов воспроизведения потока
    - Воспроизведение потока с Origin без транскодинга при совпадении профилей публикации и воспроизведения
    - Воспроизведение потока с Transcoder с транскодингом звука
    - Воспроизведение потока с Transcoder с транскодингом видео
  - Анонсирование потока в CDN по получению ключевого кадра
- Распределение нагрузки по Transcoder узлам
  - Определение загрузки узла, исходя из загрузки процессора
  - Ограничение количества кодировщиков (encoders) видео на Transcoder узлах
  - Ограничение общей нагрузки на декодирование
  - Ограничение общей нагрузки на кодирование
  - Ограничение процента деградации потоков
  - Равномерное распределение нагрузки по узлам
  - Предотвращение транскодирования на Origin
  - Управление текущим состоянием узла при помощи REST API
    - REST-методы и статусы ответа
    - Параметры
  - Вычисление текущего состояния узла и его рассылка
    - Настройка состояния Transcoder узла при достижении ограничений по производительности
- Авторизация узлов в CDN
- Обратная совместимость с CDN 2.0
- Ограничения
- Принудительное указание версии CDN, поддерживаемой узлом
- Известные проблемы

## Описание

Новая реализация CDN 2.1 отличается от [CDN 2.0](#) дополнительными возможностями:

- разделение узлов CDN по географическому (EU, US) или другому признаку при помощи групп
- выделение узлов под задачи транскодинга, назначая таким узлам роль Transcoder

## Группировка узлов CDN

Серверы в CDN могут быть объединены в группу по географическому (расположение), аппаратному (использование GPU для транскодинга) и другим признакам. Группа указывается в файле настроек [flashphoner.properties](#) при помощи параметра

```
cdn_groups=group1
```

Один и тот же сервер может входить в несколько групп. Допустим, сервер находится в Европе и используется для транскодинга при помощи GPU:

```
cdn_groups=EU,GPU
```

Группы узлов используются при выборе маршрута для воспроизведения потока

Серверы, для которых не задана группа, считаются принадлежащими к одной группе

## Выделение узлов под задачи транскодинга

Чтобы не загружать серверы, используемые для публикации и просмотра видео в крупной разветвленной CDN, задачами транскодинга, целесообразно выделить под эти задачи узлы с повышенной вычислительной мощностью. Для таких узлов, кроме Origin и Edge, добавлена роль Transcoder.

```
cdn_role=transcoder
```

Узел Transcoder не должен использоваться для публикации или воспроизведения потока. Он взаимодействует с Origin и Edge серверами следующим образом:

1. Поток публикуется на Origin сервер.
2. Transcoder забирает с Origin поток по запросу Edge сервера.
3. Transcoder транскодирует поток в соответствии с настройками, указанными Edge сервером.
4. Edge забирает с Transcoder поток для воспроизведения.

Конкретный узел для транскодирования выбирается при выборе маршрута для воспроизведения потока.

## Профили транскодинга

Для настройки транскодинга на Edge сервере используются профили. Файл профилей транскодинга `cdn_profiles.yml` должен располагаться в каталоге `/usr/local/FlashphonerWebCallServer/conf` на Edge сервере:

```
profiles:

- webrtc-144:
  audio:
    codec : opus
    rate  : 48000
  video:
    width  : 256
    height : 144
    codecImpl : OPENH264
```

При запросе на воспроизведение потока, транскодированного по указанному профилю, имя профиля должно быть добавлено к имени потока, например

```
test-webrtc-144
```

Рекомендуется для удобства задавать имена профилей, начинающиеся с дефиса.

Для применения изменений в файле `cdn_profiles.yml` необходим перезапуск сервера.

## Параметры профиля

Параметр	Допустимые значения	Описание
Параметры аудио		
codec	opus mpeg4-generic speex ulaw	Используемый кодек аудио

bitrate	Зависит от требуемого качества и пропускной способности каналов	Битрейт звука, бит/с
rate	8000 11025 12000 16000 22050 24000 32000 44100 48000	Частота дискретизации звука
channels	1 2	Количество каналов
<b>Параметры видео</b>		
codec	h264 mpv vp8	Используемый кодек видео
bitrate	Зависит от требуемого качества и пропускной способности каналов	Битрейт видео, кбит/с
width	Зависит от требуемого качества и пропускной способности каналов	Ширина картинки
height	Зависит от требуемого качества и пропускной способности каналов	Высота картинки
codecmpl	FF OPENH264	Используемый кодек видео, по умолчанию FF
gop	Зависит от требуемого качества и пропускной способности каналов	Частота отправки ключевых кадров (GOP)
fps	Зависит от требуемого качества и пропускной способности каналов	Частота кадров
quality	Зависит от требуемого качества и пропускной способности каналов	Качество видео
preset	ultrafast superfast veryfast faster fast medium slow slower veryslow placebo	Пресет FF, по умолчанию ultrafast
profile	66 (Baseline) 77 (Main) 88 (Extended) 100 (High)	Профиль кодирования FF, по умолчанию 66 (Baseline)
level	Зависит от требуемого качества	Уровень кодирования FF, по умолчанию 3.1

Если ширина картинки не указана или равна 0, видео масштабируется по указанной высоте, с [сохранением соотношения сторон](#).

Если ширина и высота картинки обе равны 0, видео масштабируется к разрешению 160x120.

## Управление параметрами кодирования

При использовании кодировщика FF можно выбрать предустановки (пресет) кодирования, профиль и уровень кодирования. Например, следующие параметры

```
profile1:
  audio:
    codec : opus
    rate  : 48000
  video:
    width  : 640
    height : 360
    gop    : 90
    fps    : 30
    codec  : h264
    profile : 77
    level  : 31
    preset : veryfast
```

задают использование пресета `veryfast`, профиля `Main` и уровня `3.1`

## Тонкая настройка кодирования

По умолчанию, кодирование видео производится одним потоком. При использовании относительно медленных пресетов с высоким качеством кодирования (например, пресет `fast`) возникает необходимость кодирования в несколько потоков, в противном случае на стороне подписчика видео может играть с низким FPS и рассинхронизацией. Начиная со сборки [5.2.347](#), для того, чтобы автоматически запускать кодирование в несколько потоков в зависимости от разрешения, заказанного подписчиком, предусмотрена настройка

```
video_encoder_second_thread_threshold=777000
```

Данная настройка задается как произведение высоты картинки на ширину. По умолчанию, видеопотоки 720p и выше будут кодироваться в два потока. Для того, чтобы уменьшить этот порог до 480p, необходимо установить настройку как  $640 * 480 = 307200$

```
video_encoder_second_thread_threshold=307200
```

В этом случае видеопотоки 480p и более высокого разрешения будут кодироваться в два потока.

Количество потоков для кодирования задается настройкой

```
video_encoder_max_threads=2
```

По умолчанию, максимальное количество потоков кодирования равно 2.

Вышеописанные настройки указываются на Transcoder узлах.

## Поддержка нескольких каналов звука

Начиная со сборки [5.2.773](#), поддерживается транскодирование звука по профилю с указанием количества каналов (1 - моно, 2 - стерео)

```
profiles:
  -240p:
    audio:
      codec : mpeg4-generic
      rate  : 48000
      channels: 1
    video:
      height : 240
      bitrate : 300
      gop    : 50
      codec  : h264
```

Для того, чтобы подписчик на Edge сервере мог играть стерео звук при публикации потока со стерео звуком, количество каналов в профиле должно соответствовать количеству каналов в оригинальном потоке, т.е.

```

profiles:
-240p:
  audio:
    codec : mpeg4-generic
    rate : 48000
    channels: 2

```

Если кодек, частота дискретизации и количество каналов профиля совпадают с параметрами звуковой дорожки в оригинальном потоке, то звук будет передан на Edge сервер без транскодирования.

## Управление профилями транскодинга при помощи REST API

Для управление профилями транскодинга на лету, без перезапуска сервера, служит специальное REST API

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

- HTTP: <http://test.flashphoner.com:8081/rest-api/cdn/profile/print>
- HTTPS: <https://test.flashphoner.com:8444/rest-api/cdn/profile/print>

Здесь:

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

REST запросы необходимо отправлять на Edge сервер. Изменения в профилях сохраняются в памяти и применяются немедленно, а также записываются в файл `cdn_profiles.yml` для использования после перезапуска сервера.

## REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/cdn/profile/add	<pre> {   "name": "-profile1",   "profile": {     "audio": {       "type":         "audio",       "bitrate": 0,       "codec":         "opus",       "rate": 48000     },     "video": {       "type":         "video",       "bitrate": 0,       "codec":         "h264",       "codecImpl":         "FF",       "fps": 30,       "gop": 90,       "height":         360,       "quality": 0,       "width": 640     }   } } </pre>		200 – OK 400 - Bad Request 409 - Conflict 500 – Internal Server Error	Добавить профиль транскодинга

/cdn/profile /modify	<pre>{   "name": "-profile1",   "profile": {     "video": {       "bitrate": 400,       "quality": 25     }   } }</pre>		200 – OK 400 - Bad Request 404 - Not Found 500 – Internal Server Error	Изменить профиль транскодинга
/cdn/profile /print		<pre>"-240p": {   "audio": {     "bitrate": 0,     "channels": 0,     "codec": "opus",     "rate": 48000,     "type": "audio"   },   "video": {     "bitrate": 0,     "codec": "h264",     "codecImpl": "FF",     "fps": 25,     "gop": 50,     "height": 240,     "preset": "veryfast",     "quality": 0,     "type": "video",     "width": 0   } }, "-480p": {   "audio": {     "bitrate": 0,     "channels": 0,     "codec": "opus",     "rate": 48000,     "type": "audio"   },   "video": {     "bitrate": 0,     "codec": "h264",     "codecImpl": "FF",     "fps": 25,     "gop": 50,     "height": 480,     "preset": "veryfast",     "quality": 0,     "type": "video",     "width": 0   } } }</pre>	200 – OK 404 - Not Found 500 – Internal Server Error	Получить список профилей транскодирования

/cdn/profile /remove	<pre>{   "name": "-profile1" }</pre>		200 – OK  404 - Not Found  500 – Internal Server Error	Удалить профиль транскодирования
-------------------------	--	--	--	----------------------------------

Параметры

Имя параметра	Описание	Пример
name	Имя профиля	-profile1
profile	Параметры профиля	<pre>{   "audio": {     "type": "audio"     "bitrate": 0,     "channels": 0,     "codec": "opus",     "rate": 48000,   },   "video": {     "type": "video",     "bitrate": 0,     "codec": "h264",     "codecImpl": "FF",     "fps": 30,     "gop": 90,     "height": 360,     "quality": 0,     "width": 640   } }</pre>

Параметры профиля обязательно должны содержать разделыaudio или video.

Основными параметрами профиля являются следующие:

- audio:
  - bitrate
  - codec
  - rate
- video:
  - bitrate
  - codec
  - width
  - height
  - quality

Если какие-либо из этих параметров не указаны при создании профиля, они будут добавлены в профиль со значением 0 для числовых и "" для строковых параметров, при этом будут применены значения по умолчанию.

Остальные параметры являются дополнительными и могут отсутствовать в профиле:

- video:
  - codecImpl
  - fps
  - gop
  - level
  - preset
  - profile

При изменении профиля изменяются только переданные параметры. Для сброса какого-либо из основных параметров к значению по умолчанию, либо для удаления из профиля дополнительного параметра необходимо передать -1, например:

```
{
  "name": "-profile1",
  "profile": {
    "video": {
      "bitrate": -1,
      "codecImpl": -1
    }
  }
}
```

## Изменение используемых профилей

Если по профилю в данный момент воспроизводится поток, изменение профиля возможно только после окончания воспроизведения потока и остановки агента, т.е. приблизительно через минуту после того, как от потока отключился последний подписчик.

## В каких случаях поток транскодируется на Edge сервере

Транскодинг потока на Edge сервере включается в следующих случаях:

1. Если аудио кодека или частоты дискретизации, указанных в настройках профиля, нет в SDP подписчика, аудио дорожка будет транскодирована в один из форматов, поддерживаемых подписчиком.
2. Если заданы граничные параметры для подписчика: например, высота или битрейт видео. Поэтому при транскодировании по профилю желаемые граничные параметры следует указывать в профиле и не задавать для подписчиков.

Для подписчика следует применять профиль, соответствующий технологии, используемой подписчиком. Например, для подписчика, использующего WSPayer, применить профиль с кодеками ulaw и mpv, а для RTMP - mpeg4-generic и h264.

## Сохранение соотношения сторон видео при транскодировании по профилю

Если на всех узлах CDN включено [сохранение соотношения сторон при транскодировании](#) (по умолчанию), то при транскодировании потока на Transcoder узле также будет сохранено исходное соотношение сторон, с которым этот поток был опубликован на Origin. Например, если к потоку 640x480 (4:3) применяется профиль 16:9

```
profile1:
  audio:
    codec : opus
    rate  : 48000
  video:
    width : 320
    height : 180
    gop   : 90
    fps   : 30
    codec : h264
```

поток будет транскодирован в 320x240 (4:3).

В этом случае можно не указывать ширину кадра в профиле транскодирования, поскольку ширина будет выбрана, исходя из соотношения сторон. Например, допускается такой профиль

```
profile3:
  audio:
    codec : opus
    rate  : 48000
  video:
    height : 180
```

## Настройка транскодеров для работы в условиях больших нагрузок

При транскодинге большого количества потоков на сервере возрастает нагрузка на процессор сервера. Если процессор не успевает обрабатывать кадры всех потоков, очереди кадров на кодирование занимают оперативную память сервера (не JVM heap!). В зависимости от используемых в системе библиотек распределения памяти (malloc) это может приводить к утечкам памяти и, как следствие, к остановке сервера.

Чтобы избежать этого, предусмотрены следующие настройки, которые рекомендуется использовать совместно:



- приостановка декодирования входящих потоков, если очереди на кодирование переполнены
- использование библиотеки jemalloc

## Приостановка декодирования входящих потоков

Приостановка декодирования входящих потоков дает возможность сгладить (но не ликвидировать) пиковые нагрузки на сервер, при условии выставления этих настроек для группы транскодеров, и включается при помощи следующего параметра

```
streaming_video_decoder_wait_for_distributors=true
```

Размер очереди, по достижении которого декодирование приостанавливается, задается в кадрах следующей настройкой

```
streaming_video_decoder_wait_for_distributors_max_queue_size=3
```

Интервал ожидания освобождения очереди задается в миллисекундах

```
streaming_video_decoder_wait_for_distributors_timeout=33
```

## Использование библиотеки jemalloc для распределения памяти

По умолчанию, в большинстве случаев используется библиотека распределения памяти, входящая в состав glibc. Чтобы избежать утечек памяти под большими нагрузками, рекомендуется переключить Transcoder узлы на использование библиотеки [jemalloc](#) следующим образом (на примере Centos 7):

1. Установить компоненты, необходимые для сборки

```
yum -y install autoconf libtool pkg-config g++ make cmake bzip
```

2. Скачать исходный код библиотеки

```
wget https://github.com/jemalloc/jemalloc/releases/download/5.2.1/jemalloc-5.2.1.tar.bz2
```

3. Распаковать архив

```
tar -xvzf jemalloc-5.2.1.tar.bz2
```

4. Собрать библиотеку

```
cd jemalloc-5.2.1
./configure && make && make install
```

5. В файле /usr/local/FlashphonerWebCallServer/bin/setenv.sh закомментировать следующую строку

```
export MALLOC_ARENA_MAX=4
```

и добавить следующую строку

```
export LD_PRELOAD=/usr/local/lib/libjemalloc.so
```

6. В файле настроек [flashphoner.properties](#) переключить используемый декодер на OpenH264

```
decoder_priority=OPENH264,FF
```

7. Перезапустить WCS.

## Снижение порога многопоточного кодирования

По умолчанию, многопоточное кодирование включается для профилей 720p. В некоторых случаях необходимо снизить этот порог, чтобы кодировать по профилям 480p также в два и более потоков

```
video_encoder_second_thread_threshold=408950
```

## Уменьшение количества выделений памяти при декодировании

Для уменьшения количества выделений памяти при декодировании в сборке [5.2.559](#) добавлена возможность использования пула буферов для декодированных картинок. Эта возможность включается настройкой

```
decoder_buffer_pool=true
```

В этом случае под каждое разрешение картинки выделяется пул буферов. По окончании использования буфера, он не удаляется, а возвращается в пул. Чтобы получить информацию об использовании памяти, необходимо включить вывод статистики при помощи настройки

```
decoder_buffer_pool_stats=true
```

Статистика доступна по запросу (только в формате JSON)

```
curl -s 'http://localhost:8081/?action=stat&format=json&groups=decoder_buffer_pool_stats'
```

и сгруппирована по пулам

```
{
  "decoder_buffer_pool_stats": {
    "decoder_buffer_pool_info": {
      "decoder_buffer_pool_info_pools": {
        "3072x1536": {
          "decoder_buffer_pool_width": 3072,
          "decoder_buffer_pool_height": 1536,
          "decoder_buffer_pool_size_bytes": 84934656,
          "decoder_buffer_pool_leased": 6,
          "decoder_buffer_pool_allocated": 12
        },
        "1280x720": {
          "decoder_buffer_pool_width": 1280,
          "decoder_buffer_pool_height": 720,
          "decoder_buffer_pool_size_bytes": 688584704,
          "decoder_buffer_pool_leased": 0,
          "decoder_buffer_pool_allocated": 3605
        }
      },
      "decoder_buffer_pool_total_size_bytes": 773519360
    }
  }
}
```

Здесь

- decoder\_buffer\_pool\_total\_size\_bytes - общий размер памяти, выделенной под пулы, в байтах
- decoder\_buffer\_pool\_width - высота картинки
- decoder\_buffer\_pool\_height - ширина картинки
- decoder\_buffer\_pool\_size\_bytes - размер памяти, выделенной под пул, в байтах
- decoder\_buffer\_pool\_leased - количество используемых буферов
- decoder\_buffer\_pool\_allocated - количество выделенных буферов в пуле

При необходимости, память, выделенная под неиспользуемые буфера, может быть очищена при помощи REST API запроса `/debug/decoder/shrink_buffer_provider_pools`

## Освобождение памяти, выделенной под неиспользуемые буфера

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

- HTTP:[http://test.flashphoner.com:8081/rest-api/debug/decoder/shrink\\_buffer\\_provider\\_pools](http://test.flashphoner.com:8081/rest-api/debug/decoder/shrink_buffer_provider_pools)
- HTTPS:[https://test.flashphoner.com:8444/rest-api/debug/decoder/shrink\\_buffer\\_provider\\_pools](https://test.flashphoner.com:8444/rest-api/debug/decoder/shrink_buffer_provider_pools)

Здесь:

- test.flashphoner.com - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /debug/decoder/shrink\_buffer\_provider\_pools- используемый REST-метод

REST запросы необходимо отправлять на Transcoder сервер.

REST-метод	Статусы ответа	Описание
/debug/decoder/shrink_buffer_provider_pools	200 – OK	Сжать пулы буферов декодирования
	500 – Internal Server Error	

## Транскодирование по двум и более профилям с одинаковыми параметрами видео

На Edge узлах могут быть определены несколько профилей с одинаковыми параметрами видео, но различными параметрами аудио, например для воспроизведения по HLS и WebRTC:

```
-240p-HLS:
  audio:
    codec: mpeg4-generic
  video:
    height: 240
    bitrate: 300
    codec: h264
-240p-WebRTC:
  audio:
    codec: opus
  video:
    height: 240
    bitrate: 300
    codec: h264
```

В этом случае, если поток транскодируется на одном и том же Transcoder узле по двум таким профилям, будет использован один кодировщик видео и два кодировщика аудио.

## Предотвращение транскодирования к более высоким разрешениям

В сборке [5.2.607](#) добавлена возможность избежать транскодирования потока к более высоким разрешениям. Эта возможность включается при помощи настройки

```
cdn_strict_transcoding_boundaries=true
```

В этом случае, если с Edge сервера запросить поток, опубликованный на Origin с разрешением 640x480, по профилю 720p, Edge получит оригинальный поток напрямую с Origin без транскодирования. Сравнение разрешений производится по высоте.

При необходимости, можно запретить запрашивать с Edge поток по профилям с более высоким разрешением при помощи параметра

```
cdn_strict_transcoding_throws_exception=true
```

Тогда при попытке запросить с Edge поток, опубликованный на Origin с разрешением 640x480, по профилю 720p, Edge не сможет выбрать маршрут для воспроизведения потока, и клиент получит ошибку.

Эти настройки должны указываться на Edge сервере.

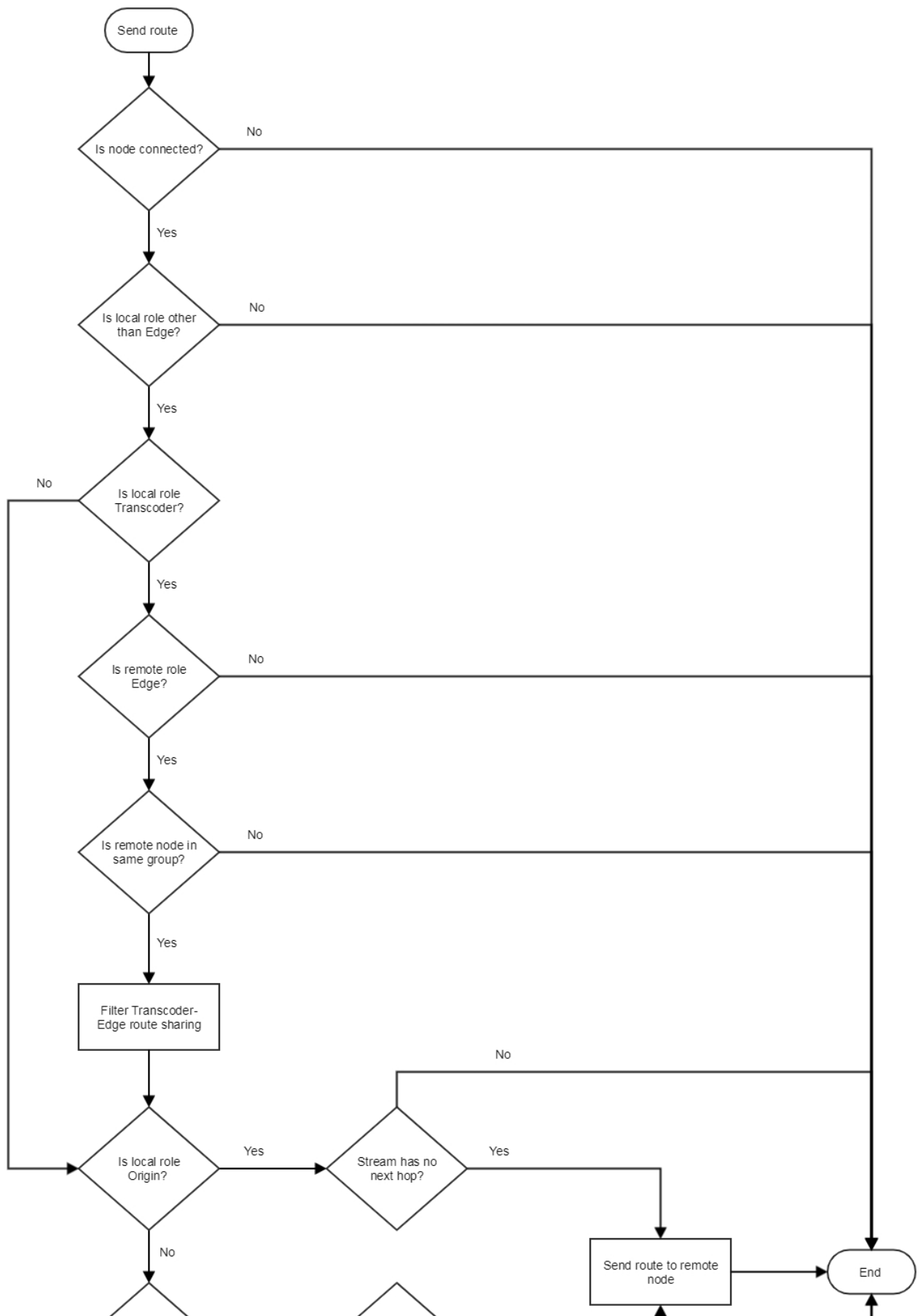
## Ограничения

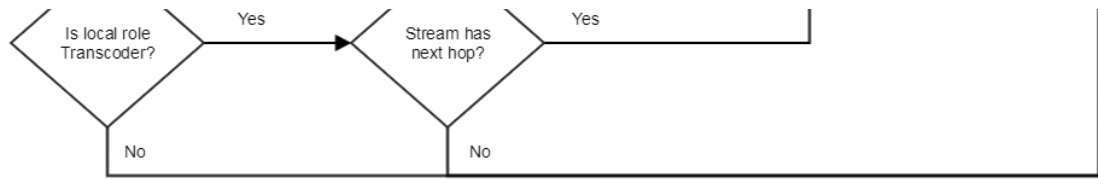
1. Если транскодирование к более высоким разрешениям запрещено, в настройках профиля транскодирования обязательно должна быть указана высота.

## Выбор маршрута для воспроизведения потока

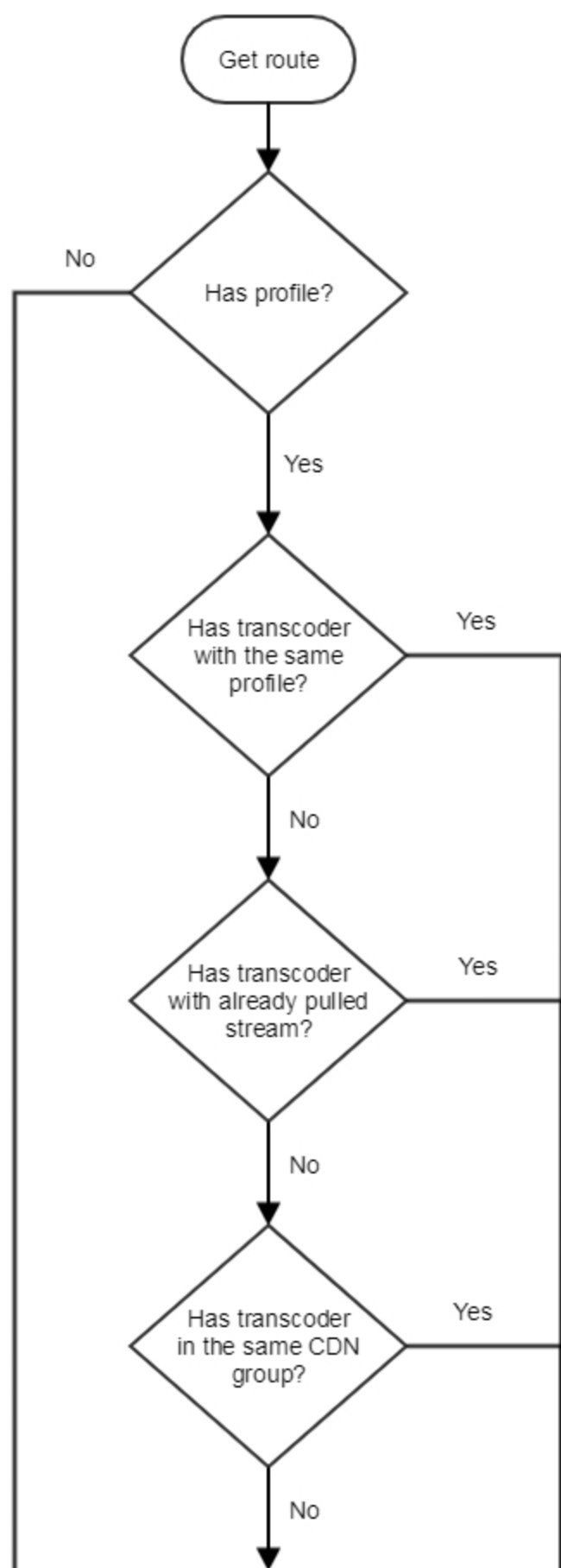
Маршруты в CDN строятся на основе периодических рассылок следующих данных:

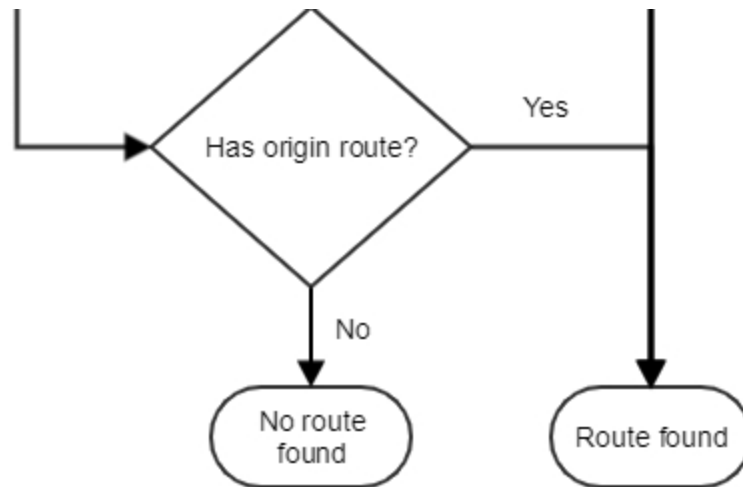
- Origin рассылает узлам с ролью Transcoder и Edge информацию об опубликованных потоках;
- Transcoder рассылает узлам с ролью Edge в пределах той же группы информацию о потоках, которые уже захвачены им с Origin;
- Edge не рассылает ничего и всегда является конечной точкой маршрута.





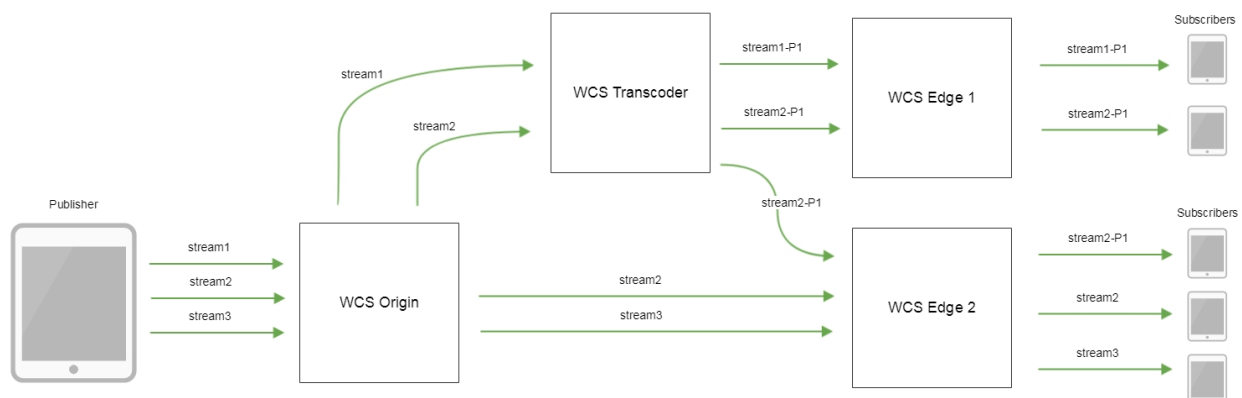
Маршрут для воспроизведения потока на Edge сервере выбирается следующим образом:





1. Если на Edge сервере настроен профиль транскодирования:
  - 1.1. Если на Transcoder в одной группе с Edge есть поток с таким именем:
    - 1.1.1. Если поток уже транскодируется по указанному профилю, Edge забирает поток с Transcoder
    - 1.1.2. Если поток транскодируется по другому профилю:
      - 1.1.2.1. Поток транскодируется по указанному профилю
      - 1.1.2.2. Edge забирает поток с Transcoder
  - 1.2. Если Transcoder в одной группе с Edge может забрать запрошенный поток с Origin:
    - 1.2.1. Transcoder забирает поток с Origin
    - 1.2.2. Поток транскодируется по указанному профилю
    - 1.2.3. Edge забирает поток с Transcoder
  - 1.3. В остальных случаях Edge забирает поток с Origin
2. Если на Edge сервере не настроен профиль транскодирования, Edge забирает поток с Origin

## Пример распределения потоков в CDN с использованием Transcoder узлов



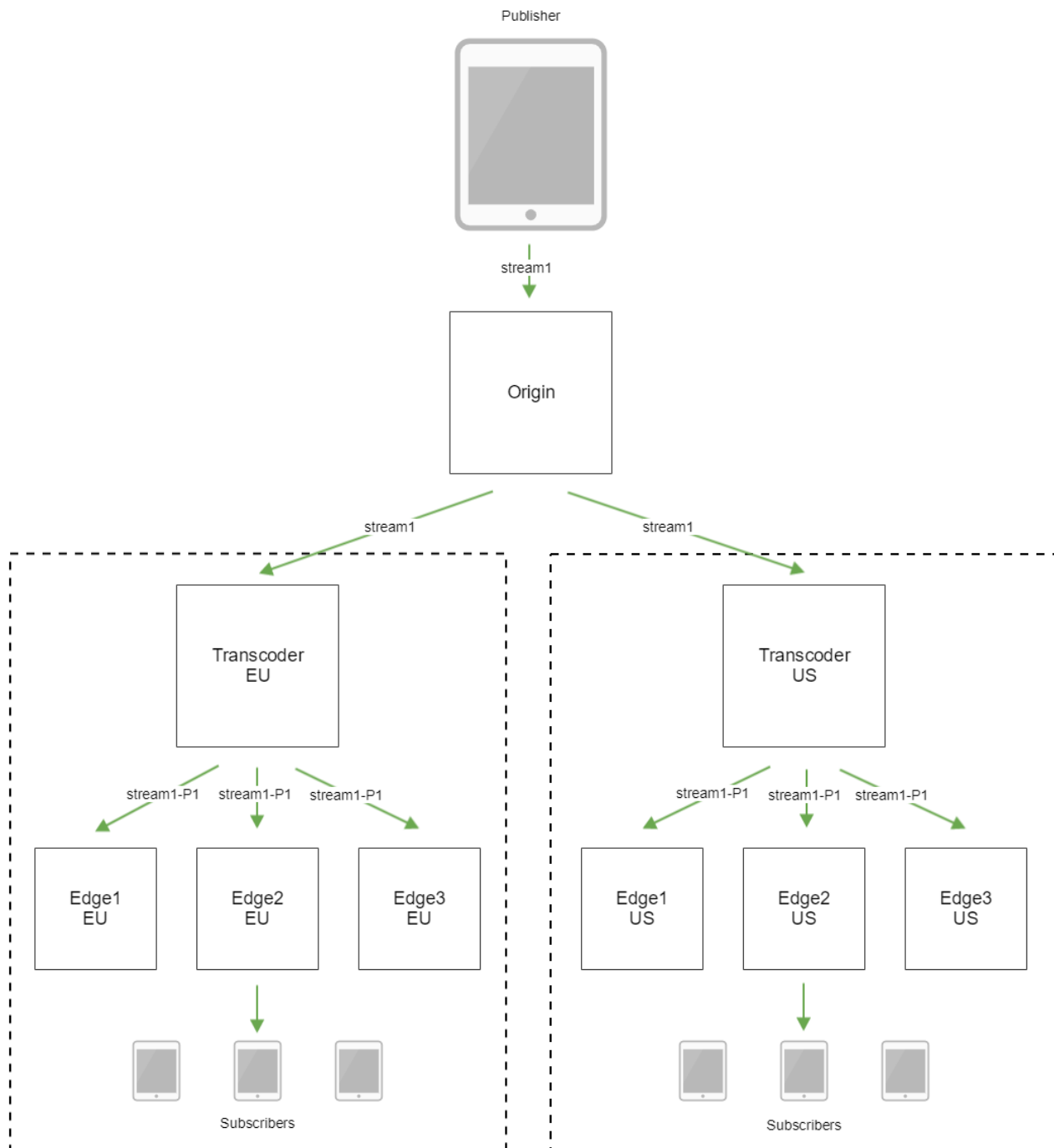
Здесь



- stream1, stream2, stream3 - потоки, опубликованные на Origin
- stream1-P1 - поток stream1, транскодированный по профилю P1
- stream2-P1 -поток stream1, транскодированный по профилю P2

## Группировка узлов и транскодинг

По умолчанию, для транскодирования потока по профилю, Edge выбирает Transcoder в той же группе. Например, если CDN имеет два сегмента, сгруппированных по географическому признаку: в Европе и в США, потокораспределение при запросе одного и того же потока по одному и тому же профилю из каждой группы будет выглядеть следующим образом:



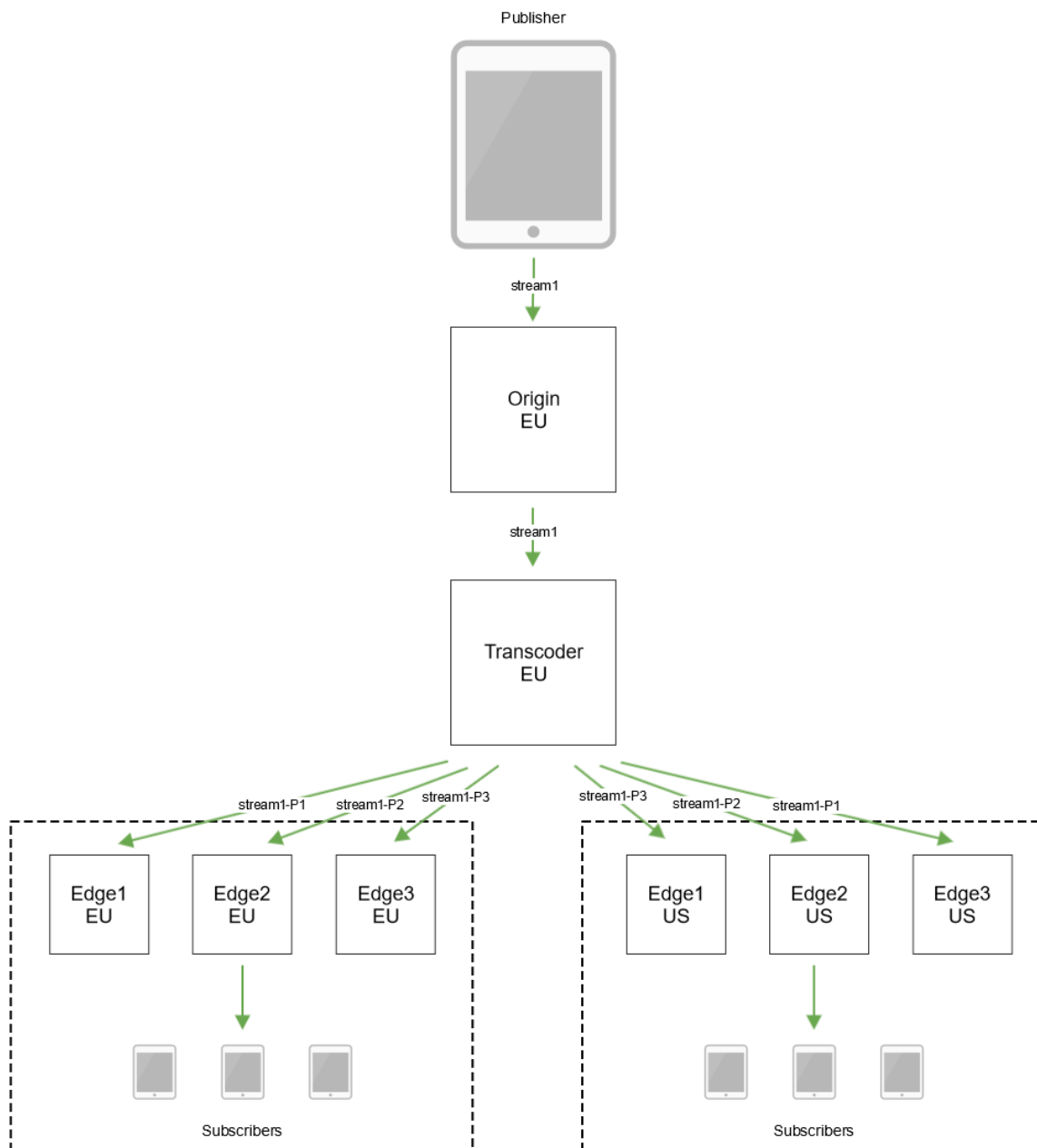
При этом в каждую из групп с Origin транслируется только один поток, что снижает нагрузку на межгрупповые каналы, но транскодинг выполняется в каждой группе независимо.

Данное поведение может быть изменено при помощи настройки

```
cdn_group_origin_to_transcoder_relation=true
```

Эта настройка должна быть включена на Transcoder и Edge-узлах.

Тогда для транскодирования потока в первую очередь будет выбираться Transcoder в одной группе с Origin, потокораспределение будет выглядеть следующим образом:



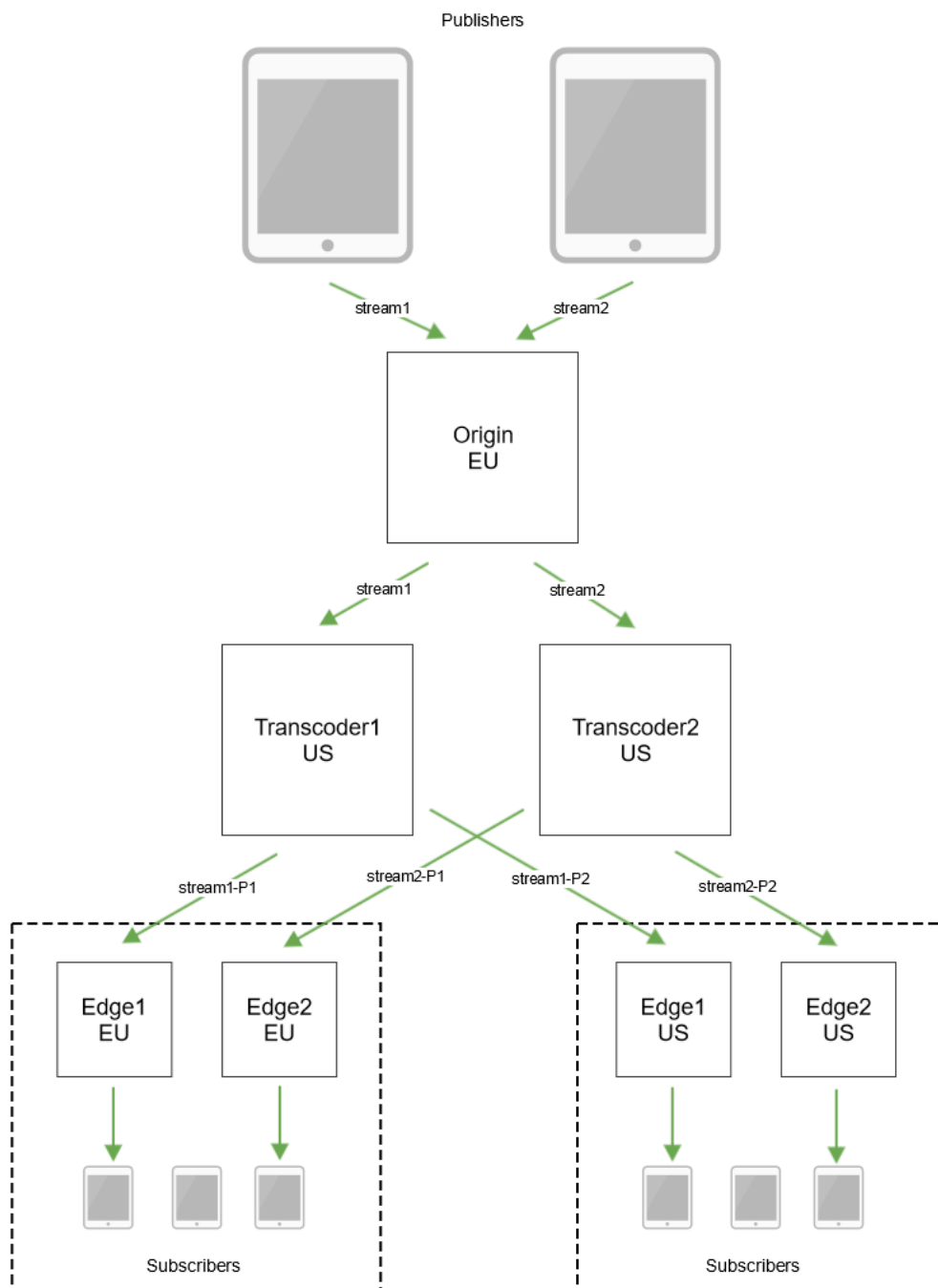
При этом поток транскодируется централизованно для всех географических областей, но в каждую группу будет транслироваться столько потоков, сколько было запрошено с Edge для воспроизведения. Это увеличит нагрузку на межгрупповые каналы.

Начиная с версии [5.2.418](#), если Transcoder из той же группы, что Origin не будет доступен, для транскодирования потока будет выбираться Transcoder из группы Edge, или при недоступности Transcoder и из группы Edge, любой другой доступный Transcoder. Таким образом, Edge сможет забрать транскодированный поток вне зависимости от того, есть ли доступные Transcoder-узлы в группах Origin или Edge.

Приоритет групп при выборе доступного Transcoder для транскодирования нового потока будет следующим:

1. Первая группа, указанная на Origin (другие группы не учитываются - приоритет переходит к группе Edge)
2. Группы Edge (если Transcoder из первой группы недоступен, то приоритет переходит ко второй и так далее)
3. Другие группы (равный приоритет)

Однако, если запрошенный поток уже транскодируется на другом Transcoder, чем приоритетный для этого Edge, то этот поток будет забран с того Transcoder, на котором уже осуществляется транскодирование.



## Получение информации о маршрутах с помощью REST API

Для получения информации о маршрутах используется REST API запрос/cdn/stream/show\_routes

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

- HTTP:[http://test.flashphoner.com:8081/rest-api/cdn/stream/show\\_routes](http://test.flashphoner.com:8081/rest-api/cdn/stream/show_routes)
- HTTPS:[https://test.flashphoner.com:8444/rest-api/cdn/stream/show\\_routes](https://test.flashphoner.com:8444/rest-api/cdn/stream/show_routes)

Здесь:

- test.flashphoner.com - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /cdn/show\_stream\_routes- используемый REST-метод

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/cdn/stream/show_routes	<pre>{   "streamName": "test-webrtc-144" }</pre>	<pre>{   "REQUESTED-PROFILE": [     "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",     "VIDEO{width=256, height=144, gop=null, fps=null, bitrate=0, codec='h264', codecImpl='OPENH264', quality=null}"   ],   "1-PROFILE-192.168.1.220": [     "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",     "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='null', quality=null}",     "VIDEO{width=320, height=180, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='FF', quality=null}",     "VIDEO{width=256, height=144, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='OPENH264', quality=null}"   ],   "2-STREAM-192.168.1.220": [     "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",     "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}",     "VIDEO{width=320, height=180, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='FF', quality=null}",     "VIDEO{width=256, height=144, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='OPENH264', quality=null}"   ],   "3-NEW-TRANSCODER-192.168.1.220": [],   "4-PROXY-192.168.1.219": [     "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",     "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}"   ] }</pre>	200 – OK  500 – Internal Server Error	Показать активные маршруты CDN

Параметры

Описание	Пример
Имя потока (с указанием профиля, при необходимости)	test-webrtc-144
Параметры запрашиваемого профиля транскодирования	
Параметры запрашиваемого профиля	REQUESTED-PROFILE
Параметры аудио, указанные в профиле	AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}
Параметры видео, указанные в профиле	VIDEO{width=256, height=144, gop=null, fps=null, bitrate=0, codec='h264', codecImpl='OPENH264', quality=null}
Параметры потока на Transcoder	
Параметры профилей транскодирования, используемых в данный момент на Transcoder	1-PROFILE-192.168.1.220

Параметры потока, который Transcoder забрал с Origin	2-STREAM-192.168.1.220
Параметры профиля аудио	AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}
Параметры профиля видео (в соответствии с работающими в данный момент профилями транскодирования)	VIDEO{width=320, height=180, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='FF', quality=null}
Параметры потока, который Transcoder может забрать с Origin	3-NEW-TRANSCODER-192.168.1.220
<b>Параметры потока на Origin</b>	
Параметры потока, который опубликован на Origin	4-PROXY-192.168.1.219
Параметры аудио	AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}
Параметры видео	VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}

## Примеры разбора маршрутов воспроизведения потока

Рассмотрим несколько примеров построения маршрутов для воспроизведения потока в CDN.

В качестве примера возьмем CDN из трех узлов:

- 192.168.1.219 - Origin
- 192.168.1.220 - Transcoder
- 192.168.1.221 - Edge

На Origin публикуется WebRTC H264+opus (48 кГц, стерео) поток test

## Воспроизведение потока с Origin без транскодинга при совпадении профилей публикации и воспроизведения

Воспроизводим поток test на Edge по профилю

```
-webrtc-opus-video-proxy:
audio:
  codec : opus
  rate  : 48000
  channels : 2
video:
  codec: h264
```

В ответ на запрос

```
http://192.168.1.221:8081/rest-api/cdn/stream/show_routes
{
  "streamName": "test-webrtc-opus-video-proxy"
}
```

получаем:

1. Запрашиваемый профиль:

```
"REQUESTED-PROFILE": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='h264', codecImpl='null', quality=null}"
],
```

2. Поток захватывается с Origin:

```
"1-PROXY-PROFILE-192.168.1.219": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}"
],
```

### 3. Поток может быть захвачен через Transcoder

```
"2-NEW-TRANSCODER-192.168.1.220": [],
```

### 4. Но захвачен с Origin, поскольку параметры публикации потока совпадают с запрошенными параметрами воспроизведения:

```
"3-PROXY-192.168.1.219": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}"
]
```

## Воспроизведение потока с Transcoder с транскодингом звука

Воспроизводим поток test на Edge по профилю

```
-webrtc-pcma-video-proxy:
audio:
  codec : pcma
  rate  : 8000
  channels : 1
```

### В ответ на запрос

```
http://192.168.1.221:8081/rest-api/cdn/stream/show_routes
{
  "streamName": "test-webrtc-pcma-video-proxy"
}
```

получаем:

#### 1. Запрашиваемый профиль:

```
"REQUESTED-PROFILE": [
  "AUDIO{bitrate=0, codec='pcma', rate=8000, channels=1}"
],
```

#### 2. На Transcoder создан профиль транскодирования:

```
"1-PROFILE-192.168.1.220": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='null', quality=null}",
  "AUDIO{bitrate=0, codec='PCMA', rate=8000, channels=1}"
],
```

#### 3. Поток захватывается через Transcoder

```
"2-STREAM-192.168.1.220": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='null', quality=null}",
  "AUDIO{bitrate=0, codec='PCMA', rate=8000, channels=1}"
],
"3-NEW-TRANSCODER-192.168.1.220": [],
```

#### 4. Параметры публикации потока на Origin:

```
"3-PROXY-192.168.1.219": [  
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",  
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}"  
]
```

## Воспроизведение потока с Transcoder с транскодингом видео

Воспроизводим поток test на Edge по профилю

```
-opus-vp8:  
audio:  
  codec : opus  
  rate : 48000  
  channels : 2  
video:  
  width : 320  
  height : 240  
  gop : 60  
  fps : 30  
  codec : vp8  
  codecImpl : FF
```

В ответ на запрос

```
http://192.168.1.221:8081/rest-api/cdn/stream/show_routes  
{  
  "streamName": "test-webrtc-opus-vp8"  
}
```

получаем:

#### 1. Запрашиваемый профиль:

```
"REQUESTED-PROFILE": [  
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",  
  "VIDEO{width=320, height=240, gop=60, fps=30, bitrate=0, codec='vp8', codecImpl='FF', quality=null}"  
],
```

#### 2. На Transcoder создан профиль транскодирования:

```
"1-PROFILE-192.168.1.220": [  
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",  
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}",  
  "VIDEO{width=320, height=240, gop=60, fps=30, bitrate=0, codec='VP8', codecImpl='FF', quality=null}"  
],
```

#### 3. Поток захватывается через Transcoder

```
"2-STREAM-192.168.1.220": [  
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",  
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}",  
  "VIDEO{width=320, height=240, gop=60, fps=30, bitrate=0, codec='VP8', codecImpl='FF', quality=null}"  
],  
"3-NEW-TRANSCODER-192.168.1.220": [],
```

#### 4. Параметры публикации потока на Origin:

```
"3-PROXY-192.168.1.219": [
  "AUDIO{bitrate=0, codec='opus', rate=48000, channels=2}",
  "VIDEO{width=0, height=0, gop=null, fps=null, bitrate=0, codec='H264', codecImpl='', quality=null}"
]
```

## Анонсирование потока в CDN по получению ключевого кадра

По умолчанию, потоки, опубликованные на Origin узлах, анонсируются в CDN и становятся доступными для построения маршрутов и проигрывания сразу после публикации. При этом разрешение и соотношение сторон картинки видеопотока неизвестны до тех пор, пока публикующий клиент не пришлет ключевой кадр.

В некоторых случаях, на Edge узлах необходимо знать разрешение и соотношение сторон потока, как только он стал доступен в CDN. Для этого, начиная со сборки [5.2.404](#), Origin может анонсировать поток только после получения ключевого кадра. Эта возможность включается настройкой

```
cdn_advertise_streams_by_kframe=true
```

Отметим, что аудио потоки (без видеодорожки) не будут доступны в CDN при такой настройке, поскольку не содержат ключевых кадров.

## Распределение нагрузки по Transcoder узлам

### Определение загрузки узла, исходя из загрузки процессора

Если в CDN входят несколько Transcoder узлов с одинаковыми профилями, нагрузка по таким узлам распределяется, исходя из загрузки процессора. Максимально допустимая загрузка конкретного Transcoder узла определяется настройкой

```
cdn_node_load_average_threshold=1.0
```

Эта величина задает соотношение между средней загрузкой процессора (параметр JVM SystemLoadAverage) и количеством процессорных ядер с учетом гипертрединга. Если значение превышено, Transcoder перестает участвовать в выборе маршрута для новых потоков и профилей. При этом в секции PROFILE данный узел отображается в маршрутах потока для тех профилей, по которым уже транскодируется этот поток.

Текущее состояние узла (участвует ли он в выборе маршрута для новых потоков) можно определить при помощи [REST API](#). Узел в статусе NEW\_STREAMS\_ALLOWED может принимать новые потоки для транскодирования, узел в статусе GROUP\_CONNECTION\_ALLOWED может только отдавать потоки, которые на нем уже транскодируются.

Настройка должна задаваться на Transcoder узлах. Также эта настройка может задаваться на Origin узлах, в этом случае при помощи REST API можно контролировать загрузку Origin сервера и при достижении статуса GROUP\_CONNECTION\_ALLOWED публиковать потоки на другом Origin сервере.

## Ограничение количества кодировщиков (encoders) видео на Transcoder узлах

При необходимости, на Transcoder узлах может быть ограничено максимальное количество одновременно используемых кодировщиков (encoders) видео при помощи настройки

```
cdn_transcoder_video_encoders_threshold=10000
```

При достижении этого количества, узел Transcoder переходит в статус GROUP\_CONNECTION\_ALLOWED. Учитываются все кодировщики на сервере, например, если один поток транскодируется по двум профилям, будет использовано 3 кодировщика, включая кодировщик PNG.

Настройка должна задаваться на Transcoder узлах

## Ограничение общей нагрузки на декодирование

В сборке [5.2.594](#) добавлена возможность установить ограничение по величине [общей нагрузки](#) на декодирование потоков. Например, для того, чтобы разрешить декодирование не более чем 20 потоков 1080p 30 fps, необходимо установить следующее значение

```
cdn_transcoder_video_decoders_load_threshold=1244160000
```

что соответствует произведению



```
1920 * 1080 * 30 * 20
```

При достижении этого количества, узел Transcoder переходит в статусGROUP\_CONNECTION\_ALLOWED.

Настройка должна задаваться на Transcoder узлах.

## Ограничение общей нагрузки на кодирование

В сборке5.2.594добавлена возможность установить ограничение по величине [общей нагрузки](#) на кодирование потоков. Например, для того, чтобы разрешить кодирование не более чем 20 потоков 720р, 480р и 240р 25 fps, необходимо установить следующее значение

```
cdn_transcoder_video_encoders_load_threshold=716400000
```

что соответствует сумме

```
1280 * 720 * 25 * 20 + 852 * 480 * 25 * 20 + 426 * 240 * 25 * 20
```

При достижении этого количества, узел Transcoder переходит в статусGROUP\_CONNECTION\_ALLOWED.

Настройка должна задаваться на Transcoder узлах.

## Ограничение процента деградации потоков

В сборке5.2.594добавлена возможность установить ограничение на процент деградации потоков. Например, для того, чтобы остановить прием новых потоков на транскодирование при деградации 10% потоков на сервере, необходимо установить следующее значение

```
cdn_transcoder_degraded_streams_threshold=10
```

При достижении этого количества, узел Transcoder переходит в статусGROUP\_CONNECTION\_ALLOWED. Необходимо отметить, что потоки начинают деградировать, как правило, из-за того, что кодировщику либо декодировщику не хватает производительности, и очереди картинок на кодирование/декодирование к моменту деградации вырастают до больших значений, что ведет к расходованию оперативной памяти. Таким образом, следует использовать этот параметр только как дополнительный к ограничениям общей нагрузки на кодирование/декодирование.

Настройка должна задаваться на Transcoder узлах.

## Равномерное распределение нагрузки по узлам

Для того, чтобы потоки по доступным Transcoder узлам распределялись равномерно, для транскодирования нового потока выбирается следующий свободный узел (алгоритм Round Robin). При этом, чтобы один и тот же опубликованный поток при одновременном запросе с нескольких клиентов не расходился по нескольким Transcoder узлам, предусмотрено кэширование по имени потока. Интервал времени кэширования задается в миллисекундах при помощи настройки

```
cdn_transcoder_for_new_connects_expire=10000
```

В данном случае интервал кэширования составит 10 секунд.

Настройка должна задаваться на Transcoder узлах

## Предотвращение транскодирования на Origin

Для того, чтобы Origin узлы не принимали на себя транскодинг, если все Transcoder узлы заняты, предусмотрена настройка

```
cdn_origin_allowed_to_transcode=false
```

В данном случае (по умолчанию), если для транскодирования нового потока по указанному профилю нет доступных Transcoder узлов, поток не будет играть с ошибкойNo available transcoders.

При необходимости, транскодирование на Origin серверах можно разрешить

```
cdn_origin_allowed_to_transcode=true
```

Эта настройка должна задаваться на Edge узлах.

## Управление текущим состоянием узла при помощи REST API

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

- HTTP:http://test.flashphoner.com:8081/rest-api/cdn/show\_nodes
- HTTPS:https://test.flashphoner.com:8444/rest-api/cdn/show\_nodes

Здесь:

- test.flashphoner.com - адрес WCS-сервера
- 8081 - стандартный REST / HTTP порт WCS-сервера
- 8444 - стандартный HTTPS порт
- rest-api - обязательная часть URL
- /cdn/show\_nodes- используемый REST-метод

### REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/cdn/show_nodes		<pre>[   {     "globalState":     "ACTIVE",     "id": "192.168.1.64",     "processingState":     "NEW_STREAMS_ALLOWED",     "role": "TRANSCODER"   },   {     "globalState":     "ACTIVE",     "id": "192.168.1.39",     "processingState":     "NEW_STREAMS_ALLOWED",     "role": "ORIGIN"   } ]</pre>	200 – OK  500 – Internal Server Error	Показать состояние узлов CDN
/cdn/show_state		<pre>NEW_STREAMS_ALLOWED</pre>	200 – OK  500 – Internal Server Error	Показать состояние узла CDN, которому направлен запрос
/cdn/enforce_state	<pre>{   "state":   "GROUP_CONNECTIONS_ALLOWED" }</pre>		200 – OK  500 – Internal Server Error	Принудительно изменить состояние узла

### Параметры

Имя параметра	Описание	Пример
globalState	Статус узла: ACTIVE или PASSIVE	ACTIVE
id	Адрес узла	192.168.1.64

processingState, state	Участвует ли узел в выборе маршрута для воспроизведения потоков:  NEW_STREAMS_ALLOWED - участвует  CONNECTION_ALLOWED - позволяет только забирать уже транскодируемые потоки, в том числе по новым профилям  GROUP_CONNECTION_ALLOWED - не участвует	NEW_STREAMS_ALLOWED
role	Роль узла: ORIGIN, TRANSCODER или EDGE	ORIGIN

Запрос /cdn/show\_nodes может быть направлен определенному узлу, в ответ узел вернет состояния всех видимых ему узлов CDN, кроме себя самого.

Запрос/cdn/enforce\_state позволяет принудительно изменить состояние узла, например, вывести определенный транскoder из выбора маршрутов. Узел также можно полностью вывести из CDN запросом

```
POST /rest-api/cdn/enforce_state HTTP/1.1
Content-Length: 20
Content-Type: application/json

{
  "state": "PASSIVE"
}
```

а затем ввести обратно

```
POST /rest-api/cdn/enforce_state HTTP/1.1
Content-Length: 19
Content-Type: application/json

{
  "state": "ACTIVE"
}
```

Для сброса заданного состояния необходимо передать пустое тело запроса

```
POST /rest-api/cdn/enforce_state HTTP/1.1
Content-Length: 4
Content-Type: application/json

{}
```

## Вычисление текущего состояния узла и его рассылка

Текущее состояние узла определяется следующим образом в порядке приоритета:

1. Если есть значение, заданное принудительно при помощи запроса /cdn/enforce\_state, то выбирается это значение.
2. Если достигнуто ограничение по нагрузке процессора, то выбирается состояние GROUP\_CONNECTIONS\_ALLOWED.
3. Если достигнуто ограничение по кодировщикам видео, то выбирается состояние GROUP\_CONNECTIONS\_ALLOWED.
4. Если ограничения не достигнуты, то выбирается состояние NEW\_STREAMS\_ALLOWED.

Узел рассылает свое состояние другим узлам с периодичностью, заданной в миллисекундах при помощи настройки

```
cdn_nodes_state_refresh_interval=60000
```

По умолчанию, периодичность рассылки составляет 60 секунд. Для Transcoder узлов рекомендуется уменьшить это значение до 1 секунды, чтобы своевременно распределять потоки на другие узлы.

## Настройка состояния Transcoder узла при достижении ограничений по производительности

В сборке 5.2.640 добавлена настройка, позволяющая указать состояние, в которое перейдет Transcoder узел при достижении [любого из ограничений](#), по умолчанию GROUP\_CONNECTIONS\_ALLOWED

```
cdn_transcoder_threshold_state=GROUP_CONNECTIONS_ALLOWED
```

При необходимости, чтобы предотвратить размазывание потока по CDN, можно переводить Transcoder в состояние `CONNECTIONS_ALLOWED`

```
cdn_transcoder_threshold_state=CONNECTIONS_ALLOWED
```

В этом состоянии, Transcoder будет позволять забирать потоки, которые уже на нем декодируются, в том числе по новым профилям. Например, если на Transcoder узле транскодируется поток test по профилю 360p, и какой-либо Edge сервер запросит поток test240p, маршрут для воспроизведения будет построен через этот же Transcoder.

Данная настройка должна устанавливаться на Transcoder узлах.

## Авторизация узлов в CDN

Авторизация узлов, которые пытаются подключиться к CDN, производится по IP-адресу. Адреса узлов, которые могут войти в CDN, должны быть указаны в следующей настройке

```
cdn_allowed_ips=192.168.1.39, 192.168.100.64, 192.168.101.65
```

Также в этой настройке могут быть указаны маски адресов, например

```
cdn_allowed_ips=192.168.1.39, 192.168.100.0/24
```

Каждый узел CDN, на котором указана данная настройка, будет принимать соединения только от тех узлов, чьи адреса совпадают с перечисленными либо подходят под указанные маски. Соединения с прочими узлами будут отвергнуты.

## Обратная совместимость с CDN 2.0

Обратная совместимость CDN 2.1 с CDN 2.0 поддерживается для следующих случаев:

1. Edge 2.0 может забирать потоки с Origin 2.1
2. Edge 2.1 может забирать потоки с Origin 2.0

При этом транскодинг работает согласно настройке кодеков и SDP [по принципам, определенным для CDN 2.0](#).

## Ограничения

1. Не рекомендуется публиковать поток с одним и тем же именем на двух Origin серверах одной CDN
2. Поток, опубликованный на одном из Origin серверов, следует забирать с этого же Origin или Edge сервера (в том числе через Transcoder сервер), но не с другого Origin сервера в той же CDN.

## Принудительное указание версии CDN, поддерживаемой узлом

В сборке [5.2.341](#) добавлена настройка, позволяющая принудительно указать версию CDN, которую поддерживает данный узел

```
cdn_force_version=2.0
```

По умолчанию, установлена версия 2.0. Далее узел динамически, в зависимости от роли (Transcoder) и настройки [ACL](#), устанавливает поддерживаемую версию в 2.1 или 2.2.

Для того, чтобы узел Edge быстрее определял роль узла Transcoder, рекомендуется на узле Transcoder принудительно указать версию 2.1

```
cdn_force_version=2.1
```

или 2.2, если в CDN используются правила ACL

```
cdn_force_version=2.2
```



В сборке [5.2.471](#) поведение данной настройки [изменилось](#).

## Известные проблемы

1. При использовании кодировщика FF в первых кадрах транскодированного потока могут быть искажения (макроблоки)

Симптомы: при использовании профилей транскодинга в первых кадрах потока зритель видит макроблоки

Решение:

а) указать в профиле параметр quality

```
profiles:
-240p:
  audio:
    codec : opus
    rate : 48000
  video:
    height : 240
    gop : 50
    fps : 25
    codec : h264
    preset : veryfast
    quality : 25
    codecImpl : FF
```

б) если необходимо удерживать при транскодинге заданное значение битрейта видео, использовать кодировщик OPENH264

```
profiles:
-240p:
  audio:
    codec : opus
    rate : 48000
  video:
    height : 240
    bitrate : 400
    gop : 50
    fps : 25
    codec : h264
    codecImpl : OPENH264
```

2. Настройка качества кодирования не применяется при использовании OpenH264

Симптомы: качество картинки не изменяется при различных значениях quality в профиле транскодирования

Решение: не использовать кодирование на базе OpenH264, поскольку управление CRF в нем не поддерживается

```
profiles:
-240p:
  audio:
    ...
  video:
    ...
    quality: 10
    codecImpl : FF
```