

CDN 2.3

Описание

CDN 2.3 дает следующие дополнительные возможности по сравнению с [CDN 2.1](#) и [2.2](#):

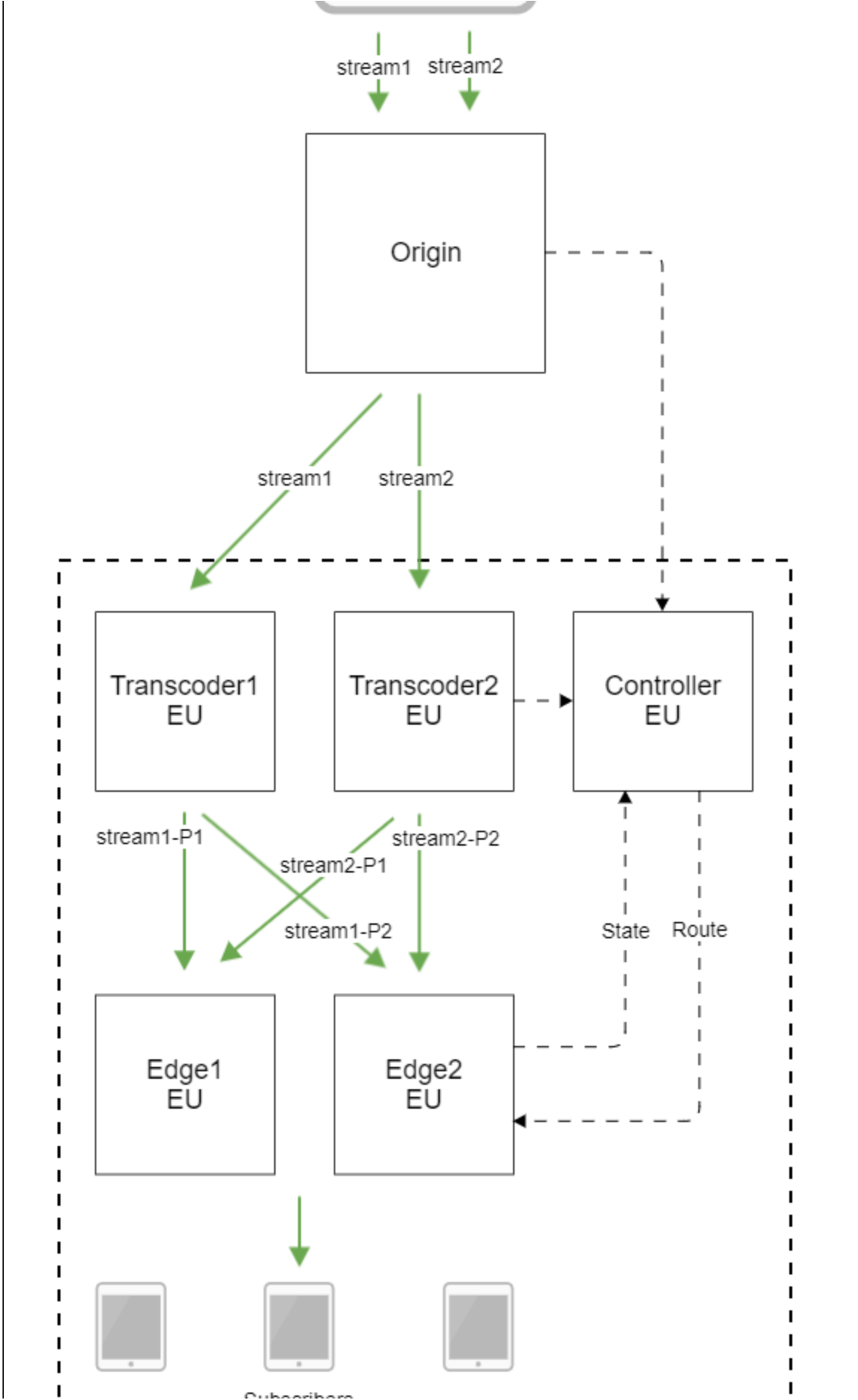
- предотвращение двойного транскодирования одного и того же потока в CDN
- отображение расширенной информации о CDN
- контроль пропускной способности каналов между серверами CDN
- периодическая отсылка состояния другим узлам

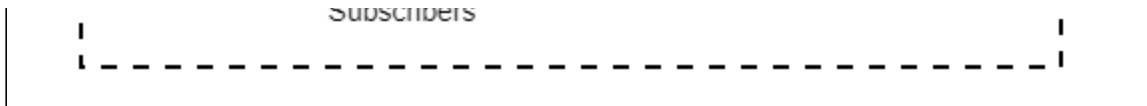
Предотвращение двойного транскодирования одного и того же потока в CDN

CDN 2.3 позволяет ограничить двойное транскодирование одного и того же входящего в CDN потока на двух и более транскодерах одной группы. Такое может происходить в крупных CDN с большими объемами входящего и исходящего медиа трафика при одновременном подключении большого числа подписчиков. В результате создается излишняя нагрузка на транскодеры, которая может приводить к выходу серверов из работы.

Для решения этой проблемы, в каждой группе выделяется специальный узел с ролью Controller, который получает информацию о потоках от всех узлов CDN и собирает ее текущее состояние. Узлы с ролью Edge должны обращаться к узлу Controller, при его наличии, для определения маршрута воспроизведения потока. Если поток уже транскодируется на каком-либо из узлов Transcoder, узел Controller не позволяет строить маршрут для этого потока через другой транскодер. Если узел Controller недоступен, Edge строит маршрут для воспроизведения потока по алгоритму [CDN 2.1](#).







Узел Controller не должен использоваться для публикации или воспроизведения потока.

Настройка

Роль Controller назначается узлу при помощи настройки

```
cdn_role=controller
```

На узле Edge определяется таймаут запроса к узлу Controller при помощи параметра

```
cdn_controller_request_timeout=5000
```

Таймаут задается в миллисекундах, по умолчанию значение установлено в 5000 мс.

Длительность кэширования ответов узла Controller на запросы к нему задается на узле Edge при помощи параметра

```
cdn_controller_response_cache_expire=10000
```

Кэширование задается в миллисекундах, по умолчанию значение установлено в 10000 мс.

Настройки таймаута и кэширования должны задаваться только на Edge узлах.

Проверка двойного транскодирования при помощи REST API

Проверка двойного транскодирования проводится при помощи REST запроса к узлу Edge.

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

- HTTP: `http://test.flashphoner.com:8081/rest-api/cdn/show_redundant_transcodings`
- HTTPS: `https://test.flashphoner.com:8444/rest-api/cdn/show_redundant_transcodings`

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_redundant_transcodings` - используемый REST-метод

REST запросы необходимо отправлять на Edge сервер.

REST-методы и статусы ответа

REST метод	Тело ответа	Статусы ответа	Описание
<code>/cdn/show_redundant_transcodings</code>	<pre>{ "test6": "192.168.1.17, 192.168.1.18" }</pre>	200 Redundant transcoding found 404 No redundant transcoding	Проверить двойное транскодирование

Параметры ответа

Описание	Пример		
Имя потока	<code>test6</code>	Список узлов, на которых транскодируется поток	<code>192.168.1.17, 192.168.1.18</code>

Отображение расширенной информации о CDN

В сборке [5.2.471](#) в статистику, в CLI и в REST API добавлен либо дополнен вывод информации о CDN, в которой участвует данный сервер

Использование CLI

Команда интерфейса командной строки WCS

```
show cdn-nodes
```

отображает версию CDN, поддерживаемую каждым узлом, и группу, в которую входит узел:

```

Ip           State Processing state  Role      Version Group
-----
192.168.0.102 ACTIVE NEW_STREAMS_ALLOWED TRANSCODER 2.3      null
192.168.0.187 ACTIVE NEW_STREAMS_ALLOWED EDGE       2.3      null

```

При этом для узла в статусе **ACTIVE** отображается версия CDN, которая получена от узла в результате согласования версий, для узла в статусе **PASSIVE** - версия, заданная в настройках текущего узла (см ниже)

Использование REST API

REST API запрос `/cdn/show_nodes` возвращает версию CDN, поддерживаемую каждым узлом

```
[
  {
    "version": "2.3",
    "role": "TRANSCODER",
    "globalState": "ACTIVE",
    "processingState": "NEW_STREAMS_ALLOWED",
    "id": "192.168.0.102"
  },
  {
    "version": "2.3",
    "role": "EDGE",
    "globalState": "ACTIVE",
    "processingState": "NEW_STREAMS_ALLOWED",
    "id": "192.168.0.187"
  }
]
```

При этом для узла в статусе **ACTIVE** отображается версия CDN, которая получена от узла в результате согласования версий, для узла в статусе **PASSIVE** - версия, заданная в настройках текущего узла (см ниже)

Контроль качества каналов между серверами CDN

В сборке **5.2.483** добавлена возможность **контроля качества каналов** в направлении Origin -> Transcoder -> Edge. Для этого необходимо установить интервал отсылки статистики исходящего битрейта

```
outbound_video_rate_stat_send_interval=1
```

на Origin и Transcoder узлах.

В этом случае на стороне узла, принимающего поток, периодически сравнивается битрейт публикации или воспроизведения с битрейтом на стороне узла, отправляющего поток. При устойчивом их расхождении диагностируется снижение пропускной способности канала. Пики и резкие изменения сглаживаются **фильтром Калмана**. Показатель качества канала и статистику для потока можно получить при помощи REST API.

Получение статистики качества при помощи REST API

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

- HTTP: `http://test.flashphoner.com:8081/rest-api/cdn/stats/print`
- HTTPS: `https://test.flashphoner.com:8444/rest-api/cdn/stats/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/stats/print` - используемый REST-метод

REST запросы необходимо отправлять на сервер, на котором необходимо получить статистику входящего потока. Например, на Edge можно получить статистику для входящего потока, полученного с Origin или Transcoder, а на Transcoder для потока, полученного с Origin.

REST-методы и статусы ответа

REST метод	Тело запроса	Тело ответа	Статусы ответа	Описание
------------	--------------	-------------	----------------	----------

REST метод	Тело запроса	Тело ответа	Статусы ответа	Описание
<code>`/cdn/stats/print`</code>	<pre>{ "name": "test", "host": "192.168.0.111", "format": "json" }</pre>	<pre>{ "192.168.0.111": [{ "fps": 30, "inboundBitrate": 1658292, "nack": 0, "name": "test", "outboundBitrate": 1692676, "quality": "PERFECT" }] }</pre>	200 OK 404 Stream not found	Получить статистику потока

Параметры

Параметр	Описание	Пример
name	Имя потока	<code>`test`</code>
host	Узел, с которого получен поток	<code>`192.168.0.111`</code>
format	Формат ответа	<code>`json`</code>

Параметр	Описание	Пример
inboundBitrate	Входящий (полученный) битрейт	`1658292`
outboundBitrate	Исходящий (передаваемый) битрейт	`1692676`
nack	Количество NACK	`0`
fps	Частота кадров потока	`30`
quality	Качество канала	`PERFECT`

Запрос без указания имени потока вернет информацию по всем потокам, получаемым с указанного узла CDN. Запрос без указания узла вернет информацию по всем потокам, получаемым с других узлов CDN, если не указано имя потока.

Форматы получения статистики

По умолчанию, статистика возвращается в формате JSON

```
POST /rest-api/cdn/stats/print HTTP/1.1
Content-Length: 19
Content-Type: application/json
Host: test.flashphoner.com:8081
{
  "name": "test"
}

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 125
Content-Type: application/json
{
  "192.168.0.111": [
    {
      "fps": 31,
      "inboundBitrate": 1052794,
      "nack": 81,
      "name": "test",
      "outboundBitrate": 1048518,
      "quality": "PERFECT"
    }
  ]
}
```

Также поддерживается формат Prometheus

```
POST /rest-api/cdn/stats/print HTTP/1.1
Content-Length: 43
Content-Type: application/json
Host: test.flashphoner.com:8081
```



```

{
  "format": "prometheus",
  "name": "test"
}

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 397
Content-Type: plain/text

cdn_connection_quality{param="inboundBitrate",node="192.168.0.111",name="test"} 1249049
cdn_connection_quality{param="outboundBitrate",node="192.168.0.111",name="test"} 1240717
cdn_connection_quality{param="quality",node="192.168.0.111",name="test"} 3
cdn_connection_quality{param="fps",node="192.168.0.111",name="test"} 30
cdn_connection_quality{param="nack",node="192.168.0.111",name="test"} 81

```

Показатель качества принимает следующие значения

JSON	Prometheus	Описание
UNKNOWN	0	Качество неизвестно, сервер не получает медиа пакеты
BAD	1	Качество плохое
GOOD	2	Качество хорошее
PERFECT	3	Качество отличное

Периодическая отсылка состояния другим узлам

Чтобы снизить влияние качества каналов между узлами на сбор информации о текущем состоянии CDN на каждом из узлов, добавлен ряд настроек, которые регулируют периодичность отсылки различных признаков состояния узла CDN другим подключенным к нему узлам

Настройка	Значение по умолчанию	Описание
`cdn_nodes_acl_refresh_interval`	60000	Периодичность рассылки списка ACL ключей, мс
`cdn_nodes_group_refresh_interval`	60000	Периодичность рассылки списка групп, к которым принадлежит узел, мс

Настройка	Значение по умолчанию	Описание
<code>`cdn_nodes_role_refresh_interval`</code>	60000	Периодичность рассылки роли узла, мс
<code>`cdn_nodes_route_refresh_interval`</code>	60000	Периодичность рассылки маршрутов, мс
<code>`cdn_nodes_state_refresh_interval`</code>	60000	Периодичность рассылки состояния узла, мс
<code>`cdn_nodes_version_refresh_interval`</code>	90000	Периодичность рассылки версии CDN, поддерживаемой узлом, мс

Обратная совместимость с CDN 2.2, 2.1, 2.0

Узел Controller с точки зрения предыдущих версий

С точки зрения узлов CDN предыдущих версий, узел Controller представляет собой Origin сервер

Узлы CDN 2.1 будут выбирать маршруты воспроизведения потоков без участия узла Controller, даже если он присутствует в группе.

Согласование версий CDN

Начиная со сборки [5.2.471](#), согласование версий CDN работает следующим образом: при установке соединения для CDN сигналинга, узлы отправляют друг другу поддерживаемую версию CDN. Если версии различаются, узел с большей версией понижает ее до меньшей. Например, если в CDN 2.2 вводится Edge с версией 2.3, он будет отправлять другим узлам только сообщения, совместимые с версией 2.2. С точки зрения всей CDN данный Edge сервер будет выглядеть, как поддерживающий версию 2.2.

Чтобы избежать потери согласования версий при разрывах связи между узлами, каждый узел периодически рассылает другим узлам свою версию. Интервал рассылки указывается в миллисекундах при помощи настройки

```
cdn_nodes_version_refresh_interval=90000
```

По умолчанию, период рассылки версии составляет 90 секунд.

Назначение версии CDN узлу в статусе **PASSIVE**

Если узел перешел в статус **PASSIVE** в результате потери связи с ним, либо явного назначения статуса REST запросом `/cdn/enforce_state`, обмен сообщениями с таким узлом прекращается. Остальные узлы будут предполагать, что узел в статусе **PASSIVE** поддерживает версию CDN, указанную настройкой

```
cdn_force_version=2.0
```

По умолчанию, узлам в статусе **PASSIVE** назначается версия CDN 2.0.

Например, если в CDN 2.3 один из Edge узлов теряет связь с остальными, узлы CDN будут предполагать, что данный Edge поддерживает версию CDN 2.0 до тех пор, пока этот Edge вновь не подключится к CDN и не пройдет процедуру согласования версий.

Ограничения

1. Для того, чтобы узел Edge выбирал маршрут воспроизведения потоков при помощи узла Controller, эти узлы должны принадлежать к одной группе, и эта группа должна быть определена явно в настройках CDN:

```
cdn_groups=g1
```

2. Узел Controller строит маршруты только через транскодеры своей группы.
3. Узел Controller, для которого группа не определена, не участвует в определении маршрутов.