CDN 2.0

- Описание
  - Принципы работы динамической CDN
  - Роли серверов в CDN
  - Схема работы
  - Принципы идентификации потоков в CDN
- Настройка
  - Параметры файла настроек
  - Примеры конфигурации
    - Пример минимальной конфигурации
    - ПримерконфигурациисдвумяOrigin-серверами
  - Использование CLI
  - Использование REST API
    - REST-методы и статусы ответа
    - Параметры
  - Удаление остановленных серверов из списка узлов
- Принципы выбора аудио кодеков при передаче потока через CDN
  - Проброс звука через CDN
  - Настройка SDP для публикации и воспроизведения RTMP потока в CDN
    - flash_handler_publish.sdp
    - flash_handler_play.sdp
  - Выбор кодека при воспроизведении потока
- Настройка таймаутов на чтение и запись в CDN для доставки RTMP потоков
- Краткое руководство по тестированию
  - Подготовка к тестированию
  - Настройка WCS
  - Запуск трансляции с веб-камеры на Origin-сервер
  - Воспроизведение потока на Edge-сервере
  - Остановка трансляции с веб-камеры на Origin-сервер
- Ограничения
- Известные проблемы

CDN на базе WCS-серверов может быть организована двумя способами:

1. Статическая CDN, набор узлов в которой жестко определяется на этапе настройки, и для изменения конфигурации CDN необходимо перезапустить сервер(а), которые являются источниками потоков в данной сети. Такая CDN 1.0 организуется на базе функции балансировки нагрузки.
2. Динамическая CDN, набор узлов в которой может изменяться на ходу. Для включения/исключения узла из такой CDN достаточно перезапустить только этот узел.

В данном разделе рассмотрим динамическую CDN на базе Web Call Server 5.1.

Описание

Принципы работы динамической CDN

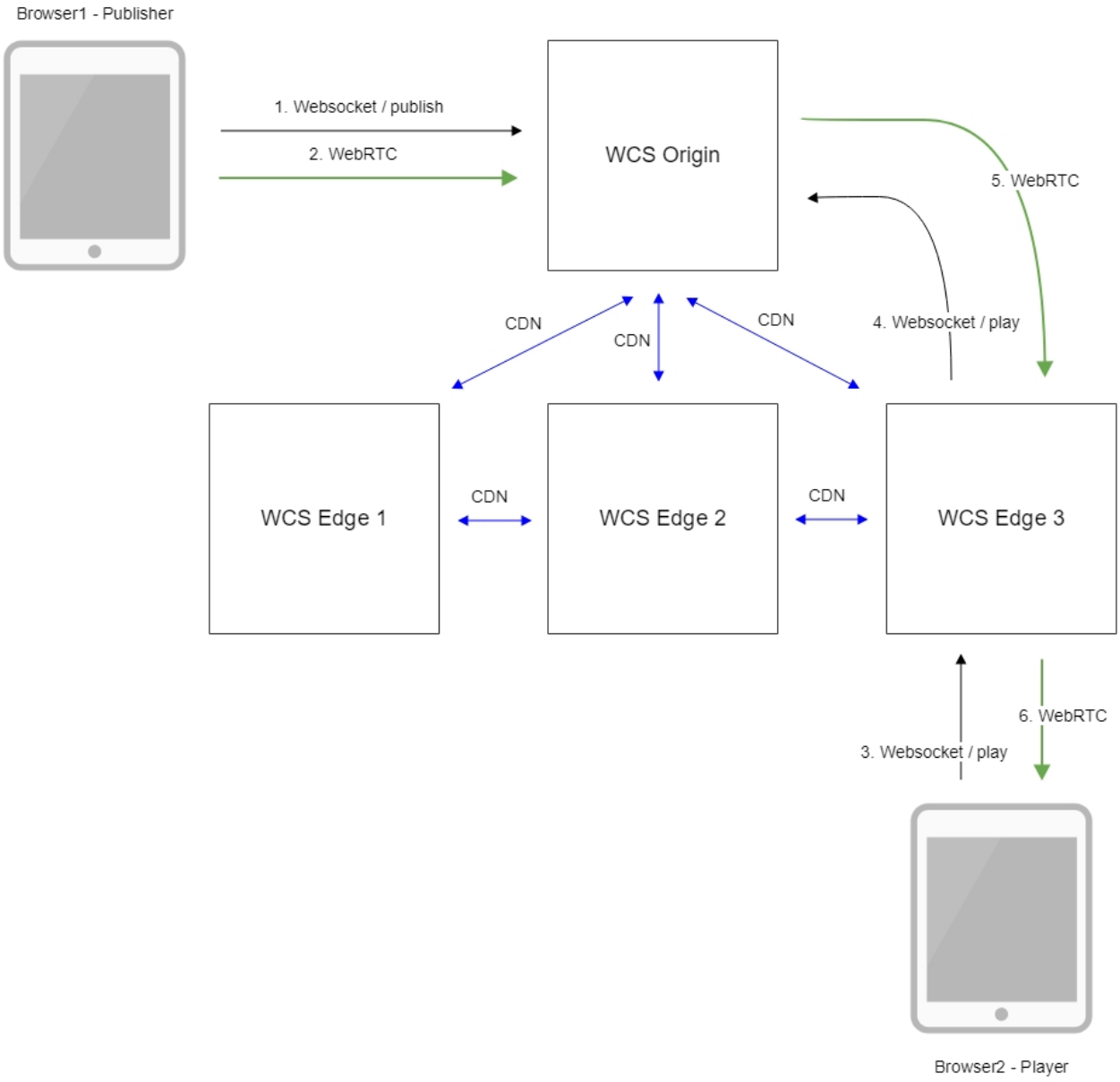
Распределенная динамическая сеть доставки контента (CDN) на базе WCS 5.1 работает следующим образом:

1. При запуске сервера запускается специальный CDN-модуль. Модуль отправляет запрос к серверу, указанному как точка входа в CDN, для получения от него списка других серверов в CDN и списка доступных потоков. Если точка входа не указана, либо указанный сервер недоступен, модуль ожидает извещения от какого-либо другого сервера в CDN (например, если этот сервер уже был добавлен в список CDN-серверов на другом сервере, или указан как точка входа). В результате каждый активный CDN-сервер в каждый момент времени имеет актуальный список всех остальных CDN-серверов.
2. Все взаимодействие между узлами CDN производится по Websocket.

Роли серверов в CDN

Сервер может иметь одну из двух ролей
1. Origin: служит источником потоков для других серверов в CDN. Может раздавать как опубликованные на нем потоки, так и потоки, которые он забирает с других серверов.
2. Edge: может забирать потоки с Origin-серверов, при этом не раздает свои (локальные) потоки другим серверам.
При этом потоки не раздаются принудительно, Edge-сервер забирает определенный поток с Origin только по запросу пользователя из браузера или мобильного приложения.

Схема работы



1. Браузер соединяется с Origin сервером по протоколу Websocket и отправляет команду publish.
2. Браузер захватывает микрофон и камеру и отправляет WebRTC поток на сервер.
3. Второй браузер устанавливает соединение с Edge3 сервером по Websocket и отправляет команду play.
4. Edge3 сервер запрашивает поток у Origin сервера
5. Edge3 сервер получает WebRTC поток.
6. Второй браузер получает WebRTC поток и воспроизводит этот поток на странице.

Принципы идентификации потоков в CDN

Имя потока уникально для каждого WCS-сервера (и однозначно идентифицирует поток), но не для CDN в целом. Таким образом, существует два ограничения:

1. Если на сервере опубликован поток с таким же именем, как у потока, уже опубликованного на каком-либо Origin-сервере, то при воспроизведении на этом сервере будет воспроизводиться локальный (не CDN) поток.
2. При воспроизведении CDN-потока на этом сервере не может быть опубликован поток с таким же именем. То же имя может быть использовано для публикации только после остановки воспроизведения и удаления pull-агента по истечении таймаута проверки активности (по умолчанию 1 минута).

Настройка

Параметры файла настроек

Для настройки CDN используются следующие основные параметры файла настроекflashphoner.properties(полный список см в описании файла настроек):

Параметр	Значение по умолчанию	Тип	Описание
cdn_enabled	false	Boolean	Включить/отключить CDN
cdn_ip	Empty	String	Адрес сервера в CDN
cdn_nodes_resolve_ip	false	Boolean	Разрешать имена серверов в IP-адреса
cdn_point_of_entry	Empty	String	Адрес сервера-точки входа в CDN для данного сервера. Если сервер выполняет роль Origin, эту настройку не следует указывать равной IP адресу или имени этого сервера
cdn_port	8084	Int	Номер порта для CDN
cdn_role	edge	String	Роль сервера в CDN: origin - источник потоков для других серверов в CDN edge - может получать потоки с других серверов

Примеры конфигурации

Пример минимальной конфигурации

Два сервера: Origin (origin.flasphoner.com) и Edge (edge.flasphoner.com)

Настройки Origin:

```
cdn_enabled=true
cdn_ip=origin.flashponer.com
cdn_nodes_resolve_ip=true
cdn_role=origin
```

Настройки Edge:

```
cdn_enabled=true
cdn_ip=edge.flashphoner.com
cdn_nodes_resolve_ip=true
cdn_point_of_entry=origin.flashponer.com
cdn_role=edge
```

ПримерконфигурациисдвумяOrigin-серверами

Четыре сервера: Origin1 (origin1.flasphoner.com), Origin2 (origin2.flasphoner.com) и Edge1 (edge1.flasphoner.com), Edge2 (edge2.flasphoner.com)

Настройки Origin1:

```
cdn_enabled=true
cdn_ip=origin1.flashponer.com
cdn_nodes_resolve_ip=true
cdn_role=origin
```

Настройки Origin2:

```
cdn_enabled=true
cdn_ip=origin2.flashponer.com
cdn_point_of_entry=origin1.flashponer.com
cdn_nodes_resolve_ip=true
cdn_role=origin
```

Настройки Edge1:

```
cdn_enabled=true
cdn_ip=edge1.flashphoner.com
cdn_point_of_entry=origin1.flashponer.com
cdn_nodes_resolve_ip=true
cdn_role=edge
```

Настройки Edge2:

```
cdn_enabled=true
cdn_ip=edge2.flashphoner.com
cdn_point_of_entry=origin1.flashponer.com
cdn_nodes_resolve_ip=true
cdn_role=edge
```

Использование CLI

Получение информации о текущем состоянии CDN изкомандной строкиWCS производится следующими командами:

Команда	Описание	Пример результата

show cdn-nodes	Отображает список узловых серверов в CDN:
	
	ACTIVE - сервер запущен, отвечает на запросы и/или рассылает извещения
	
	PASSIVE - сервер остановлен или недоступен
		`Ip State`
		
		`---`
		`edge1.flashphoner.com ACTIVE`
		`edge2.flashphoner.com ACTIVE`
		`origin2.flashphoner.com PASSIVE`

show cdn-routes	Отображает список активных потоков в CDN	`Route Stream`
		
		`---`
		`1-origin2.flashphoner.com-2 stream1`
		`0-origin2.flashphoner.com-0 stream2`
		`2-origin1.flashphoner.com-1 stream1`

Использование REST API

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

- HTTP:http://test.flashphoner.com:8081/rest-api/cdn/show_routes
- HTTPS:https://test.flashphoner.com:8444/rest-api/cdn/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_routes - используемый REST-метод

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/cdn /show_routes		<pre>{ "1-origin2.flashphoner.com-2": "stream1", "0-origin2.flashphoner.com-0": "stream2", "2-origin1.flashphoner.com-1": "stream1" }</pre>	200 – OK 500 – Internal Server Error	Показать активные маршруты CDN

Параметры

Описание	Пример
Идентификатор маршрута	`1-origin2.flashphoner.com-2`
Имя потока в CDN	`stream1`

Удаление остановленных серверов из списка узлов

По умолчанию, остановленный или недоступный сервер будет отображаться в списке узлов CDN в статусе PASSIVE до тех пор, пока он вновь не будет запущен, либо вся CDN не будет остановлена. При необходимости, может быть установлен интервал, по истечении которого неактивный узел будет удаляться из списка. Интервал задается в секундах при помощи параметра в файле настроекflashphoner.properties, например

```
cdn_nodes_timeout=60
```

В данном случае неактивные узлы будут удаляться из списка через 60 секунд.

Принципы выбора аудио кодеков при передаче потока через CDN

Проброс звука через CDN

При пробросе в CDN набор предлагаемых кодеков формируется с учетом

- кодека, используемого для публикации (приоритетный), и
- кодеков, подерживаемых на Origin и Edge (параметр `codecs` в файле flashphoner.properties)

с исключением кодеков, указанных в параметре `codecs_exclude_cdn` в файлеflashphoner.propertiesна Edge.

По умолчанию (если AAC не исключен настройками) в SDP указаны следующие частоты дискретизации AAC: 48, 44.1, 32, 24, 22.05, 16, 12, 8 kHz.

Например, если Edge запрашивает с Origin RTMP-поток, опубликованный с AAC 48 kHz, аудио SDP будет следующим

```
m=audio 31006 RTP/SAVPF 102 111 8 18 100 9 103 104 105 106 107 108 109 110
c=IN IP4 192.168.1.5
a=mid:1
a=rtpmap:102 mpeg4-generic/48000/2
a=rtpmap:111 opus/48000/2
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:100 speex/16000
a=rtpmap:9 G722/8000
a=rtpmap:103 mpeg4-generic/44100/2
a=rtpmap:104 mpeg4-generic/32000/2
a=rtpmap:105 mpeg4-generic/24000/2
a=rtpmap:106 mpeg4-generic/22050/2
a=rtpmap:107 mpeg4-generic/16000/2
a=rtpmap:108 mpeg4-generic/12000/2
a=rtpmap:109 mpeg4-generic/11025/2
a=rtpmap:110 mpeg4-generic/8000/2
```

PCMU по умолчанию не предлагается и будет включен, только если исключить PCMA:

```
codecs_exclude_cdn=alaw
```

Настройки SDP для RTMP (flash_handler_publish.sdp и flash_handler_play.sdp) на формирование SDP для проброса в CDN не влияет.

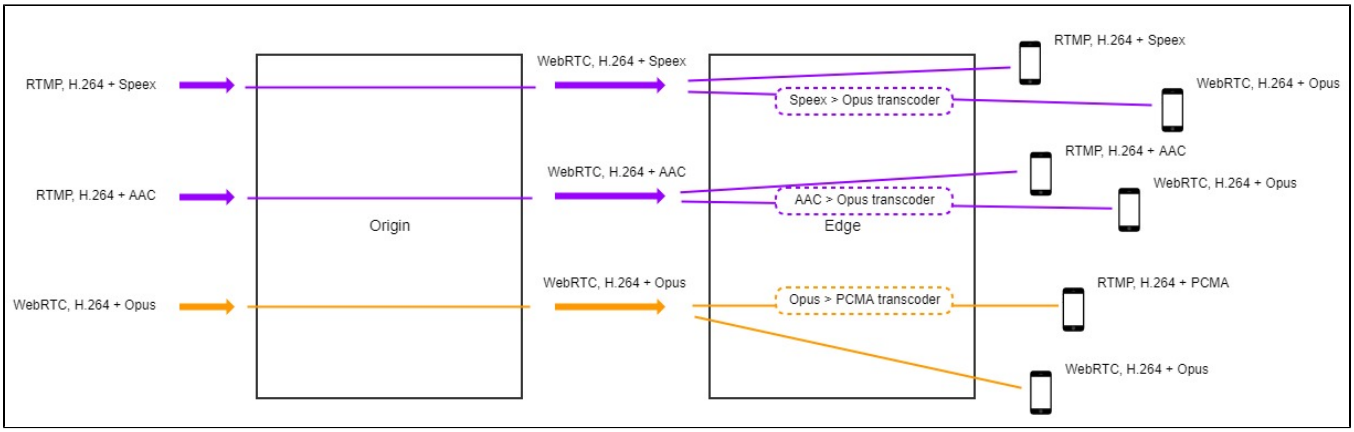
С настройками по умолчанию транскодинга не будет, например, в следующих случаях:

- публикация на Origin WebRTC (Opus), воспроизведение как WebRTC на Edge (Opus)
- публикация на Origin RTMP (AAC, любая из перечисленных выше частот дискретизации), воспроизведение как RTMP на Edge (AAC та же частота дискретизации, без ресемплинга)
- публикация на Origin RTMP (Speex), воспроизведение как RTMP на Edge (Speex)

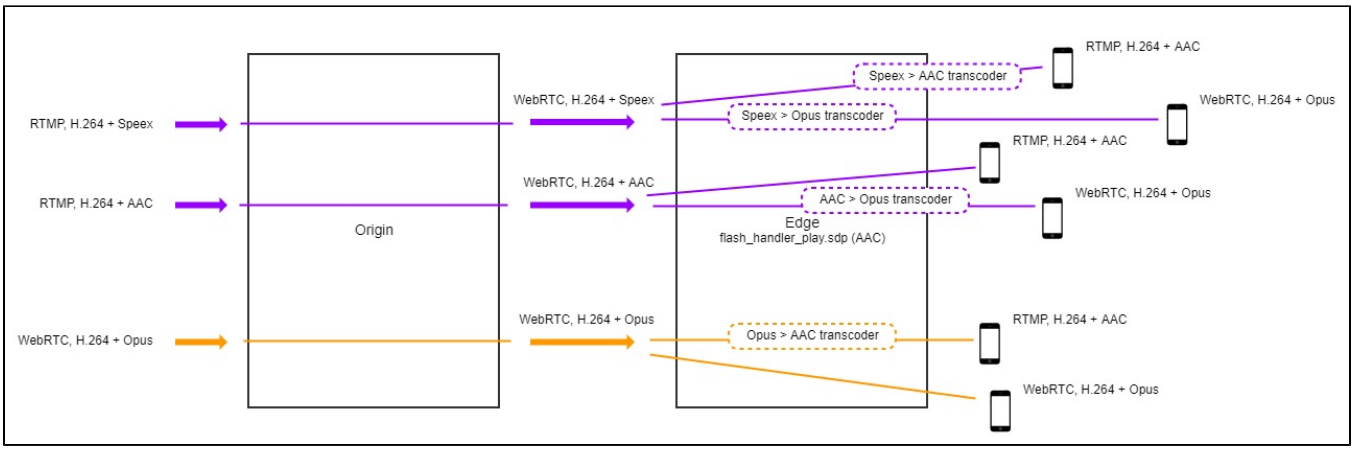
Если в наборе кодеков подписчика нет того, что используется для публикации, то транскодинг осуществляется на Edge-сервере:

- публикация на Origin WebRTC (Opus), воспроизведение как RTMP на Edge (AAC): транскодинг Opus-AAC на Edge
- публикация на Origin RTMP (AAC), воспроизведение как WebRTC на Edge (Opus): транскодинг AAC-Opus на Edge
- публикация на Origin RTMP (Speex), воспроизведение как WebRTC на Edge (Opus): транскодинг Speex-Opus на Edge
- публикация на Origin RTMP (Speex), воспроизведение как RTMP на Edge (AAC): транскодинг Speex-AAC на Edge

Если набор кодеков подписчика включает не только AAC, то при воспроизведении RTMP как RTMP без транскодинга может использоваться, например, и Speex.



Если же ограничить кодеки подписчика только AAC (flash_handler_play.sdp на Edge включает только AAC), то в случае использования другого кодека при публикации RTMP транскодинг будет осуществляться на Edge:

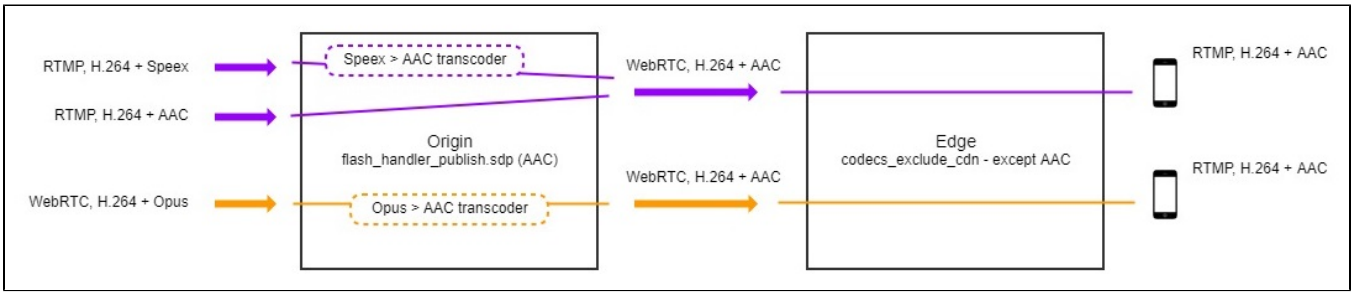


Для исключения кодеков при пробросе в CDN, чтобы избежать транскодинга на Edge-сервере, следует использовать настройку `codecs_exclude_cdn` на Edge.Тогда для проброса в CDN будет использоваться единственный оставшийся кодек, и в случае публикации с каким-либо другим кодеком транскодинг будет осуществляться на Origin-сервере.

Например, если предполагается, что к Edge будут подключаться

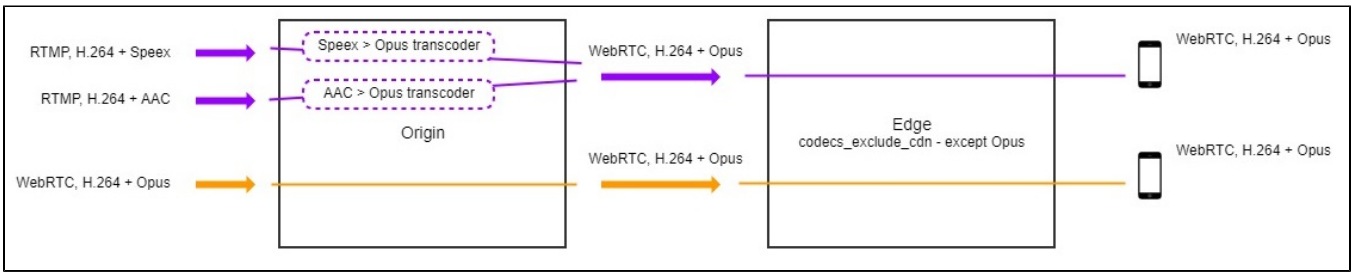
1. Только RTMP-подписчики с AAC – исключить все, кроме mpeg4-generic:

```
codecs_exclude_cdn=opus,alaw,ulaw,g729,speex16,g722,telephone-event,flv
```



2. Только WebRTC-подписчики с Opus – исключить все, кроме opus:

```
codecs_exclude_cdn=mpeg4-generic,alaw,ulaw,g729,speex16,g722,telephone-event,flv
```



Настройка SDP для публикации и воспроизведения RTMP потока в CDN

flash_handler_publish.sdp

flash_handler_publish.sdp используется в качестве SDP при публикации в случае подключения к потоку подписчика до того, как WCS получил аудио и видео от публикующего клиента (то есть до того, как известен реально используемый кодек).

В этом случае приоритет кодеков будет определяться порядком в flash_handler_publish.sdp. Если же flash_handler_publish.sdp нет, то по умолчанию приоритетным аудио кодеком будет Speex.

Например, с настройками по умолчанию (без flash_handler_publish.sdp) при публикации RTMP (AAC) на Origin и воспроизведении как WebRTC (Opus) на Edge в CDN

- если подписчик подключается после того, как получено аудио от публикующего (то есть известно, что используется AAC): транскодинг AAC-Opus на Edge
- если подписчик подключается до того, как получено аудио от публикующего (используется кодек по умолчанию - Speex): транскодинг AAC-Speex на Origin и транскодинг Speex-Opus на Edge

Если же на Origin есть flash_handler_publish.sdp, в котором первым указан не Speex, а, например, AAC, то AAC будет использоваться и в том случае, если подписчик подключается до того, как получено аудио от публикующего. То есть, если заранее известен кодек публикации, то указав его в flash_handler_publish.sdp, можно избежать дополнительного транскодинга.

flash_handler_publish.sdp не применяется для ограничения кодеков публикации: если исключить какой-либо кодек из flash_handler_publish.sdp, то он все равно может использоваться для публикации.

flash_handler_play.sdp

flash_handler_play.sdp используется только для SDP, предлагаемого подписчику, и не применяется для ограничения кодеков при проброске в CDN. Если исключить какой-либо кодек из flash_handler_play.sdp, то он не будет использоваться для воспроизведения потока RTMP-подписчиком.

Выбор кодека при воспроизведении потока

Если в наборе кодеков подписчика есть такой же кодек, как тот, что используется для публикации, то для воспроизведения будет использован этот кодек (вне зависимости от его приоритета). В противном случае, для воспроизведения будет использован приоритетный кодек подписчика, и будет осуществляться транскодинг.

Настройка таймаутов на чтение и запись в CDN для доставки RTMP потоков

Если CDN используется преимущественно для публикации и воспроизведения RTMP потоков, и по каким-либо причинам на серверах CDN отключены посылки Keep Alive пакетов (например, публикующие и воспроизводящие клиенты не поддерживают Keep Alive), для контроля состояния RTMP-соединенийнеобходимо настроить таймауты на чтение и запись:

1.Таймаут на чтение может применяться на Origin-серверах, на которые производитсятолько публикация RTMP-потоков без воспроизведения их непосредственно с сервера

```
keep_alive.algorithm=NONE
rtmp.server_read_socket_timeout=120
```

2.Таймаут на запись может применяться на Edge-серверах, которые используютсятолько для воспроизведения CDN-потоков

```
keep_alive.algorithm=NONE
rtmp.server_write_socket_timeout=120
```

3.Таймаут на чтение и запись может применяться и наOrigin, и на Edge-серверах. При этом возможно как подключение подписчиков для воспроизведения потока, так и публикаторов.

```
keep_alive.algorithm=NONE
rtmp.server_socket_timeout=120
```

Краткое руководство по тестированию

Подготовка к тестированию

1. Для тестирования используем:

- два WCS сервера;
- веб-приложениеTwo Way Streamingдля публикации потока
- веб-приложениеPlayerдля воспроизведения потока.

Настройка WCS

2. Для тестирования настройте CDN в минимальной конфигурации: один Origin и один Edge-сервер

Настройки Origin:

```
cdn_enabled=true
cdn_nodes_resolve_ip=true
cdn_ip=origin.flashphoner.com
cdn_role=origin
```

Настройки Edge:

```
cdn_enabled=true
cdn_nodes_resolve_ip=true
cdn_point_of_entry=origin.flashphoner.com
cdn_ip=edge.flashphoner.com
cdn_role=edge
```

Здесь origin.flashphoner.com и edge.flashphoner.com - примеры имен WCS-серверов
Перезапустите Origin и Edge-серверы. Войдите винтерфейс командной строкиEdge-сервера, авторизуйтесь и введите команду

```
show cdn-nodes
```

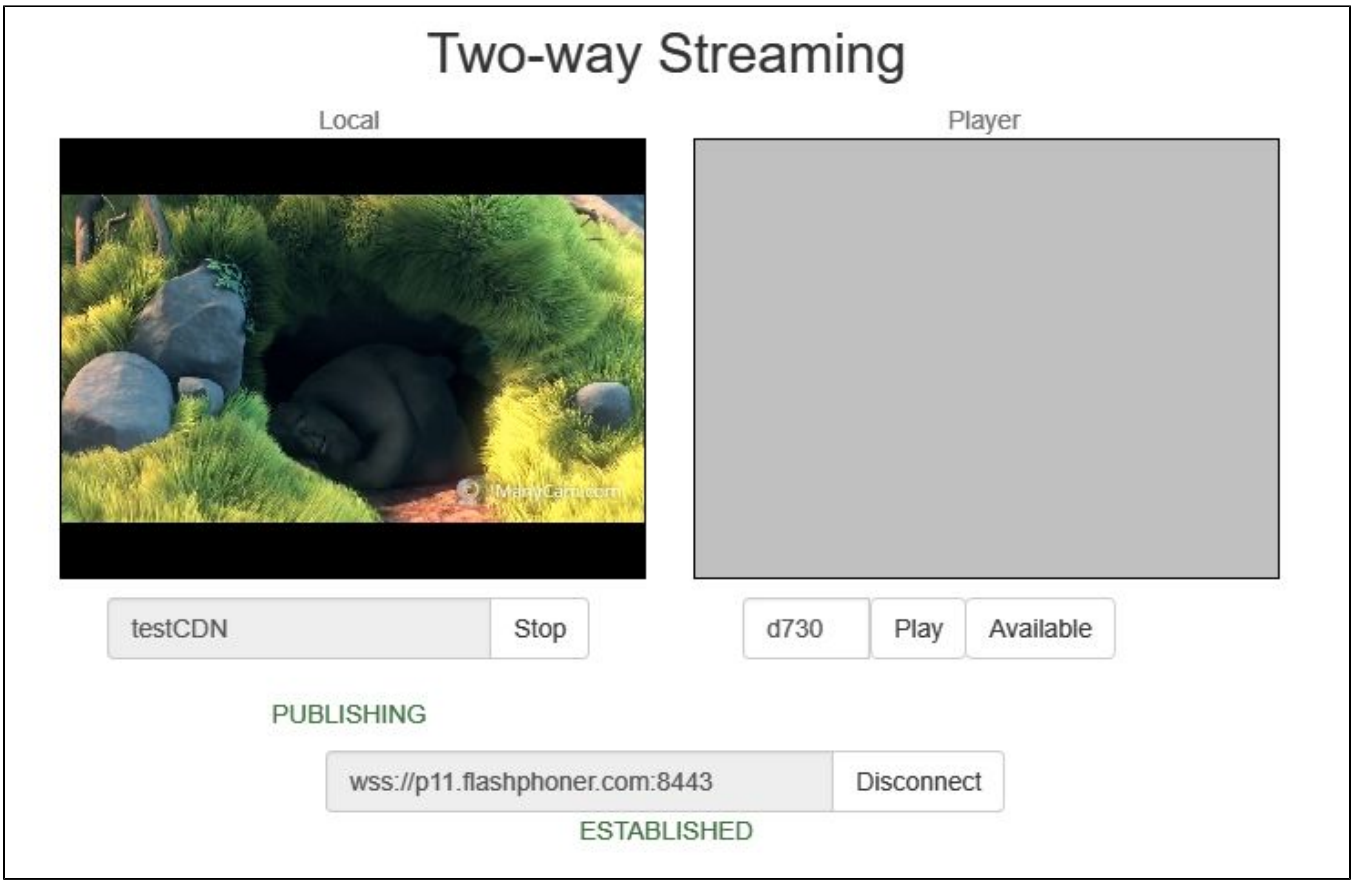
Результат выполнения команды будет следующим:

```
> show cdn-nodes
Ip                  State
--------------------------
origin.flashphoner.com ACTIVE
```

Запуск трансляции с веб-камеры на Origin-сервер

3. Откройте веб-приложение Two Way Streaming на Origin-сервере. Нажмите кнопку Connect и укажите имя транслируемого потока testCDN:

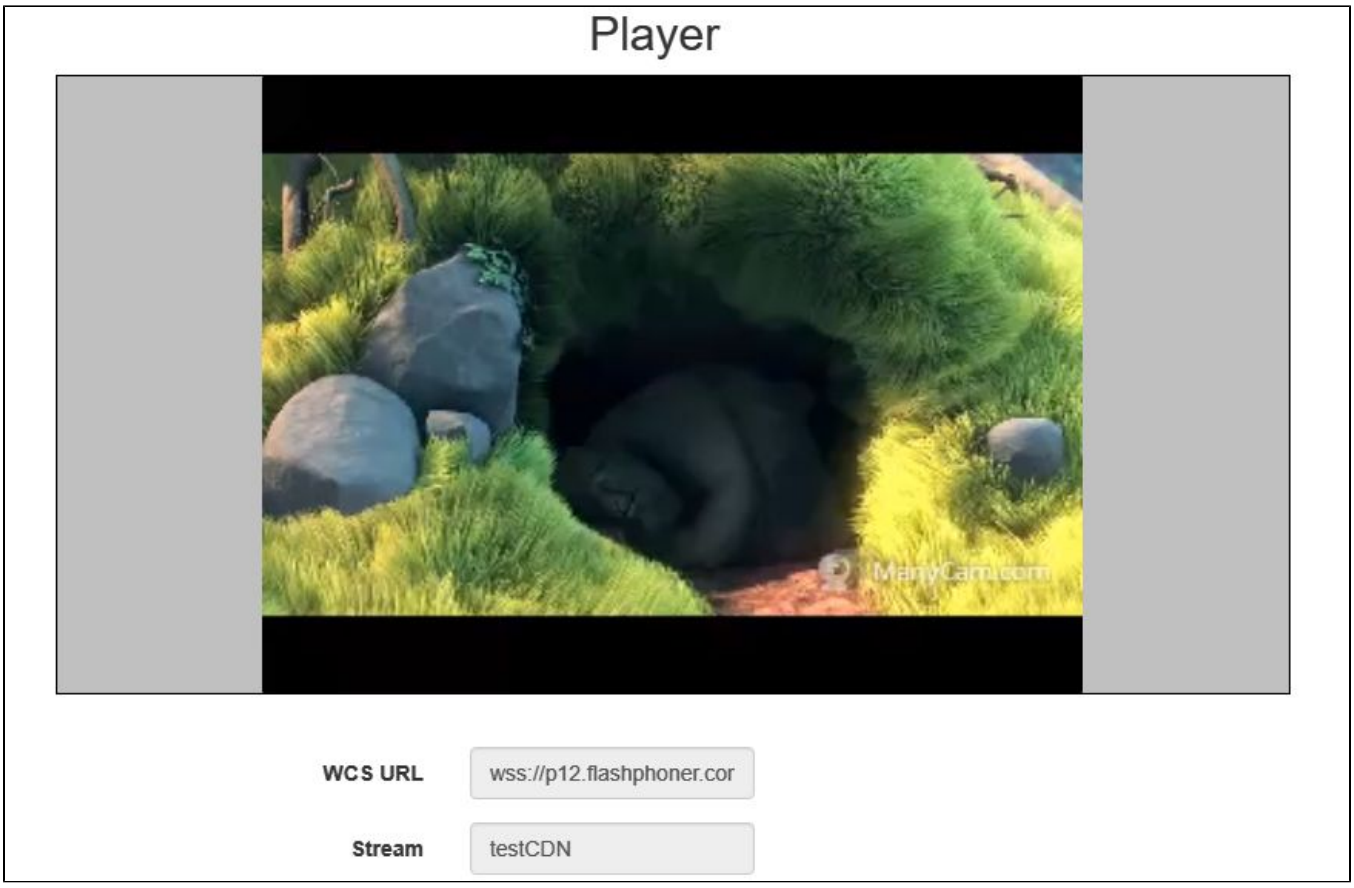
4. Нажмите кнопку Publish. Начнется трансляция потока с веб-камеры:



Воспроизведение потока на Edge-сервере

5. Откройте веб-приложение Player на Edge-сервере. Укажите имя потока, транслируемого на Origin-сервер, testCDN:

6. Нажмите кнопку Start. Начнется воспроизведение потока testCDN:



7. В[интерфейсе командной строки](#)Edge-сервера введите команду

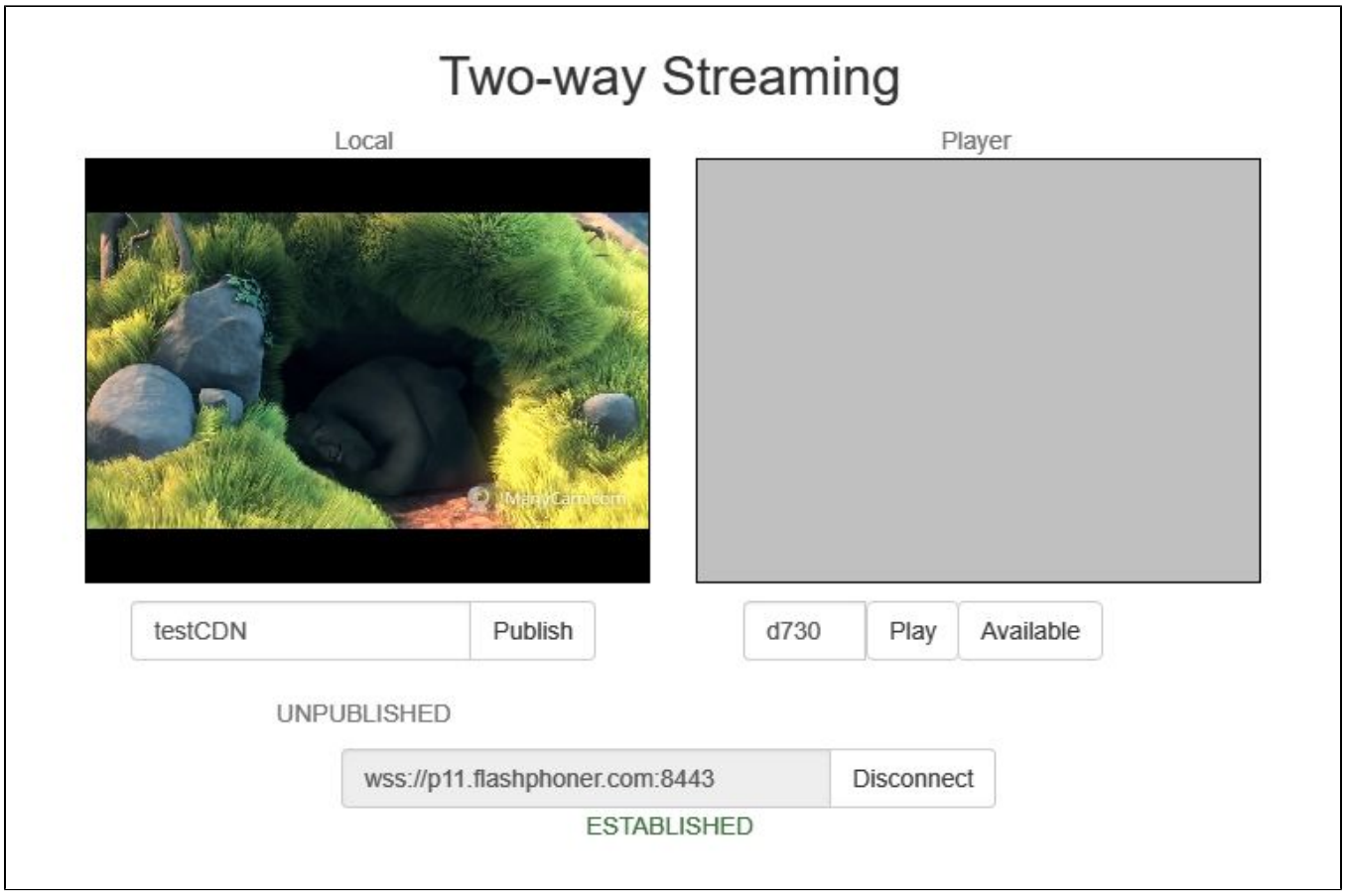
```
show cdn-routes
```

Результат выполнения команды будет следующим:

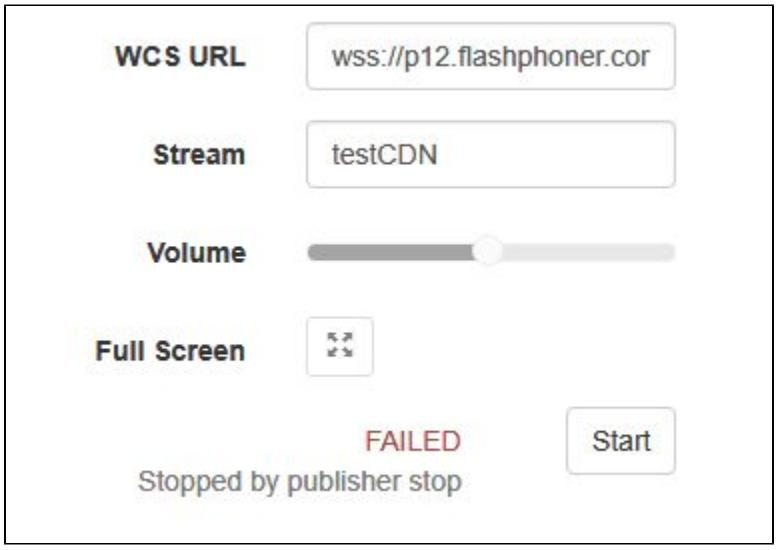
```
> show cdn-routes
Route                   Stream
-------------------------------
0-origin.flashphoner.com-0 testCDN
```

Остановка трансляции с веб-камеры на Origin-сервер

8. Нажмите кнопку Stop на странице публикации потока. Публикация остановится:



При этом на странице воспроизведения Edge-сервера проигрывание потока также остановится с вывоом сообщения "Stopped by publisher stop"



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, но не с другого Origin сервера в той же CDN.

Известные проблемы

1. Изменение настройки кодеков на Edge сервере может приводить к возрастанию нагрузки на сервер в результате включения транскодинга

Симптомы: при ретрансляции большого количества потоков возрастает нагрузка на CPU Edge-сервера
Решение:согласовать настройку кодековна Origin и Edge серверах таким образом, чтобы исключить излишнее включение транскодинга

Например, если в настройке Origin сервера указан следующий набор кодеков:

```
codecs=opus,mpeg4-generic,alaw,ulaw,g729,g722,telephone-event,h264,vp8
```

а в настройке Edge сервера - такой

```
codecs=opus,speex16,mpeg4-generic,g729,g722,h264,vp8
```

то при ретрансляции потока на Edge сервере включается транскодирование звука в Speex16 или AAC, что при большом количестве потоков существенно увеличивает нагрузку на CPU сервера

2. Воспроизведение потока на Edge сервере может останавливаться при отключении Keep Alive на Origin сервере

Симптомы: при настройке на Origin сервере

```
keep_alive.algorithm=NONE
```

воспроизведение потока на Edge сервере отключается даже при установке

```
wcs_agent_session_use_keep_alive_timeout=false
```

Решение: на Origin сервере отключить таймаут на чтение WebSocket при помощи настройки

```
ws_read_socket_timeout=false
```