В браузере по HLS

Описание

HTTP Live Streaming (HLS) — это технология воспроизведения потокового видео по протоколу HTTP, разработанная Apple. HLS видеопоток кодируется в H.264 и AAC и проигрывается на любом совместимом устройстве, браузере или плеере.

Web Call Server конвертирует в HLS видео, полученное издругихподдерживаемых источниковтрансляции, таких как веб-камеры и профессиональные устройства видеозахвата, SIP-звонки и т.д..

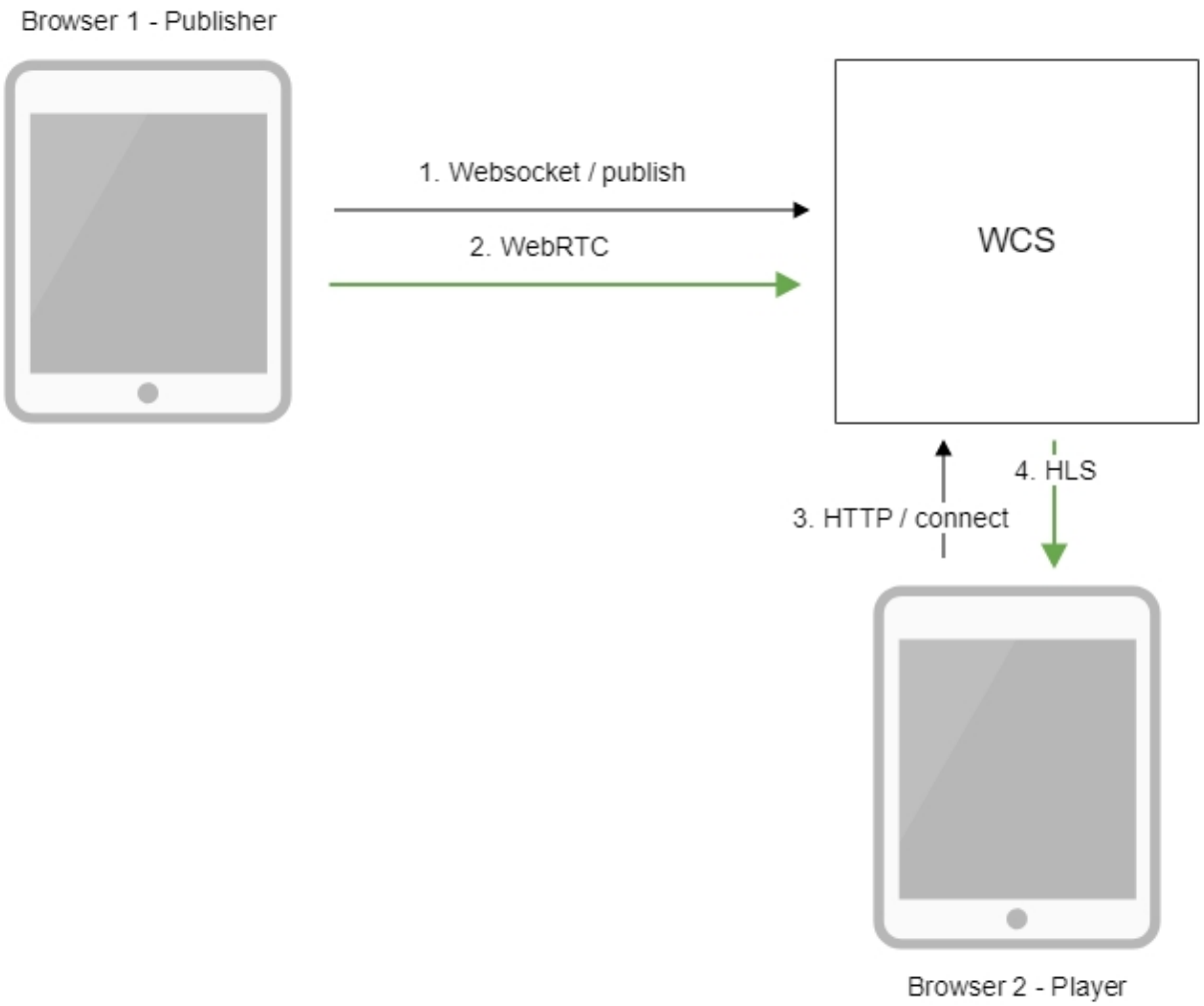
Поддерживаемые платформы и браузеры

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

Поддерживаемые кодеки

- Видео: H.264
- Аудио: AAC

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. Второй браузер устанавливает соединение по HTTP.
4. Второй браузер получает HLS поток и воспроизводит этот поток на странице.

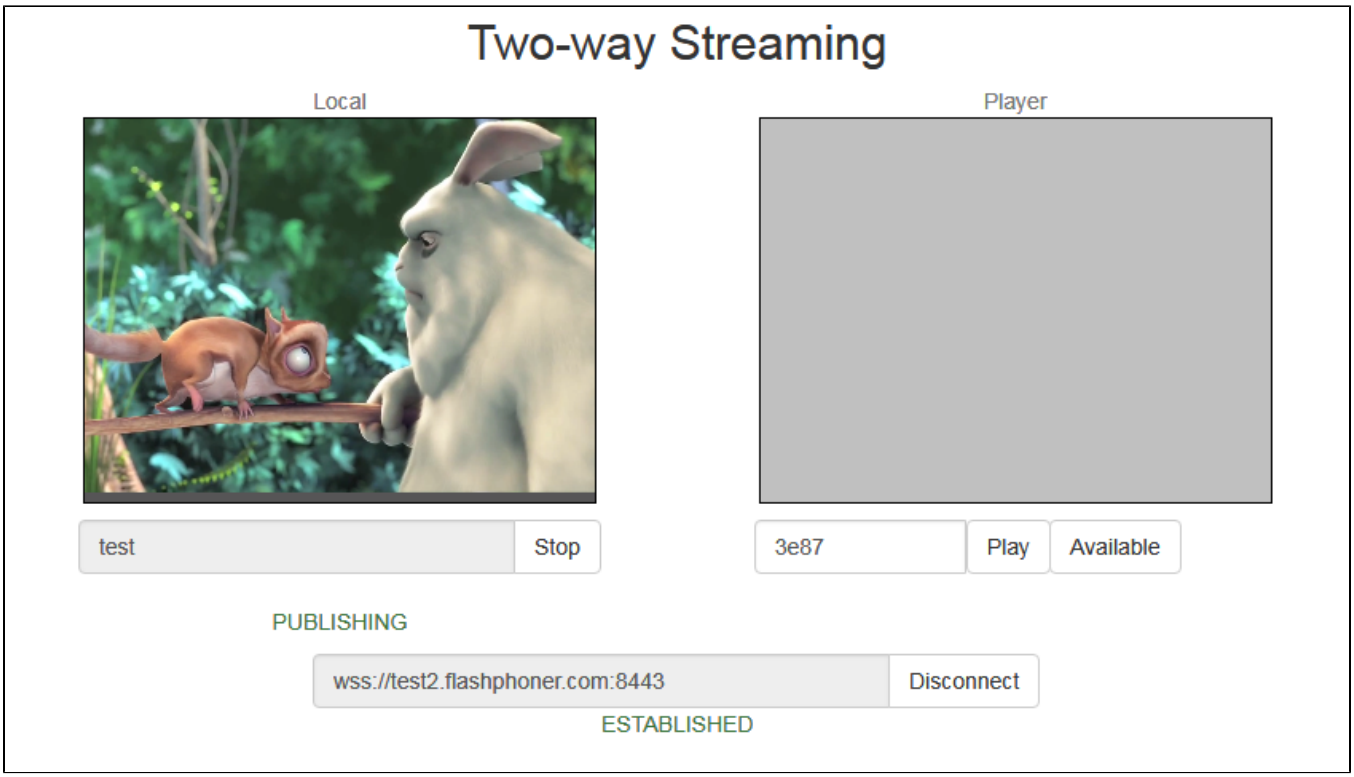
Краткое руководство по тестированию

Трансляция видеопотока на сервер и воспроизведение его по HLS в браузере

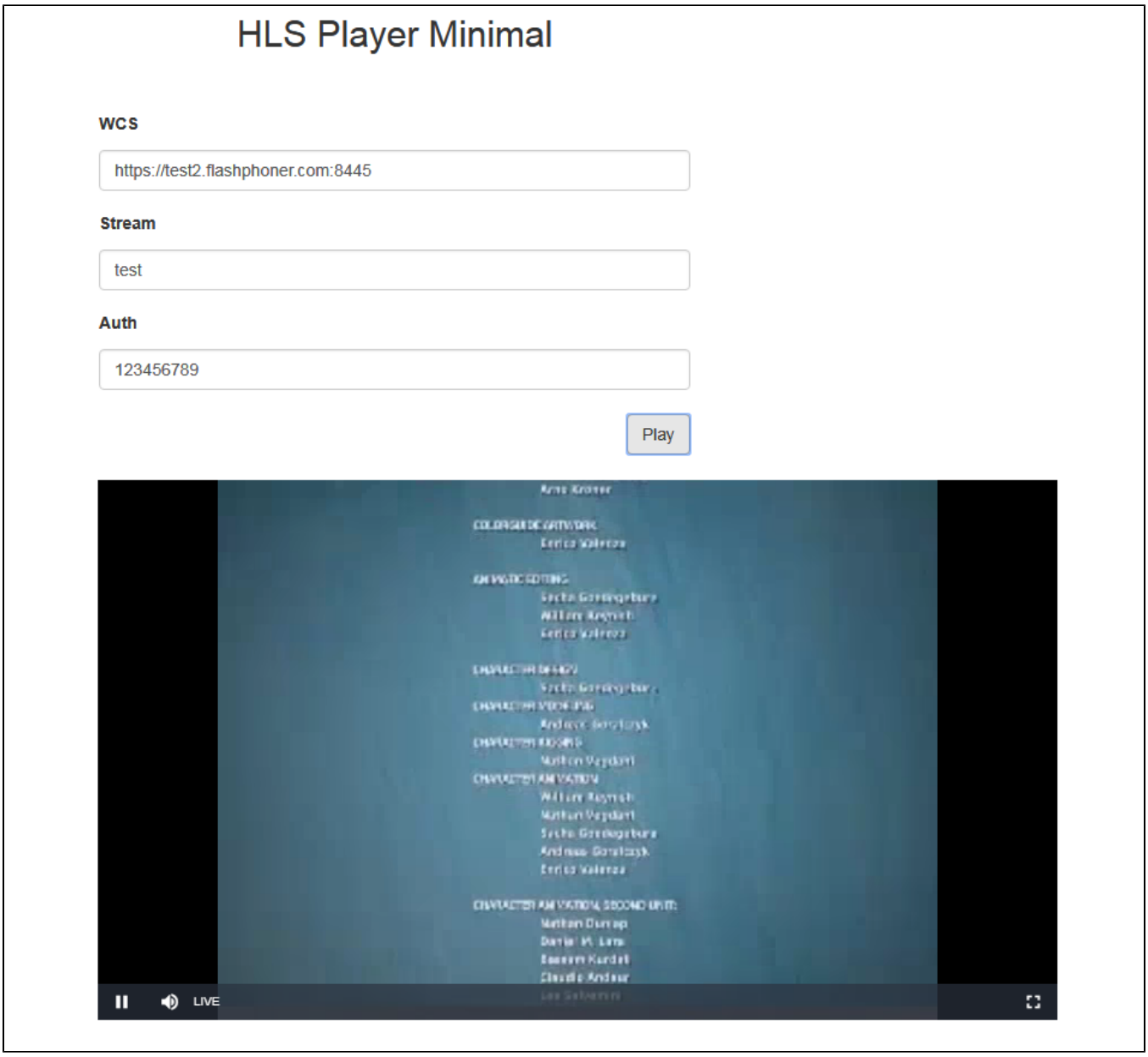
1. Для теста используем:

- WCS сервер
- веб-приложениеTwo Way Streamingдля публикации потока
- веб-приложениеHLS Player Minimalдля воспроизведения потока

2. Откройте веб-приложение Two Way Streaming. Нажмите Connect, затем Publish. Скопируйте идентификатор потока:



3. Откройте веб-приложение HLS Player Minimal. Укажите в поле Stream идентификатор потока и нажмите Play. начнется воспроизведение потока:

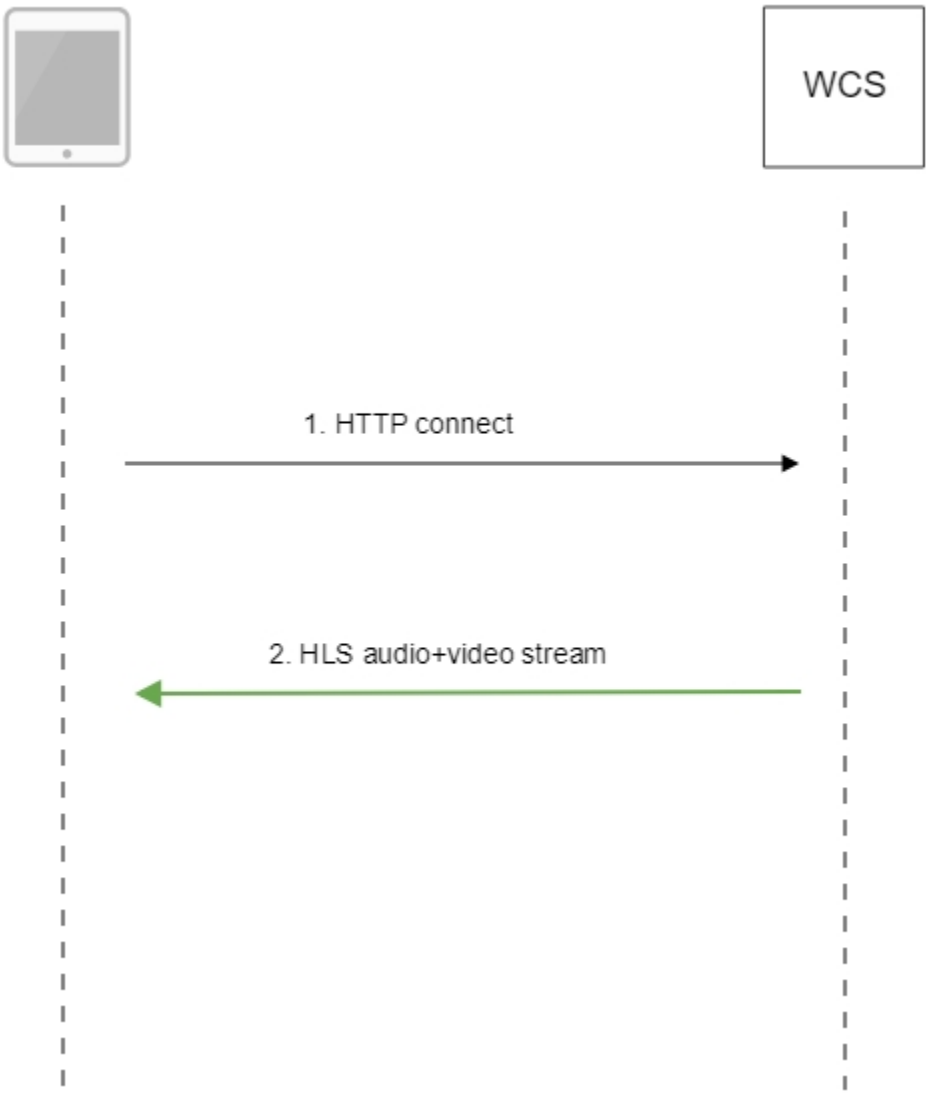


Последовательность выполнения операций (Call flow)

Ниже описана последовательность вызовов при использовании примера HLS Player Minimal для воспроизведения потока по HLS

hls-player.html

hls-player.js



1. Обращение к серверу и воспроизведение.

code

```
var player = videojs('remoteVideo');
```

URL HLS

code

```
        player.src({
            src: $("#urlServer").val() + "/" + streamName + "/" + streamName + ".m3u8",
            type: "application/vnd.apple.mpegurl"
        });
```

Запуск воспроизведения

code

```
player.play();
```

2. Получение HLS-потока от сервера

Типы потоков, воспроизводимых по HLS

По HLS может быть воспроизведен любой поток, опубликованный на WCS под заданным именем

```
http://wcs:8082/streamName/streamName.m3u8
```

Имя может быть задано при публикации из браузераили RTMP кодировщика, либо захвате RTSP, RTMP или VOD потока при помощи REST API

Начиная со сборки5.2.771,можно указать URI RTSP

```
http://wcs:8082/rtsp%3A%2F%2Frtspserver%2Flive.sdp/rtsp%3A%2F%2Frtspserver%2Flive.sdp.m3u8
```

RTMP потока

```
http://wcs:8082/rtmp%3A%2F%2Frtmpserver%3A1935%2Flive%2Fstream/rtmp%3A%2F%2Frtmpserver%3A1935%2Flive%2Fstream.
m3u8
```

или файла для VOD live трансляции

```
http://wcs:8082/vod-live%3A%2F%2Ffile.mp4/vod-live%3A%2F%2Ffile.mp4.m3u8
```

В этом случае поток будет захвачен из указанного источника, и после публикации на сервере начнется его проигрывание по HLS. Обратите внимание, что URI должен быть закодирован, все символы, кроме алфавитно-цифровых, должны быть экранированы.

При обращении к Edge серверу в CDN, если поток с указанным именем или URI опубликован на Origin сервере, по HLS начнет проигрываться поток из CDN. Если такого потока в CDN нет, Edge попытается захватить поток по указанному URI локально.

Аутентификация воспроизведения HLS с помощью REST hook

При необходимости, может быть настроена аутентификация клиентов для воспроизведения потока по HLS. В файле flashphoner.properties должна быть установлена настройка

```
hls_auth_enabled=true
```

При обращении к потоку на клиенте в HLS URL необходимо добавить параметрс указанием токена, полученного, например, от бэкенд-сервера. Наименование параметра задается настройкой

```
client_acl_property_name=aclAuth
```

В этом случае, обращение к потоку должно быть сформировано следующим образом:

```
        var src = $("#urlServer").val() + "/" + streamName + "/" + streamName + ".m3u8";
        var token = $("#token").val();
        if (token.length > 0) {
            src += "?aclAuth=" +token;
        }
```

На бэкенд-сервере в приложении defaultApp должен быть реализован REST hook /playHLS. WCS сервер отправляет на бэкенд запрос, содержащий полученный от клиента токен

```
URL:http://localhost:8081/apps/EchoApp/playHLS
OBJECT:
{
  "nodeId" : "NTk1tLorQ00llGbPJuFexrKceubGCR0k@192.168.1.5",
  "appKey" : "defaultApp",
  "sessionId" : "/192.168.1.100:59473/192.168.1.5:8445",
  "mediaSessionId" : "60709c5b-6950-40c3-8a3d-37ea0827ae32-
727473703a2f2f73747238312e63726561636173742e636f6d2f6772616e646c696c6c6574762f6c6f77-HLS",
  "name" : "test",
  "mediaProvider" : "HLS",
  "custom" : {
    "aclAuth" : "12345789"
  }
}
```

Бэкенд сервер должен вернуть 200 OK, если токен клиента проходит проверку, и 403 Forbidden, если не проходит. В свою очередь, клиент получает либо HLS поток, либо 401Unauthorized.

Настройка

```
hls_auth_token_cache=10
```

задает время кэширования токена в секундах (по умолчанию 10 секунд). До тех пор, пока токен находится в кэше, т.е. либо есть подписчик потока с таким токеном, либо не истекло указанное время, запросы /playHLSc этим токеном не отправляются на бэкенд. Если настройка кэширования установлена в 0

```
hls_auth_token_cache=0
```

запросы /playHLS отправляются на бэкенд при каждом HTTP GET запросе от клиента.

Эти настройки могут быть изменены без перезапуска сервера. При этом настройка`hls_auth_enabled` влияет на существующих подписчиков, а настройка`hls_auth_token_cache` на новые подключения.

Использование собственного приложения на бэкенде для аутентификации

В сборке5.2.1008добавлена возможность указать ключ приложения для аутентификации в HLS URL, например

```
http://wcs:8445/streamName/streamName.m3u8?appKey=customAppKey&aclAuth=1254789
```

В этом случае запрос /playHLS будет отправлен в указанное приложение (customAppKey в примере выше)

Предотвращение несанкционированного доступа к сегментам HLS

Для снижения нагрузки на сервер, проверка токена, а также доступности потока в CDN производится при запросе плейлиста. Для защиты отдельных сегментов в сборке5.2.436добавлена настройка

```
hls_segment_name_suffix_randomizer_enabled=true
```

В этом случае к имени файла сегмента добавляется случайным образом сгенерированный суффикс, например

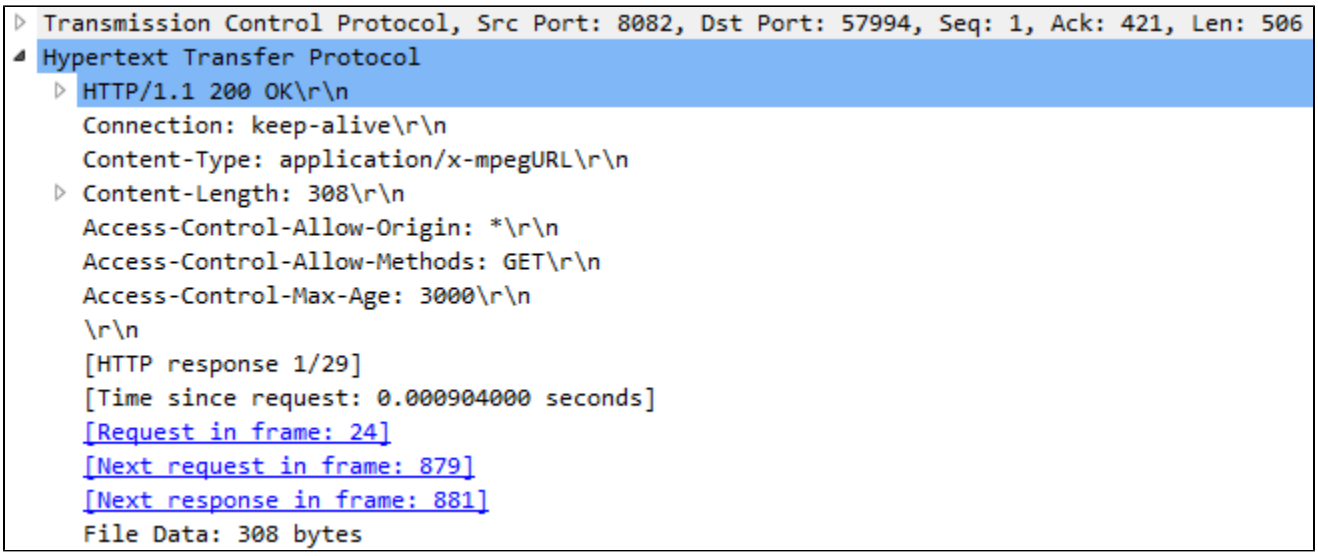
```
test16d2da4658f4374953a120f3c95bc715ea.ts
```

Таким образом, исключается перебор сегментов на стороне клиента.

Отметим, что суффикс не добавляется к сегментам прелоадера.

Добавление HTTP-заголовков для управления кросс-доменным воспроизведением HLS

По умолчанию, в ответ 200 OK на запрос HTTP GET добавляются следующие заголовки:

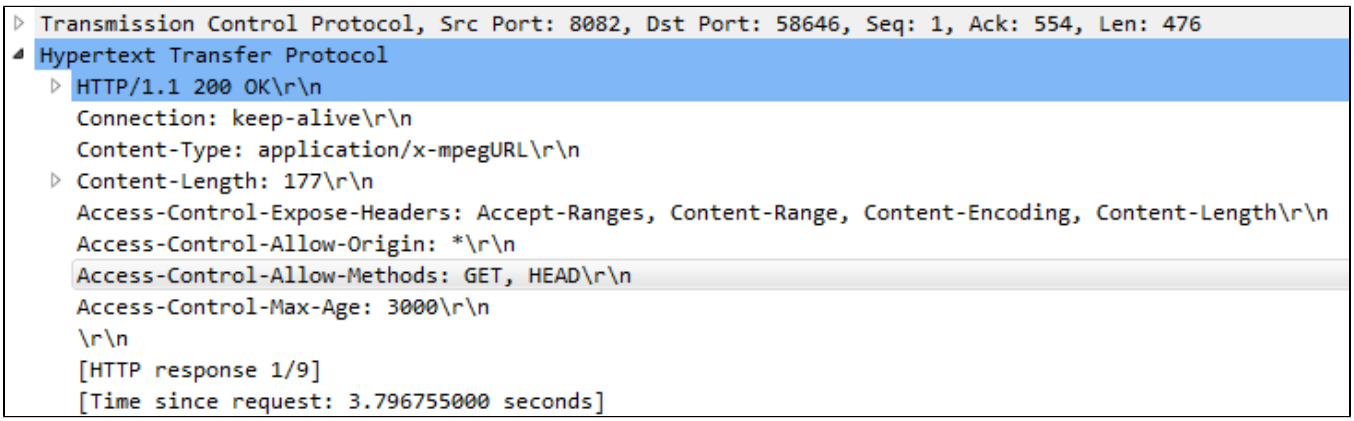



```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET
Access-Control-Max-Age: 3000
```

При необходимости, если, например, воспроизводимый контент и страница HLS плеера находятся в разных доменах, можно добавить собственные заголовки при помощи следующей настройки вфайлеflashphoner.properties:

```
hls_access_control_headers=Access-Control-Allow-Origin: *;Access-Control-Allow-Methods: GET, HEAD;Access-Control-Max-Age: 3000;Access-Control-Expose-Headers: Accept-Ranges, Content-Range, Content-Encoding, Content-Length
```

В этом случае в ответ 200 OK будут добавлены заголовки, перечисленные в настройке:



Поддержка маски в ACAO заголовке

В некоторых случаях, например, при использовании балансировщика нагрузки AWS LB, в ACAO заголовке, передаваемом в ответ на запрос GET, необходимо указать источник запроса с точностью до порта, например

```
Access-Control-Allow-Origin: http://test.flashphoner.com:8081
```

Однако, при конфигурировании сервера, этот адрес не всегда известен. В связи с этим в сборке5.2.755была добавлена настройка, которая включает поддержку маски при указании ACAO заголовкав настройках сервера

```
hls_acao_header_domain_mask=true
```

По умолчанию, данная возможность включена. При этом, если указать в настройке символ '*'

```
hls_access_control_headers=Access-Control-Allow-Origin: *
```

сервер вернет в ответе на запрос GET ACAO заголовок с указанием источника запроса

```
Access-Control-Allow-Origin: http://lb.yourdomain.com:8081
```

При необходимости, данное поведение можно отключить

```
hls_acao_header_domain_mask=false
```

Использование nginx в качестве обратного прокси для воспроизведения по HLS

В некоторых случаях для воспроизведения потока с сервера по HLS может быть использован веб-сервер nginx в качестве обратного прокси. Как правило, это может потребоваться для обхода ограничений на кросс-доменные запросы к различным портам, если добавление HTTP-заголовков не помогает.

Например, если браузер требует, чтобы страница HLS-плеера и HLS-поток находились в одном домене `your.domain` и были доступны по одному и тому же порту 443 (HTTPS), nginx должен быть настроен следующим образом:

```
#  HTTP-  80   443
server {
    listen 80;
    server_name docs.flashphoner.com;
        return 301 https://$server_name$request_uri;
}

#   HTTPS  443
server {
            listen 443 ssl;
            ssl_certificate /etc/letsencrypt/live/your.domain/fullchain.pem;
            ssl_certificate_key /etc/letsencrypt/live/your.domain/privkey.pem;
            server_name your.domain;
            server_tokens off;
            client_max_body_size 500m;
            proxy_read_timeout 10m;

    root         /usr/share/nginx/html;


    location / {
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }

    #  -      https://your.domain/client2
    location /client2/ {
        alias /usr/local/FlashphonerWebCallServer/client2/;
    }

    #         443 ,  https://your.domain/test.m3u8
            location ~* ^.+.(m3u8|ts)$ {
                proxy_pass https://localhost:8445;
                proxy_http_version  1.1;
                proxy_set_header    Host $server_name:$server_port;
                proxy_set_header    X-Forwarded-Host $http_host;
                proxy_set_header    X-Forwarded-Proto $scheme;
                proxy_set_header    X-Forwarded-For $remote_addr;
                proxy_set_header    Upgrade $http_upgrade;
                proxy_set_header    Connection "upgrade";
            }

}
```

Может оказаться полезным кэширование HLS-потока. В этом случае nginx должен быть настроен следующим образом:

1. В секции `http` файла настроек /etc/nginx.conf указываются параметры кэша

```
proxy_cache_path /var/cache/nginx/proxy levels=1:2 keys_zone=proxy_cache:1024m max_size=2048m inactive=10d;
proxy_cache_min_uses 1;
proxy_ignore_headers X-Accel-Expires;
proxy_ignore_headers Expires;
proxy_ignore_headers Cache-Control;
```

2. В секции `server` файла настроек сайта настраивается кэширование HLS-сегментов, при этом плейлисты не должны кэшироваться:

```
        location ~* ^.+.(ts)$ {
            proxy_pass https://localhost:8445;
            proxy_http_version  1.1;
            proxy_set_header    Host $server_name:$server_port;
            proxy_set_header    X-Forwarded-Host $http_host;
            proxy_set_header    X-Forwarded-Proto $scheme;
            proxy_set_header    X-Forwarded-For $remote_addr;
            proxy_set_header    Upgrade $http_upgrade;
            proxy_set_header    Connection "upgrade";
    proxy_cache proxy_cache;
    proxy_cache_key $host$uri$is_args$args;
    proxy_cache_valid 200 2m;
       }

        location ~* ^.+.(m3u8)$ {
            proxy_pass https://localhost:8445;
            proxy_http_version  1.1;
            proxy_set_header    Host $server_name:$server_port;
            proxy_set_header    X-Forwarded-Host $http_host;
            proxy_set_header    X-Forwarded-Proto $scheme;
            proxy_set_header    X-Forwarded-For $remote_addr;
            proxy_set_header    Upgrade $http_upgrade;
            proxy_set_header    Connection "upgrade";
    proxy_cache off;
    expires -1;
       }
```

Отображение статических HTML страниц на порту HLS

Еще один способ обхода ограничений на кросс-доменные запросы в браузере - отображение статического контента, например, страницы плеера, на том же порту, который отдает HLS контент. Чтобы включить данную возможность, необходимо указать следующую настройку в файлеflashphoner.properties

```
hls_static_enabled=true
```

Страница плеера должна располагаться в каталоге, определяемом настройкой

```
hls_static_dir=client2/examples/demo/streaming/hls_static
```

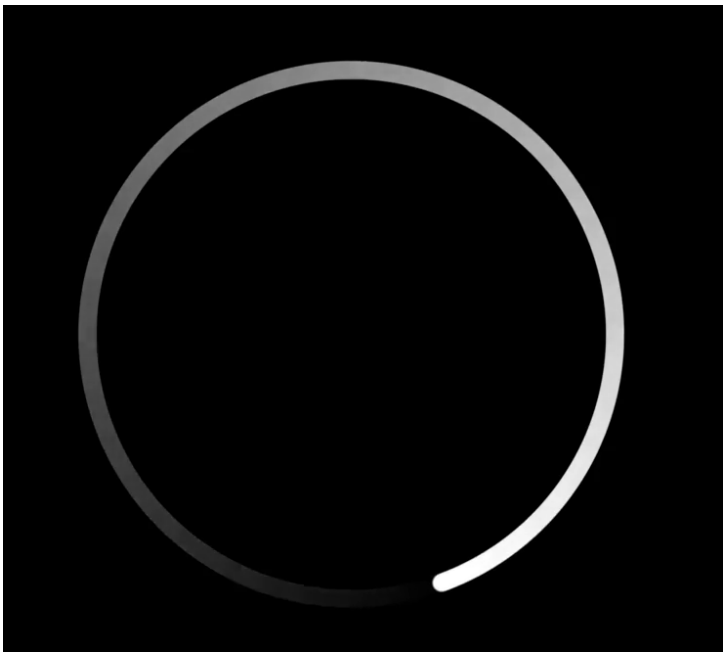
В данном случае (по умолчанию), путь к файлам страницы указан относительно каталога установки WCS.Может быть указан и полный путь, например

```
hls_static_dir=/var/www/html/hls_static
```

Если отображение статического контента включено, при обращении к WCS серверу по адресуhttps://host:8445/hls-player.htmlбраузер отобразит страницу HLS плеера. Если данная возможность отключена, при обращении по такому адресу сервер вернет ошибку 404 Not found.

Preloader для воспроизведения потока по HLS

При подключении первого HLS-подписчика к потоку, в особенности к потоку из CDN, необходимо определенное время, чтобы началась нарезка потока на HLS-сегменты, и был сформирован плейлист. В результате, браузер Safari на устройствах iOS может не подключиться к потоку по HLS с первой попытки. Чтобы подключение всегда проходило успешно, в сборке5.2.371добавлено воспроизведение прелоадера.Пре лоадер по умолчанию выглядит следующим образом



В сборке5.2.408прелоадеры разделены по соотношениям сторон картинки потока: 16:9, 4:3, 2:1

Сегменты прелоадера по умолчанию записываются в каталог`/usr/local/FlashphonerWebCallserver/hls/.preloader` при запуске сервера

```
tree /usr/local/FlashphonerWebCallServer/hls/.preloader
/usr/local/FlashphonerWebCallServer/hls/.preloader
 16x9
    index0.ts
    index10.ts
    index11.ts
    index12.ts
    index13.ts
    index14.ts
    index15.ts
    index16.ts
    index17.ts
    index18.ts
    index19.ts
    index1.ts
    index2.ts
    index3.ts
    index4.ts
    index5.ts
    index6.ts
    index7.ts
    index8.ts
    index9.ts
 2x1
    index0.ts
    index10.ts
    index11.ts
    index12.ts
    index13.ts
    index14.ts
    index15.ts
    index16.ts
    index17.ts
    index18.ts
    index19.ts
    index1.ts
    index2.ts
    index3.ts
    index4.ts
    index5.ts
    index6.ts
    index7.ts
    index8.ts
    index9.ts
 4x3
     index0.ts
     index10.ts
     index11.ts
     index12.ts
     index13.ts
     index14.ts
     index15.ts
     index16.ts
     index17.ts
     index18.ts
     index19.ts
     index1.ts
     index2.ts
     index3.ts
     index4.ts
     index5.ts
     index6.ts
     index7.ts
     index8.ts
     index9.ts
```

Минимальная длительность одного сегмента прелоадера по умолчанию составляет 2 секунды, и может быть задана в миллисекундах при помощи настройки

```
hls_preloader_time_min=2000
```

Отключение прелоадера

При необходимости, прелоадер может быть отключен, эта возможность доступна, начиная со сборки5.2.396. Для отключения HLS прелоадера используется параметр

```
hls_preloader_enabled=false
```

Настройка собственного прелоадера

Чтобы заменить прелоадер по умолчанию на собственный, необходимо сделать следующее:

1. Выбрать видеоклип (например, логотип) в трех соотношениях сторон: 16:9, 4:3, 2:1

2. С помощью ffmpeg закодировать видео в H264, добавить к видеоклипу аудиодорожку, задать периодичность ключевых кадров и убрать B-фреймы

```
ffmpeg -i clip16x9.mp4 -f lavfi -i anullsrc=channel_layout=mono:sample_rate=44100 -c:v h264 -g 30 -bf 0 -
shortest 16x9/preloader16x9.mp4
ffmpeg -i clip4x3.mp4 -f lavfi -i anullsrc=channel_layout=mono:sample_rate=44100 -c:v h264 -g 30 -bf 0 -
shortest 4x3/preloader4x3.mp4
ffmpeg -i clip2x1.mp4 -f lavfi -i anullsrc=channel_layout=mono:sample_rate=44100 -c:v h264 -g 30 -bf 0 -
shortest 2x1/preloader2x1.mp4
```

3. Загрузить и установить инструменты для подготовки HLS-сегментовс сайта Apple

4. Подготовить HLS сегменты прелоадера, указав длительность сегмента, например, 2 секунды

```
cd 16x9
mediafilesegmenter -t 2 -B index -start-segments-with-iframe preloader16x9.mp4
tar -cvzf preloader.tar.gz index*.ts
```

Этот шаг необходимо повторить для всех соотношений сторон.

5. На сервере создать каталог для прелоадера

```
mkdir /opt/custom_preloader
mkdir /opt/custom_preloader/16x9
mkdir /opt/custom_preloader/4x3
mkdir /opt/custom_preloader/2x1
```

6. Распаковать прелоадер из архива, подготовленного на шаге 4

```
cd /opt/custom_preloader/16x9
tar -xvzf ~/preloader16x9.tar.gz
```

Этот шаг также необходимо повторить для всех соотношений сторон

7. Указать в настройках сервера расположение прелоадера и длительность одного сегмента

```
hls_preloader_time_min=2000
hls_preloader_dir=/opt/custom_preloader
```

Управление HLS подписками при помощи REST API

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

- HTTP:http://test.flashphoner.com:8081/rest-api/hls/startup

- HTTPS:https://test.flashphoner.com:8444/rest-api/hls/startup

Здесь:

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

REST-методы и статусы ответа

REST-метод	Пример тела REST-запроса	Пример тела REST-ответа	Статусы ответа	Описание
/hls /startup	`{ "name":" test" }`		200 - OK 404 - Stream not found 500 - Internal error	Запустить HLS нарезку указанного потока
/hls /find_all		`{ "test": { "handler": "com.flashphoner.server.client. handler.wcs4.WCS4Handler@74dbf27b", "state": "ACTIVE", "writer": "HLS-test", "streamStatus": "PLAYING", "writerStarted": "true" } }`	200 – OK 404 – Not found	Найти все потоки, для которых есть HLS нарезки
/hls /terminate	`{ "name":" test" }`		200 – OK 404 – Not found	Завершить или перезапустить HLS нарезку указанного потока

Параметры

Имя параметра	Описание	Пример
name	Имя потока, опубликованного на сервере	test

Особенности

1.Если HLS нарезка потока запущена при помощи REST запроса /hls/startup, и нет активных HLS подписчиков, нарезка остановится по истечении интервала в секундах

```
hls_manager_provider_timeout=300
```

По умолчанию, интервал составляет 5 минут. То же касается автоматически созданных HLS нарезок при установленной настройке

```
hls_auto_start=true
```

2. Если HLS нарезка потока останавливается при помощи REST запроса /hls/terminate, и есть активные HLS подписчики, то нарезка будет перезапущена. При этом активные HLS подписчики должны повторно подключиться к потоку.

Поддержка HLS ABR

Устаревшая реализация в сборках5.2.484-5.2.582

В сборке5.2.484добавлена поддержка HLS ABR плейлистов. Использование этой возможности включается при помощи настройки

```
hls_master_playlist_enabled=true
```

Имя основного плейлиста указывается при помощи настройки

```
hls_manifest_file=index.m3u8
```

Браузер должен запросить основной плейлист по URL

```
https://wcs_address:8445/streamName/index.m3u8
```

Здесь

- wcs_address - адрес WCS сервера
- streamName - имя потока на сервере
- index.m3u8 - имя основного плейлиста

При запросе основного плейлиста сервер проверяет наличие потоков, транскодируемых из указанного потока согласно профилям транскодинга, перечисленным в файле настроек cdn_profiles.yml, например:

```
profiles:
  -720p:
    video:
      height: 720
      bitrate: 1000
      codec: h264
  -480p:
    video:
      height: 480
      bitrate: 1000
      codec: h264
  -240p:
    video:
      height: 240
      bitrate: 400
      codec: h264
```

Все транскодированные потоки по профилям, которые в момент запроса опубликованы на сервере, попадают в основной плейлист, например:

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1000000,RESOLUTION=1280x720,CODECS="avc1.42e01f,mp4a.40.2"
../streamName-720p/streamName-720p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1000000,RESOLUTION=852x480,CODECS="avc1.42e01f,mp4a.40.2"
../streamName-480p/streamName-480p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=400000,RESOLUTION=426x240,CODECS="avc1.42e01f,mp4a.40.2"
../streamName-240p/streamName-240p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2500000,RESOLUTION=1280x720,CODECS="avc1.42e01f,mp4a.40.2"
../streamName/streamName.m3u8
```

Затем браузер, в зависимости от пропускной способности канала, переключается между потоками, перечисленными в основном плейлисте.

> В момент, когда браузер запрашивает основной плейлист, потоки уже должны быть опубликованы на сервере и нарезаться на HLS сегменты

Чтобы обеспечить наличие потоков, необходимо:

1. На отдельно стоящем сервере:

1.1. Периодически проверять, транскодируются ли потоки к указанным параметрам, и запускать транскодинг при необходимости при помощи REST API

```
        curl -s -X POST -d "{\"uri\":\"transcoder://tcode_test-240p\",\"remoteStreamName\":\"test\",\"
localStreamName\":\"test-240p\",\"encoder\":{\"width\":320,\"height\":240}}" http://localhost:8081/rest-api
/transcoder/startup
        curl -s -X POST -d "{\"uri\":\"transcoder://tcode_test-480p\",\"remoteStreamName\":\"test\",\"
localStreamName\":\"test-480p\",\"encoder\":{\"width\":640,\"height\":480}}" http://localhost:8081/rest-api
/transcoder/startup
        curl -s -X POST -d "{\"uri\":\"transcoder://tcode_test-720p\",\"remoteStreamName\":\"test\",\"
localStreamName\":\"test-720p\",\"encoder\":{\"width\":1280,\"height\":720}}" http://localhost:8081/rest-api
/transcoder/startup
```

1.2. Периодически запускать HLS потоки для включения в основной плейлист, например

```
curl -s -X POST -d "{\"name\":\"test\"}" http://localhost:8081/rest-api/hls/startup
sleep 1
curl -s -X POST -d "{\"name\":\"test-240p\"}" http://localhost:8081/rest-api/hls/startup
sleep 1
curl -s -X POST -d "{\"name\":\"test-480p\"}" http://localhost:8081/rest-api/hls/startup
sleep 1
curl -s -X POST -d "{\"name\":\"test-720p\"}" http://localhost:8081/rest-api/hls/startup
```

2. На Edge сервере в CDN периодически запрашивать HLS потоки по профилям, например

```
curl -s http://localhost:8082/test/test.m3u8
sleep 1
curl -s http://localhost:8082/test-240p/test-240p.m3u8
sleep 1
curl -s http://localhost:8082/test-480p/test-480p.m3u8
sleep 1
curl -s http://localhost:8082/test-720p/test-720p.m3u8
sleep 1
```

Актуальная реализация в сборках5.2.585и новее

В сборке5.2.585реализация HLS ABR существенно изменена. Как и прежде, HLS ABR может использоваться только в CDN, при этом все транскодированные потоки для вариантов в ABR манифесте Edge забирает одновременно в пределах одной медиасессии, чтобы варианты одного потока были синхронизированы друг с другом. Это требует совместной настройки Transcoder и Edge узлов и накладывает ряд ограничений. Рассмотрим их ниже.

Настройки Transcoder узлов

Для того, чтобы все варианты одного потока были синхронизированы между собой, на Transcoder узлах должно быть включено выравнивание кодирования

```
transcoder_align_encoders=true
```

Кроме того, должен быть включен FPS фильтр

```
video_filter_enable_fps=true
video_filter_fps=25
```

Ключевые фреймы в вариантах потока должны быть синхронизированы. Например, при 25 кадрах в секунду будем отправлять ключевой фрейм каждые 2 секунды

```
video_filter_fps_gop_synchronization=50
```

Настройки HLS Edge узлов

На HLS Edge узлах необходимо отключить использование прелоадера и транскодирование потоков

```
hls_preloader_enabled=false
hls_player_width=0
hls_player_height=0
```

Необходимо также настроить профили транскодирования в файле cdn_profiles.yml

```
profiles:
 -240p:
  audio:
    codec : mpeg4-generic
    rate : 48000
  video:
    height : 240
    bitrate : 300
    gop : 50
    codec : h264
 -480p:
  video:
    height : 480
    bitrate : 600
    gop : 50
    codec : h264
 -720p:
  video:
    height : 720
    bitrate : 1000
    gop : 50
    codec : h264
```

Обратите внимание, что параметры звука можно указать для первого профиля, т.к. для всех профилей эти параметры должны быть идентичными и будут применены по первому из профилей.

Затем необходимо включить HLS ABR

```
hls_abr_enabled=true
```

Использование

Клиент должен запрашивать плейлист HLS ABR, указывая имя потока с суффиксом

```
https://server:8445/test_0-HLS-ABR-STREAM/test_0-HLS-ABR-STREAM.m3u8
```

Суффикс задается при помощи настройки

```
hls_abr_stream_name_suffix=-HLS-ABR-STREAM
```

Плейлист содержит ссылки на плейлисты вариантов потока, между которыми клиент может переключаться

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=614400,RESOLUTION=852x480,CODECS="avc1.42e01f,mp4a.40.2"
-480p/-480p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1024000,RESOLUTION=1278x720,CODECS="avc1.42e01f,mp4a.40.2"
-720p/-720p.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=307200,RESOLUTION=426x240,CODECS="avc1.42e01f,mp4a.40.2"
-240p/-240p.m3u8
```

Если запросить поток без суффикса, будет играть HLS без поддержки ABR.

Предотвращение транскодирования к более высоким разрешениям

Если транскодирование к более высоким разрешениями отключено на HLS ABR Edge сервере

```
cdn_strict_transcoding_boundaries=true
```

то варианты, соответствующие профилям с более высокими разрешениями, для данного потока не будут нарезаться и не будут доступны плееру.

Ограничения

1. HLS Edge может быть использован только для воспроизведения HLS потоков, клиентские сессии с использованием других протоколов работать не будут.

2. Не работают такие функции, как запись, снятие снапшотов, микширование, захват потоков с другого сервера и прочие функции обработки потоков

Хранение сегментов HLS

Использование диска

По умолчанию HLS-сегменты записываются на диск сервера, в каталог /usr/local/FlashphonerWebCallServer/hls. Начиная со сборки 5.2.687, каталог для сохранения сегментов можно изменить при помощи параметра

```
hls_dir=/usr/local/FlashphonerWebCallServer/hls
```

(Расположение прелоадера настраивается отдельно при помощи параметра hls_preloader_dir.)

На диске хранится количество сегментов, соответствующее заданному размеру плейлиста, по умолчанию 10

```
hls_list_size=10
```

Чем меньше количество сегментов в плейлисте, тем меньше задержка при воспроизведении. Однако при коротком плейлисте подписчики с недостаточной пропускной способностью каналов могут запрашивать сегменты, которых уже нет в плейлисте и на диске. В связи с этим, в сборке5.2.581добавлена возможность хранить некоторое число сегментов на диске после их ухода из плейлиста. Эта возможность включается настройкой

```
hls_hold_segments_before_delete=true
```

По умолчанию, будет храниться 5 последних сегментов

```
hls_hold_segments_size=5
```

Например, если плейлист содержит 3 сегмента

```
#EXTM3U
#EXT-X-VERSION:8
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:15
#EXT-X-DISCONTINUITY-SEQUENCE:1
#EXTINF:3.415,
test_017.ts
#EXTINF:10.417,
test_018.ts
#EXTINF:9.084,
test_019.ts
```

на диске будут храниться 3 текущих сегмента из плейлиста и 5 предшествующих

```
test_012.ts
test_013.ts
test_014.ts
test_015.ts
test_016.ts
test_017.ts
test_018.ts
test_019.ts
```

Использование оперативной памяти

При больших нагрузках на сервер, например, если он выделен для раздачи потоков по HLS, чтение сегментов с диска для отправки подписчикам может давать задержки. В этом случае необходимо включить хранение HLS сегментов в памяти

```
hls_store_segment_in_memory=true
```

Для отправки подписчикам сегменты будут считываться из памяти, а также будут записываться на диск, в целях отладки. Необходимо отметить, что в этом случае потребуется больше памяти под Java heap для хранения сегментов.

Отладочные логи для HLS-сессии

Для отчета об ошибке можно, используя CLI, включить сбор отладочных логов для HLS-сессий

```
update node-setting --value true hls_enable_session_debug
```

Следует учесть, что файл настроек flashphoner.properties будет перезаписан после этой команды.

Известные проблемы

1. Невосстанавливаемый фриз HLS потока при воспроизведении в iOS Safariчерез CDN

Симптомы: через одну минуту после начала публикации изображение останавливается, звук продолжает воспроизводиться

Решение:

а) включить транскодинг на сервере при помощи настройки в файле flashphoner.properties

```
disable_streaming_proxy=true
```

б) если включение транскодинга нежелательно, установитьв файлеflashphoner.properties

```
hls_discontinuity_enabled=true
```

в этом случае возможны щелчки в аудиопотоке, но изображение останавливаться не будет.

2. Прекращение записи сегментов HLS при воспроизведении потока. опубликованного в браузере Firefox

Симптомы: через несколько минут после начала воспроизведения прекращается запись HLS-сегментов, при этом директория потока в директории hls не удаляется, в логе сервера продолжают появляться сообщения

```
INFO HLSStreamManager - HLSStreamProviderKeepaliveThread-80 Remove hls channel
```

Для восстановления публикующая сторона должна заново опубликовать поток.

Решение: использовать другой браузер для публикации потока, который предполагается воспроизводить по HLS.

3. При воспроизведении HLS в Safari в iOS 12.4 не играет видео для первого подписчика

Симптомы: при подключении первого подписчика к потоку по HLS вSafari в iOS 12.4 видео не воспроизводится, если HLS-подписчики уже есть, видео играет нормально

Решение: установить минимальное количество сегментов в плейлисте HLS не менее 2

```
hls_min_list_size=2
```

4. При воспроизведении RTMP-потока как HLS в Safari в iOS 12.4 не играет видео для любого подписчика, если активна настройка

```
hls_auto_start=true
```

Симптомы: при подключении подписчиков к RTMP потоку по HLS вSafari в iOS 12.4 видео не воспроизводится

Решение: при публикации файла со звуковой дорожкой стерео, использовать моно звук, например, для ffmpeg указать настройку

```
-acodec aac -ac 1
```

5. Если по HLS проигрывается поток, транскодированный в CDN, и если при этом изменяется соотношение сторон потока, HLS прелоадер отображается в соответствии с соотношением сторон оригинального потока

Симптомы: при заказе транскодированного потока с указанием профиля в имени, например test-640x480p, отображается прелоадер 16:9, если исходный поток опубликован в разрешении 1280x720

Решение: включить на транскодере сохранение соотношения сторон

```
video_transcoder_preserve_aspect_ratio=true
```

6. Если в исходном потоке содержатся B-фреймы, в некоторых плеерах могут наблюдаться подергивания

Симптомы: сильно дергается картинка при проигрывании HLS, возможен эффект низкого FPS

Решение: обновить WCS до сборки 5.2.863, в которой решена данная проблема