В браузере по HLS

- Описание
  - Поддерживаемые платформы и браузеры
  - Поддерживаемые кодеки
  - Схема работы
- Краткое руководство по тестированию
  - Трансляция видеопотока на сервер и воспроизведение его по HLS в браузере
- Последовательность выполнения операций (Call flow)
- Аутентификация воспроизведения HLS с помощью REST hook
- Добавление HTTP-заголовков для управления кросс-доменным воспроизведением HLS
- Использование nginx в качестве обратного прокси для воспроизведения по HLS
- Отображение статических HTML страниц на порту HLS
- Известные проблемы

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..

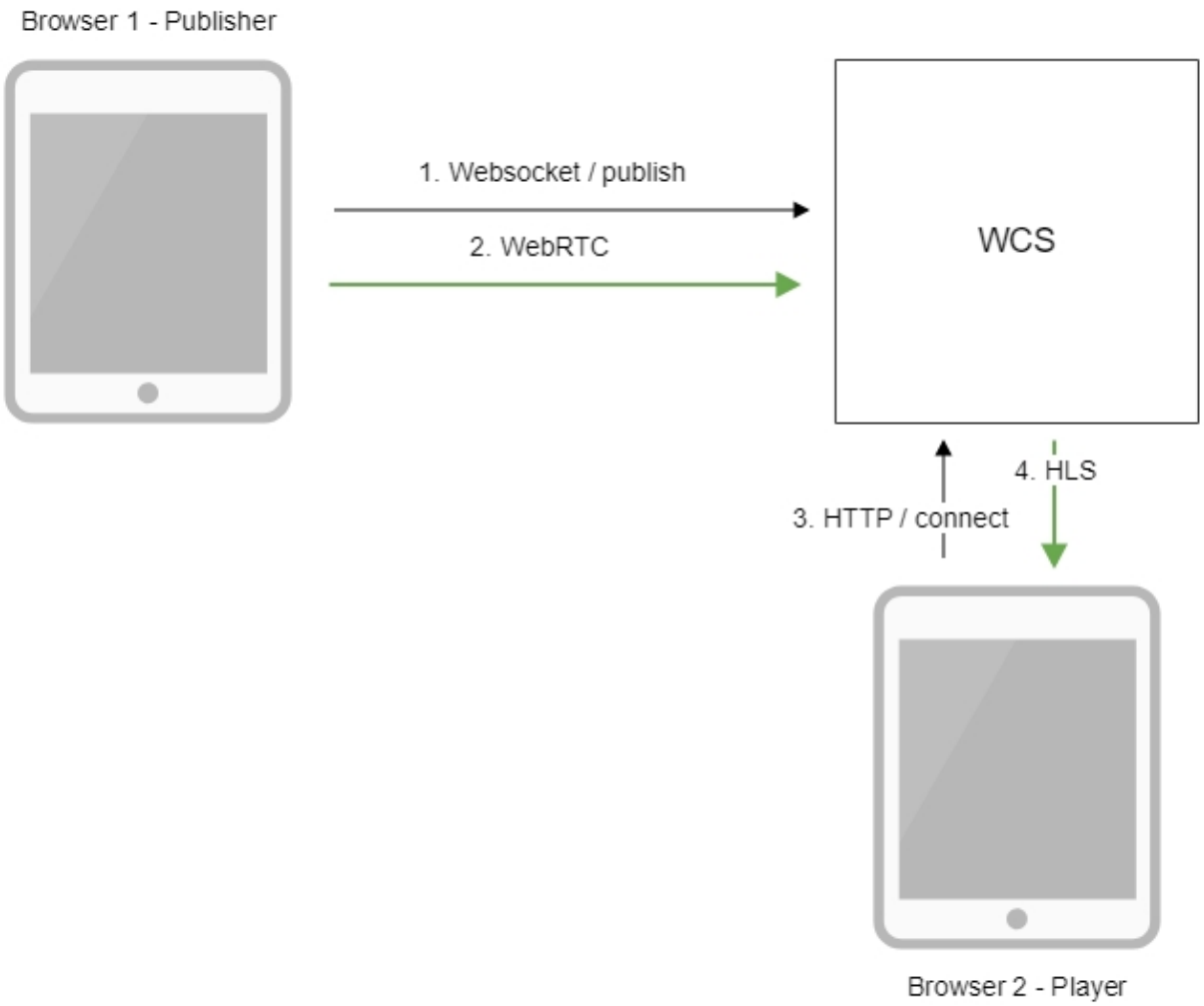
Поддерживаемые платформы и браузеры

	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.

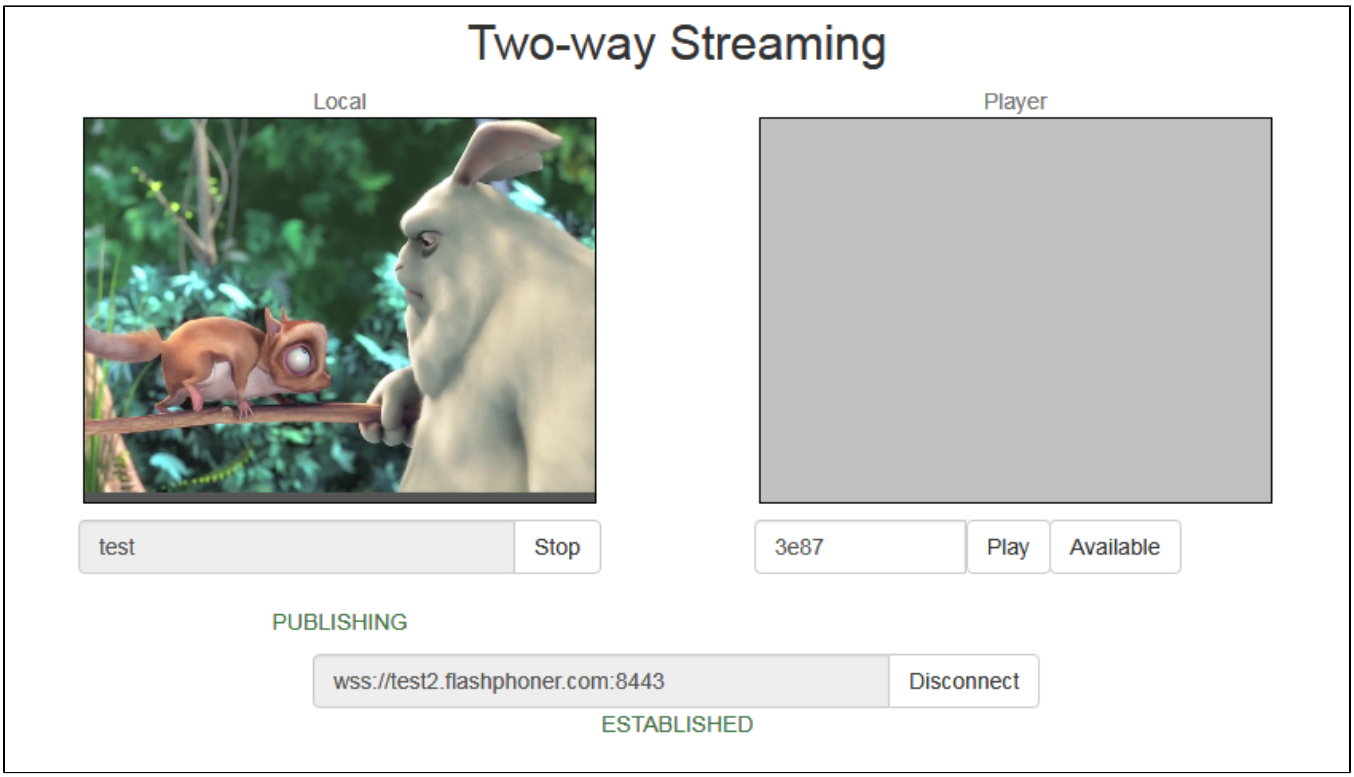
Краткое руководство по тестированию

Трансляция видеопотока на сервер и воспроизведение его по HLS в браузере

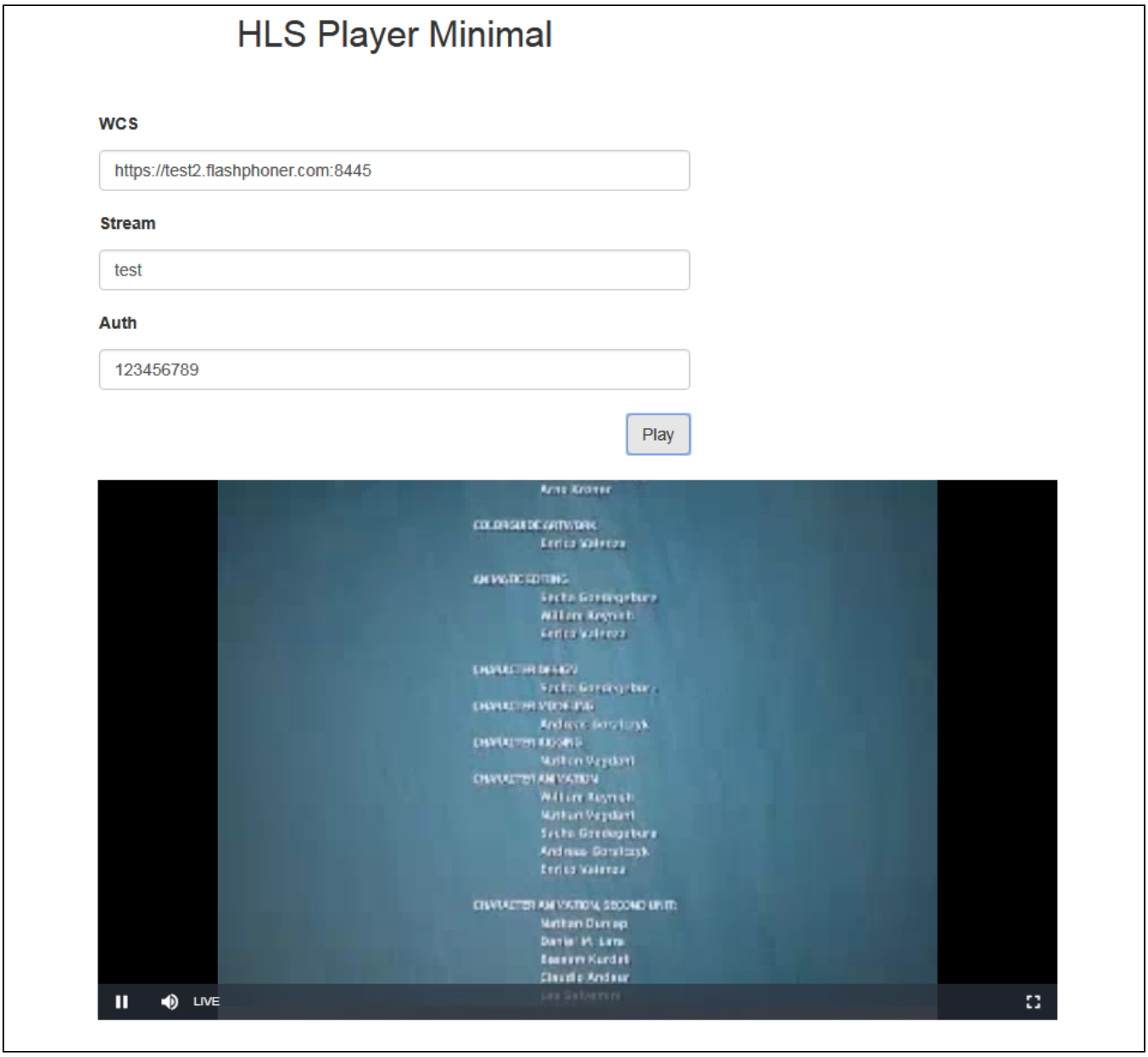
1. Для теста используем:

- 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:

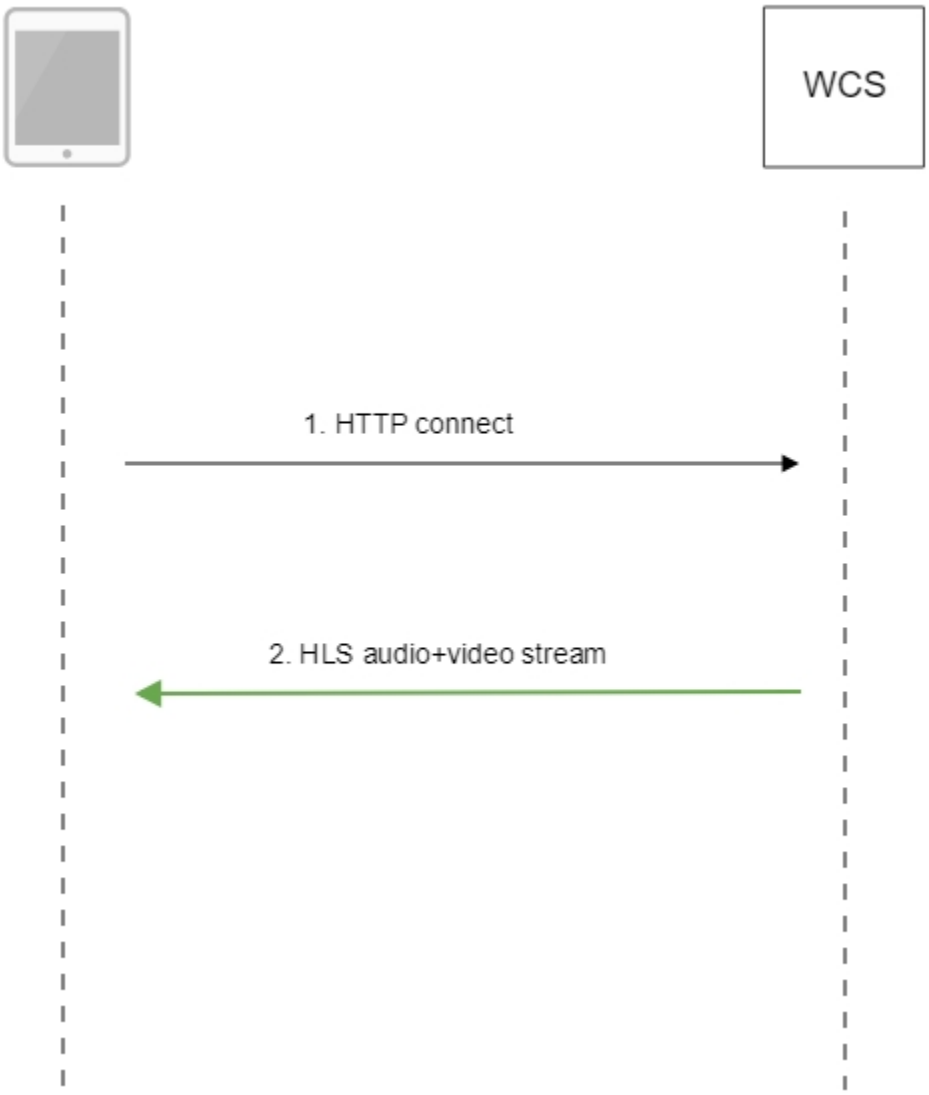


Последовательность выполнения операций (Call flow)

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

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 параметр `token` с указанием токена, полученного, например, от бэкенд-сервера.

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

На бэкенд-сервере должен быть реализован REST hook /playHLS. WCS сервер отправлет 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" : {
    "token" : "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

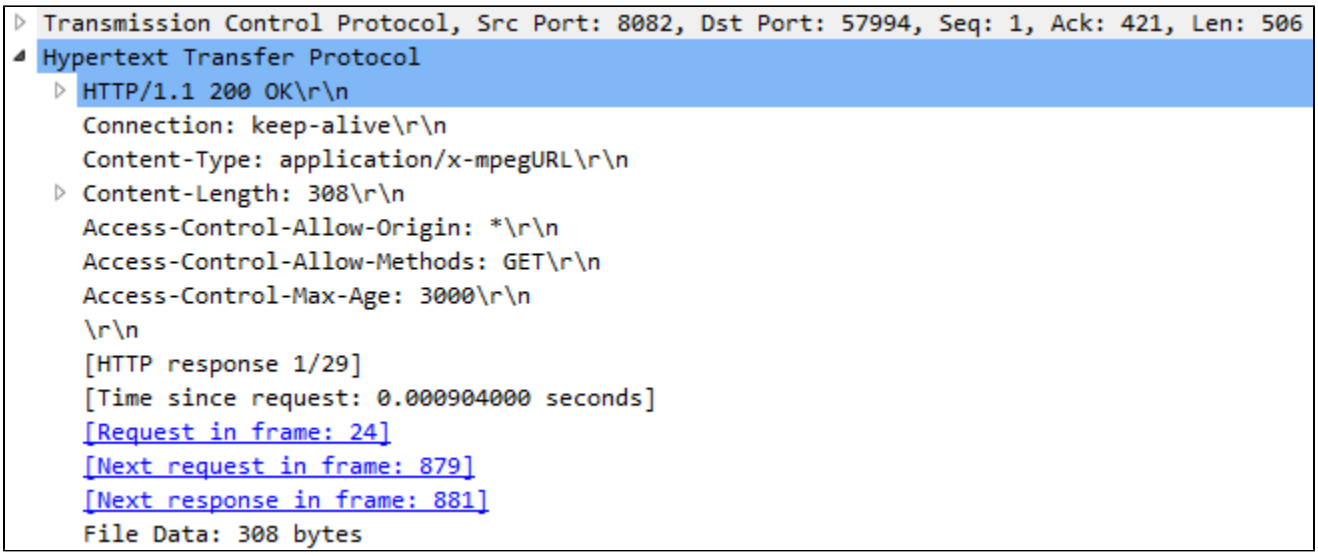
```
hls_auth_token_cache=0
```

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.

Добавление HTTP-заголовков для управления кросс-доменным воспроизведением HLS

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

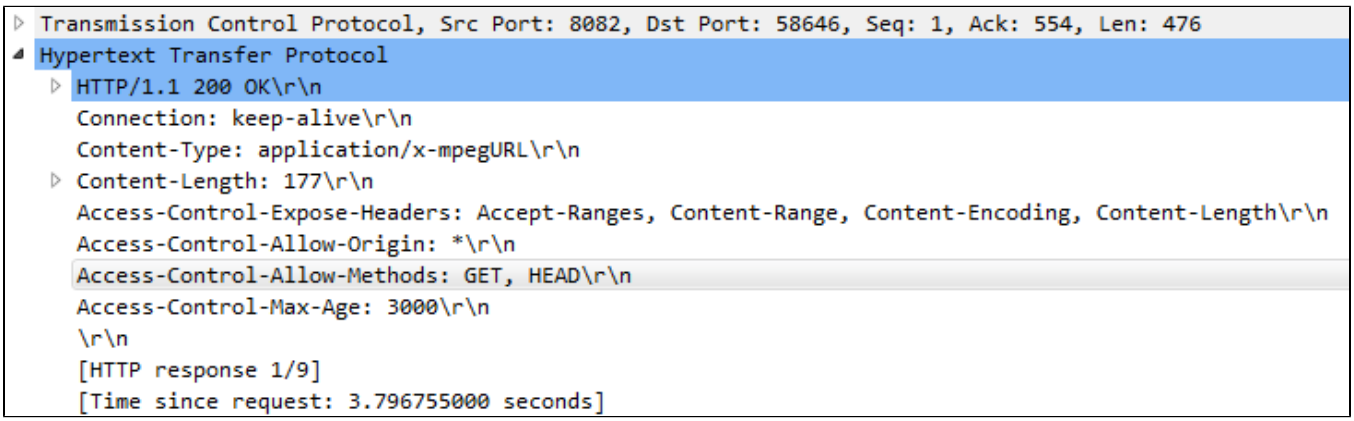



```
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET
Access-Control-Max-Age: 3000
```

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:



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.

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.