



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.773.5

РЕАЛИЗАЦИЯ ПРЯМОЙ ТРАНСЛЯЦИИ С ПОМОЩЬЮ NGINX-RTMP МОДУЛЯ

Микрюков И.И.

ФГАОУ ВО «Санкт-Петербургский государственный университет аэрокосмического приборостроения», Санкт-Петербург, Россия (190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А), e-mail: ilya.mikryukov.2001@mail.ru.

В данной статье рассматривается реализация прямой трансляции с использованием веб-сервера NGINX и его модуля NGINX-RTMP. Для работы сервера используется дистрибутив Linux – Ubuntu. Для запуска трансляции использовалась программа OBS, а воспроизведение происходило в плеере веб-браузера. Помимо описания работы прямых трансляций есть сравнение между собой некоторых протоколов для передачи видео. После выбора протоколов были проведены тесты с различными настройками и получено самое наименьшее время задержки.

Ключевые слова: Прямая трансляция, NGINX, NGINX-RTMP-module, HLS, DASH, протокол передачи видео.

IMPLEMENTING LIVE STREAMING USING THE NGINX-RTMP MODULE

Mikryukov I.I.

St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia (190000, St. Petersburg, Bolshaya Morskaya St., 67, letter A), e-mail: ilya.mikryukov.2001@mail.ru.

This article covers the implementation of live streaming using the NGINX web server and its NGINX-RTMP module. The server uses a Linux distribution - Ubuntu. The OBS program was used to start the broadcast, and the playback took place in the player of the web browser.

In addition to describing the operation of live broadcasts, there is a comparison between some protocols for transmitting video. After choosing the protocols, tests were carried out with various settings and the shortest delay time was obtained.

Keywords: Live streaming, NGINX, NGINX-RTMP-module, HLS, DASH, video transfer protocol.

Передача видео и звука по сети в реальном времени используется во многих приложениях и сайтах, которые работают как на стационарных компьютерах, так и на телефоне.

Существуют различные типы прямых трансляций, например, peer to peer подключения для общения между двумя людьми в мессенджере. В данном случае часто используется технология WebRTC (real-time communications — коммуникации в реальном времени). Такой тип подключения хорошо подходит для небольшой группы людей и не требует отдельно сервера для обработки видео. Также есть прямые трансляции, где один вещающий пользователь и много слушателей, подобные трансляции использует YouTube, Twitch.

Для создания сервиса с прямой трансляцией для очень большого количества людей уже необходимо использовать сервер, который будет обрабатывать получаемое видео и отправлять всем пользователям. В таком случае используются дополнительные сервисы,

которые могут это реализовать за определенную плату или пишутся свои решения, что довольно затратно.

В данной статье рассмотрена задача создания прямой трансляции, а также выбор технологий и инструментов, которые позволят это реализовать за короткий промежуток времени и будут бесплатными. После необходимо будет сделать тесты задержки видео на трансляции и подобрать подходящие настройки, чтобы как можно сильнее уменьшить задержку.

Рассмотрим прямую трансляцию, когда ведущий отправляет видео на сервер, сервер обрабатывает его и рассылает слушателям в браузер. В данном случае будет один источник видео и много слушателей, то есть это пример обычной прямой трансляции для большого количества людей. Дополнительную информацию можно найти на сайте [1]. Подробную информацию об установке сервера можно найти на сайте [2].

Перед тем как видео дойдёт до зрителя, ему необходимо пройти несколько этапов, рисунок 1:

- с помощью энкодера (в данном случае используется программа OBS (Open Broadcaster Software)) видео записывается, сжимается, преобразуется в заданный формат;
- данные отправляются на сервер для дальнейшей обработки по протоколу RTMP (Real Time Messaging Protocol);
- сервер, получив данные, может обработать их с целью, например, изменения. После сервер транскодирует видео в определенные форматы, которые могут воспроизводить зрители;

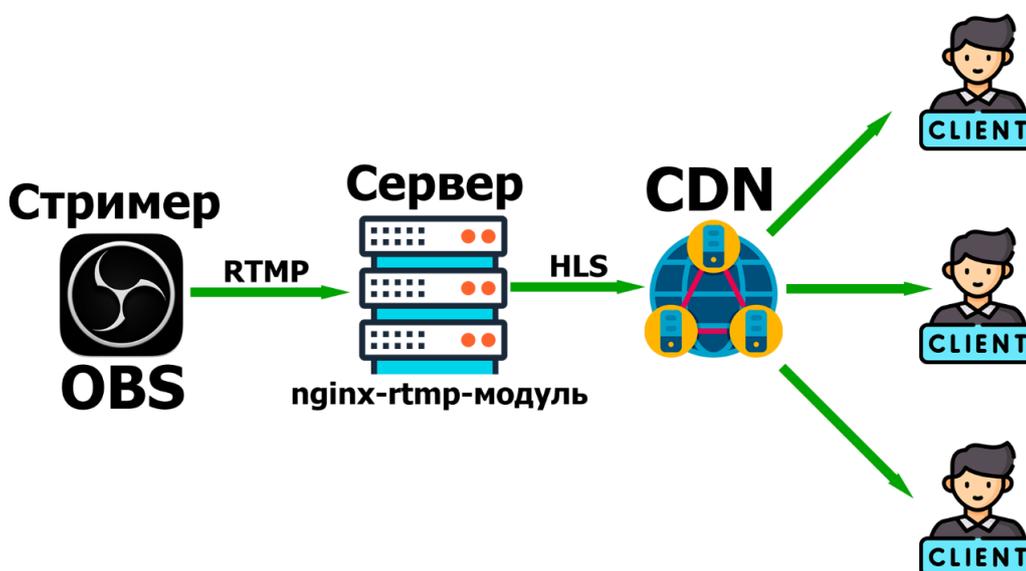


Рисунок 1 – Этапы доставки видео

- транскодированное видео передаётся по определенным протоколам (в данном случае используется протокол HLS (HTTP Live Streaming)) передачи видео зрителю. Часто используются CDN (Content Delivery Network), что представляет собой систему серверов, расположенных в разных физических местах.

Протоколы передачи видео в сети Интернет

Сначала рассмотрим протокол RTMP. Это протокол обмена информацией в реальном времени. Поддерживается на различных платформах и имеет совместимость с разными устройствами записи и серверами обработки видео.

Данные передаются с задержкой около 5 секунд, создаётся постоянный поток данных пользователям. Но этот протокол в данный момент уже мало где поддерживается в браузерах. Данные необходимо преобразовывать, чтобы их могло воспроизвести как можно большее число устройств, и передавать уже по другим протоколам.

Для решения этой задачи используются следующие протоколы: DASH (Dynamic Adaptive Streaming over HTTP) или HLS. В данном случае используется протокол HLS так как его было проще настроить. Этот протокол хорошо подходит для трансляции очень большому количеству людей. Его поддерживают много устройств и браузеров. Он поддерживает передачу видео с адаптивным битрейтом, то есть он будет изменяться в зависимости от скорости интернет-соединения.

Изначально задержка доставки видео могла достигать десятков секунд, что явно не подходит для прямых трансляций. Но в новой версии, которую поддерживают все браузеры и мобильные устройства, можно добиться времени задержки, не превышающего двух секунд.

В данном протоколе видео делится на фрагменты заданной длины. Также создаётся индексный файл, в котором хранится порядок фрагментов, он имеет заданную длину. Клиент скачивает с сервера индексный файл и доступные на данный момент фрагменты видео. Индексный файл содержит ссылки на несколько последних фрагментов. При повторном обращении к индексному файлу ссылки будут в нём обновляться на новые доступные фрагменты. Зная последовательность фрагментов из индексного файла и имея несколько последних, браузер (плеер) добавляет эти фрагменты в буфер воспроизведения в необходимом порядке. Аналогично работает доставка видео по протоколу DASH.

Измерение задержки на прямой трансляции

В данном случае используется Ubuntu server, на который установлен веб-сервер NGINX. Для обработки видеопотока используется NGINX-RTMP-модуль, который позволяет принимать по RTMP протоколу видео, обрабатывать его, и отправлять пользователям в браузер по протоколу HLS или DASH с различными настройками. Больше информации о возможных настройках и параметров можно узнать на сайтах [3] и [4]. Ниже показаны настройки сервера, а на рисунке 3 приведен программный код для работы плеера в браузере.

```
GNU nano 6.2 nginx.conf
worker_processes auto;
events {
    worker_connections 1024;
}

# RTMP configuration
rtmp {
    server {
        listen 1935; # Listen on standard RTMP port
        chunk_size 4096;

        application show {
            live on;
            # Turn on HLS
            hls on;
            hls_path /nginx/hls/;
            hls_fragment 2;
            hls_playlist_length 4;
        }
    }
}
```

Рисунок 2 – Настройки nginx-rtmp-модуля

```
function vvv()-{
    if (Hls.isSupported()){
        var video = document.getElementById('video');
        var hls = new Hls();
        hls.lowLatencyMode = true;
        hls.loadSource('http://192.168.0.171:8080/hls/test.m3u8');
        hls.attachMedia(video);
        hls.on(Hls.Events.MANIFEST_PARSED, function()-{ video.play(); });
    }
}
```

Рисунок 3 – Настройка плеера в браузере

При настройках, указанных на рисунке 2, задержка составляет 5,15 секунд, и трансляция работает стабильно. Настройки прямой трансляции вещающего в программе OBS:

- разрешение 1280x720,
- 30 кадров в секунду,
- битрейт 3000 Мбит/сек,
- ключевой кадр каждые 2 сек.

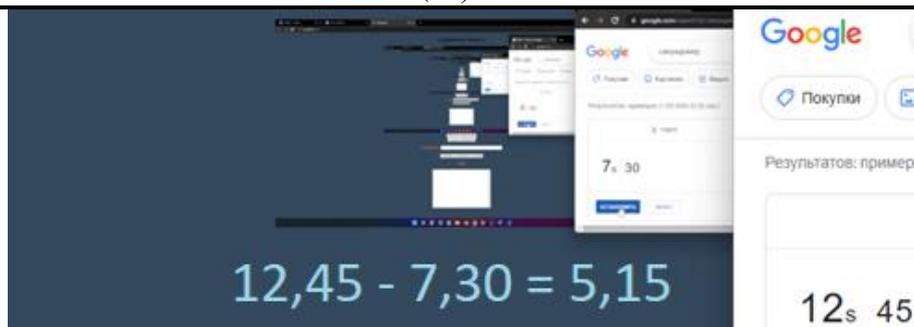


Рисунок 4 – Задержка 5,15 сек

При настройках (hls_fragment 1, hls_playlist_length 2), задержка составляет 3,54 секунд, но возможны появления “подгрузок” видео, которые скорее всего связаны с малой мощностью используемого сервера, в данном случае сервер был установлен на виртуальной машине. Настройки прямой трансляции вещающего в программе OBS:

- разрешение 1280x720,
- 30 кадров в секунду,
- битрейт 3000 Мбит/сек,
- ключевой кадр каждую 1 сек.

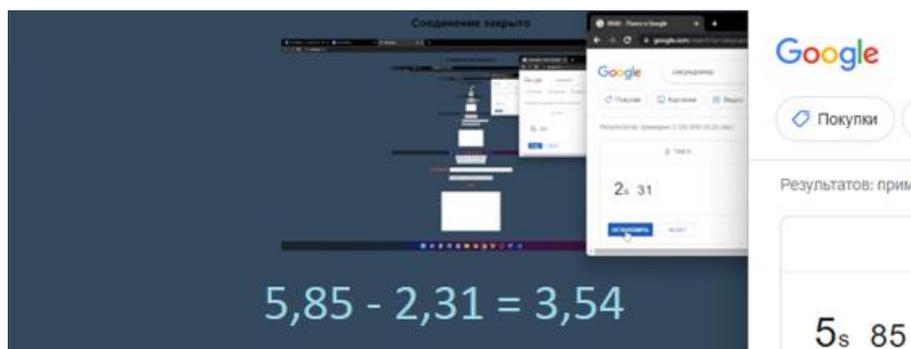


Рисунок 5 – Задержка 3,54 сек

С помощью каких команд можно обработать видео на сервере можно узнать на этом сайте [5]. В данном случае изменения видео на сервере не происходит и получаемое видео без изменений его параметров, например, битрейта будет отправляться зрителю, чтобы снизить время задержки на трансляции.

Заключение

В итоге мы смогли создать прямую трансляцию используя бесплатные инструменты. Провести тесты времени задержки и подобрать значения параметров на стороне сервера и вещающего для уменьшения задержки на трансляции.

Список литературы

1. Как делать прямые трансляции? Протоколы, кодирование, CDN. [Электронный ресурс] – Режим доступа: <https://stream-park.ru/blog/kak-delat-pryamyie-translyatsii-termyny-protokoly-kodirovanie-cdn/>

2. NGINX + rtmp. Трансляция видео с веб-сервера. Установка и настройка nginx-rtmp-module. [Электронный ресурс] – Режим доступа: <https://www.dmosk.ru/instruktions.php?object=nginx-rtmp>
3. arut/nginx-rtmp-module: NGINX-based Media Streaming Server. [Электронный ресурс] – Режим доступа: <https://github.com/arut/nginx-rtmp-module>
4. Directives · arut/nginx-rtmp-module Wiki. [Электронный ресурс] – Режим доступа: <https://github.com/arut/nginx-rtmp-module/wiki/Directives>
5. Онлайн вещание через Nginx-RTMP: несколько готовых рецептов / Хабр. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/articles/174089/>

References

1. How to make live broadcasts? Protocols, coding, CDN. [Electronic resource] – Access mode: <https://stream-park.ru/blog/kak-delat-pryamyie-translyatsii-terminy-protokoly-kodirovanie-cdn/>
 2. NGINX + rtmp. Broadcast video from a web server. Installing and configuring nginx-rtmp-module. [Electronic resource] – Access mode: <https://www.dmosk.ru/instruktions.php?object=nginx-rtmp>
 3. arut/nginx-rtmp-module: NGINX-based Media Streaming Server. [Electronic resource] – Access mode: <https://github.com/arut/nginx-rtmp-module>
 4. Directives arut/nginx-rtmp-module Wiki. [Electronic resource] – Access mode: <https://github.com/arut/nginx-rtmp-module/wiki/Directives>
 5. Online broadcasting via Nginx-RTMP: several ready-made recipes / Sudo Null IT News [Electronic resource] – Access mode: <https://habr.com/ru/articles/174089/>
-