Чернов Дмитрий Андреевич, 11И1

Никитин Александр Дмитриевич, 11И1

Предуниверситарий НИЯУ МИФИ, Москва, 2022

Разработка прототипа модульного интернет-радио

I. Введение

За последние несколько лет на различных вебсайтах получили широкое распространение[1] так называемые “интернет радио”. Это трансляции, не предназначены для прямого взаимодействия авторов с аудиторией – вместо этого, вся длительность эфира занята проигрыванием музыки какой-либо тематики. Это позволяет подключившимся пользователям слушать музыку без необходимости создания собственных плейлистов, откуда и произошло название.

Несмотря на популярность, разработчики стриминговых сервисов не предоставляют каких-либо дополнительных инструментов для организации и координирования работы таких трансляций. Этот проект призван решить данную проблему посредством создания приложения, позволяющего пользователям удобно создавать и настраивать в реальном времени интернет радио для нескольких стриминговых площадок одновременно.

II. Аналогичные проекты

На текущий момент существует несколько способов решения данной задачи, но все они не лишены своих недостатков. Популярный облачный сервис Restream заблокирован в России, а функционал для стриминга на собственные сайты доступен только в его платной версии[2]. С другой стороны, существует ПО OBS (использующее ffmpeg), но оно потребляет большое количество ресурсов компьютера, интерфейс не оптимизирован для управления интернет-радио, а также, как и в Restream, не поддерживаются некоторые протоколы передачи данных[3].

Все недостатки аналогичных сервисов обусловлены спецификой задачи, которая решается в данном проекте. Разрабатываемое приложение должно не только иметь пользовательский интерфейс управления трансляциями, но и поддерживать кодирование и передачу данных для любых возможных требований сервера назначения. Подробнее эти функции разобраны в следующих 2 параграфах.

III. Форматы файлов

Первое препятствие при создании сервиса, способного передавать мультимедиа файлы через интернет – их кодирование в определённом формате. Интеграция ПО FFmpeg, которое обладает всем необходимым функционалом[4], является наиболее оптимальным с точки зрения производительности[5], и, как было показано выше, проверенным методом решения данной проблемы.

При выборе форматов файлов, которые будет поддерживать приложение, важно ориентироваться не только на наиболее популярные варианты, но и учитывать особенности требований некоторых крупных сервисов. Исходя из спецификаций API созданных Google, а также нескольких независимых исследований, минимальный необходимый набор кодеков должен включать версии AAC, Vorbis и MP3 для аудио[6][7], а также FLV, AV1 и H.266/VVC для видео[8].

IV. Передача данных

Интеграция Ffmpeg является важным шагом в построении приложения, но кодирование данных не принесёт никакой пользы если связь с сервером назначения невозможно установить. Поэтому следующее препятствие, встречаемое в ходе разработки приложения – поддержка протоколов передачи данных. Возможность создания трансляций на любом из протоколов будет являться важным преимуществом данного проекта перед конкурентами.

Исследование документации аналогичных проектов, а также обзорных работ показывает, что в число необходимых протоколов входят RMTP[9], HLS[10] и DASH[6][11]. Но для того чтобы разрабатываемое приложение было совместимо с наибольшим количеством сервисов, важно поддерживать и менее популярные протоколы, такие как SRT[12] и RIST[13].

V. Заключение

Таким образом, довольно очевидным является факт того что интернет радио, несмотря на распространённость, ещё не полностью охвачена различными интернет-сервисами, и существует свободная и востребованная ниша для проекта, позволяющего администрировать подобные трансляции. Но для того чтобы быть полезным, а также положительно выделяться на фоне других аналогичных сервисов, в задачах проекта по разработке приложения должны содержаться внедрение современных технологий кодирования информации в множестве форматов с оптимальной производительностью, поддержка различных протоколов передачи данных и создание интуитивного интерфейса. На данный момент уже существуют все необходимые для этого инструменты, и остаётся только правильно использовать их для достижения поставленной цели.

VI. Ссылки

1. Andrew J. Bottomley, “[*Sound Streams: A Cultural History of Radio-Internet Convergence*](https://books.google.com/books?hl=ru&lr=&id=f9fkDwAAQBAJ&oi=fnd&pg=PR7&dq=internet+music+broadcasting&ots=VF6jTpwrDY&sig=RAXvI11GF2Ic7EoLFVbsRw33yN4)” (2020)

2. Halios00, *“*[*Restram.io – Multi-Streaming Within Your Grasp!*](https://streamersquare.com/restream-io-multi-streaming-within-grasp/)*”* (2020)

3. OBS Wiki, [*[1]*](https://obsproject.com/wiki/System-Requirements), [*[2]*](https://obsproject.com/wiki/Streaming-With-SRT-Or-RIST-Protocols)

4. The FFmpeg developers, *“*[*ffmpeg Documentation*](https://ffmpeg.org/ffmpeg.html)*”* (2022)

5. R. Safin, E. Garipova, R. Lavrenov, H. Li, M. Svinin, E. Magid, “[*Hardware and software video encoding comparison*](https://ieeexplore.ieee.org/abstract/document/9240439)*”* (2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2020)

6. Google LLC, *“*[*YouTube Livestreaming API Overview*](https://developers.google.com/youtube/v3/live/getting-started)*”* (2021)

7. Vladimir Britanak, K. R. Rao, “[*Audio Coding Standards, (Proprietary) Audio Compression Algorithms, and Broadcasting/Speech/Data Communication Codecs: Overview of Adopted Filter Banks*](https://link.springer.com/chapter/10.1007/978-3-319-61080-1_2)*”* (Cosine-Sine Modulated Filter Banks, 2018)

8. M. Saldanha et al., *“*[*An Overview of Dedicated Hardware Designs for State-of-the-Art AV1 and H.266/VVC Video Codecs*](https://ieeexplore.ieee.org/abstract/document/9294862)*”* (2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020)

9. K. Obraczka, *“*[*Multicast transport protocols: a survey and taxonomy*](https://ieeexplore.ieee.org/abstract/document/649333)*”* (IEEE Communications Magazine, vol. 36, no. 1, 1998)

10. A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, R. Zimmermann, “[*A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP*](https://ieeexplore.ieee.org/abstract/document/8424813)*”* (IEEE Communications Surveys & Tutorials, vol. 21, no. 1, 2019)

11. J. Kua, G. Armitage, P. Branch, *“*[*A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP*](https://ieeexplore.ieee.org/abstract/document/7884970)*”* (IEEE Communications Surveys & Tutorials, vol. 19, no. 3, 2019)

12. R. Viola et al., *“*[*Adaptive Rate Control for Live streaming using SRT protocol*](https://ieeexplore.ieee.org/abstract/document/9379708)*”* (2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2020)

13. M. Wånggren and L. Thyresson, *“*[*Live Cloud Ingest Using Open Alternatives RIST & SRT*](https://ieeexplore.ieee.org/abstract/document/9258429)*”* (SMPTE Motion Imaging Journal, vol. 129, no. 10, 2020)