

**Аміров Олександр Романович**

*Студент*

*Національного технічного університету України*

*«Київський політехнічний інститут імені Ігоря Сікорського»*

**Амиров Александр Романович**

*Студент*

*Национального технического университета Украины*

*«Киевский политехнический институт имени Игоря Сикорского»*

**Amirov O.**

*Student of the*

*National Technical University of Ukraine*

*«Igor Sikorsky Kyiv Polytechnic Institute»*

## СИСТЕМА ОНЛАЙН ВИДЕОТРАНСЛЯЦІЙ ЗА ДОПОМОГОЮ ВЕБ-СОКЕТІВ

## СИСТЕМА ОНЛАЙН ВИДЕОТРАНСЛЯЦІЙ С ПОМОЩЬЮ ВЕБ-СОКЕТОВ

## ONLINE VIDEOSTREAMING SYSTEM WITH WEB-SOCKETS

**Анотація.** У цій роботі досліджені шляхи побудови системи онлайн відеотрансляції з використанням мережевого протоколу веб-сокет.

**Ключові слова:** відеотрансляція, веб-сокет, джаваскрипт, медіа-інтерфейси браузера, веб-сокет сервер.

**Аннотация.** В данной работе исследованы пути построения системы онлайн видеотрансляции с использованием сетевого протокола веб-сокет.

**Ключевые слова:** видеотрансляция, веб-сокет, джаваскрипт, медиа-интерфейсы браузера, веб-сокет сервер.

**Summary.** This work deals with investigations of ways to design of online videostreaming system with usage of web protocol web-socket.

**Key words:** videostreaming, web-socket, javascript, browser media-interfaces, web-socket server.

Відеотрансляції стають в наш час все популярнішими, адже вони дозволяють отримувати досвід живої взаємодії, наприклад лектора зі студентами, на великих відстанях. Користувачі не мають бути обмеженими у виборі девайсів для використання відотрансляції, тому бажаним є рішення універсальне та незалежне від пристроїв чи додаткових програм.

При реалізації відеотрансляції в першу чергу постає задача вибору протоколу онлайн взаємодії, і типовий для веб HTTP протокол для вирішення такої проблеми є неефективним. Причина цього — в тому, що його архітектура взаємодії запит/відповідь не дає необхідної в такому випадку підтримки комунікації — двосторонньої швидкої передачі інформації між сервером та багатьма користувачами, яка також називається СОМЕТ [1].

Натомість протокол веб-сокет дає можливість встановити канал з'єднання між сервером і клієнтом і в подальшому обмінюватись даними в обох напрямках. Вебсокет використовує передачу даних фреймами, які є доволі економними в плані розміру порівняно з HTTP запитами. Це дає змогу легкої реалізації такого функціоналу, як чати, ігрові сервери тощо.

Також суттєвою проблемою є доступ до засобів запису аудіо та відео з пристрою користувача. Автор провів дослідження нових, навіть експериментальних, технологій сучасних браузерів і в результаті обрав mediaDevices API [3], оскільки цей інтерфейс дозволяє зробити запит на використання камери та мікрофону користувача і отримати аудіо чи відеопотік.

Для реалізації поставленої задачі необхідно вирішити низку проблем: розбиття потоку на частини

і підготовка до відправлення на сервер, транспортування частин потоку від клієнта до підписників трансляції, відтворення частин потоків, уникнення низької якості трансляції та оптимізація процесу.

Сучасне API браузерів налаштоване для відеодзвінків між двома браузерами. Поставлена ж задача потребує передачі даних через сервер великій кількості клієнтів. Тому для її реалізації вдалим рішенням було використати API виключно для того, щоб отримати потік даних, а далі налаштувати роботу вручну.

На самому початку процесу відотрансляції необхідно зробити запит до користувача на дозвіл використання камери на його девайсі за допомогою методу navigator.mediaDevices.getUserMedia [3]. Цей метод в разі згоди користувача повертає відеопотік зображення. При цьому роздільну здатність зображення можна налаштувати в параметрах методу.

Далі треба було б розбивати відеодані на частини та відправляти на сервер, але розмір таких частин досить великий для швидкої передачі через мережу. Тому було обрано інший шлях — відеодані розбиваються на звичайні знімки, що робляться з заданою частотою. Цього можна досягнути за допомогою використання елементу canvas, який дозволяє отримати знімок з відеопотоку та перетворити його у вигляд dataURL, що є придатним для пересилання мережею. Формат таких знімків jpeg обраний через його низький розмір, бо в цьому випадку можна знехтувати дуже високою якістю зображення. Такі знімки по одному пересилаються на сервер. Клієнт може отримати ці дані та відобразити як звичайне зображення img.

Трансляція аудіо не вимагає таких маніпуляцій, як трансляція зображення — формат ogg є досить компактним, щоб пересилати навіть великі відрізки запису. Спочатку іде запит на використання мікрофона користувача, і отримується потік аудіо. Потім

аудіо розбивається на частини за допомогою інтерфейсу mediaRecorder [2], що може записувати медіапотік частинами. Але на цьому етапі виникає проблема, бо в разі відправлення аудіо з такою ж частотою, як і зображення, відтворення матиме помітні дефекти. Це пояснюється тим, що якщо під час відтворення зображення з частотою 24 кадри на секунду людина не помічає зміни кадрів, то такі переривання між фрагментами аудіо роблять запис майже непридатним для сприйняття. Тому аудіо записується частинами фіксованої довжини в декілька секунд і перетворюється в бінарний тип Blob. Далі дані передаються клієнтам через сервер і програвуються за допомогою елементу audio.

Щоб синхронізувати зображення і аудіо, відправлення знімків відео можна відкласти на певний проміжок часу, що відповідає тривалості аудіочастин. Завдяки такій затримці ми отримуємо синхронізоване аудіо та відео досить високої якості.

Для роботи системи необхідно спроектувати і власне веб-сокет сервер. Це можна зробити за допомогою серверної платформи node. Її модуль socket.io [4] дозволяє легко налаштувати пересилання даних. В цьому випадку сервер має виконувати роль шини подій в патерні pubsub (підписник/видавець) і окремими каналами передавати аудіо- та відеодані між клієнтами.

### Висновки

Для того щоб ефективно реалізувати відеотрансляцію як на сервері, так і на клієнті, можна використати сучасні технології браузерів та протокол веб-сокет. Використання мови javascript допомагає зручному використанню таких ізоморфних модулів, як socket.io. Така система є адаптивною для різних пристроїв, оскільки вона реалізована як веб-застосунок, тобто використовуваність цієї системи обмежена тільки версією веб браузера.

### Література

1. Мельников А. The WebSocket Protocol: Стандарт [Електронний ресурс] / Мельников А., Фіт І. — 2011 — Режим доступу: <https://tools.ietf.org/html/rfc6455>
2. Кормік Л. MediaDevices: Стаття [Електронний ресурс] / Кормік Л. — 2016 — Режим доступу: <https://developer.mozilla.org/ru/docs/Web/API/MediaDevices>
3. Дуглас Д. Web Audio API: Стаття [Електронний ресурс] / Дуглас Д. — 2016 — Режим доступу: [https://developer.mozilla.org/ru/docs/Web/API/Web\\_Audio\\_API](https://developer.mozilla.org/ru/docs/Web/API/Web_Audio_API)
4. Ванг В. The Definitive Guide to HTML5 WebSocket / Ванг В., Салім Ф., Московіц П. — Apress, 2013. — 208 с.