

GENEROWANIE, EDYCJA I TRANSMISJA WIELOŹRÓDŁOWYCH STRUMIENI AUDIO Z WYKORZYSTANIEM WEB AUDIO API, WEBRTC I WEB MIDI API

Marcin WALCZAK¹

Ewa ŁUKASIK²

Instytut Informatyki, Politechnika Poznańska, ul. Piotrowo 2, 60-965 Poznań

¹kontakt@marcin-walczak.pl

²ewa.lukasik@cs.put.poznan.pl

Streszczenie. W czasach gdy konieczne jest stosowanie dystansu społecznego istnieje szczególnie zapotrzebowanie na niezawodne, gwarantujące bardzo małe opóźnienia systemy komunikacji audio/wideo online do przeprowadzania zdalnych prób, nagrywania i nadawania muzyki online. W artykule dokonano przeglądu systemów wykorzystywanych przez muzyków do zdalnej współpracy. Omówiono cechy technologii Web Audio API umożliwiającej syntezę, reprodukcję i przetwarzanie dźwięku w przeglądarce internetowej. W połączeniu z technologią WebRTC oraz Web MIDI API istnieje możliwość tworzenia zaawansowanych aplikacji webowych audio. W artykule opisano prototypowe wdrożenie i testy przykładowego systemu wykorzystującego Web Audio API, Web RTC i Web MIDI do tworzenia i przesyłania strumieni audio z wielu źródeł.

1. WPROWADZENIE

Komunikacja i wymiana treści multimedialnych za pośrednictwem Internetu stała się powszechna i ogólnie akceptowana, choć ostatnie miesiące koniecznej izolacji związanej z rozprzestrzenianiem się choroby COVID-19 sprawiło, że wykorzystanie środków komunikacji internetowej stało się wręcz nieodzowne we wszystkich dziedzinach życia. Bezpłatne komunikatory internetowe, np. Skype, Facebook Messenger, WhatsApp, Signal, Google Hangout stosowane są do komunikacji prywatnej, a dla celów zawodowych do listy tej dopisują się Zoom, Webex czy otwarte systemy oparte na BigBlueButton [2]. Istnieją również darmowe komunikatory głosowe dedykowane komunikującym się ze sobą uczestnikom internetowych gier komputerowych.

Internet nie został pierwotnie opracowany w celu przesyłania ruchu w czasie rzeczywistym. Utrata pakietów i opóźnienia wpływają negatywnie na jakość przesyłanego sygnału audio. Usługi wideokonferencji rozwiązują problemy związane z utratą pakietów, stosując długie ramki audio, duże bufony sieciowe i zasadę retransmisji pakietów. W konsekwencji powoduje to dodatkowe opóźnienia rzędu kilkuset milisekund. Takie opóźnienia nie stanowią problemu w kontekście komunikacji głosowej, jednak dla wymiany sygnałów muzycznych w czasie rzeczywistym jest to znaczne ograniczenie. Tymczasem muzycy jeszcze w czasach przed pandemią byli zainteresowani wspólnym muzykowaniem online i wspólnymi próbami. Dlatego od początku XXI-go wieku trwały prace nad realizacją sieci audio zoptymalizowanej pod kątem opóźnienia. Celem było umożliwienie zespołowego wykonywania muzyki przez artystów rozproszonych geograficznie za pośrednictwem sieci internetowej, tzw. Networked Music Performance (NMP)[9] z zapewnieniem wysokiej jakości interakcji. Problemy, jakie należało rozwiązać w celu realizacji NMP zostały przedstawione w rozprawie doktorskiej Ca-

rot'a [4]. W efekcie wieloletnich prac powstało kilka systemów wykorzystywanych przez muzyków do prowadzenia zdalnych prób i przygotowania nagrań za pomocą sieci. Są to zarówno systemy otwarte, jak np. Jamulus [5], SoundJack [11], LoLa [6] jak i komercyjne, np. Artsmesh [1].

Mimo wspomnianych wyżej zaawansowanych rozwiązań, przetwarzanie audio nie było przez długi czas wspierane w przeglądarkach internetowych. Wprowadzenie elementu `<audio>` w HTML5 umożliwiło strumieniowe odtwarzanie dźwięku, ale nie pozwalało na budowę bardziej skomplikowanych aplikacji audio, ani na obsługę audio w zaawansowanych grach internetowych i aplikacjach interaktywnych. Nie zapewniało miksowania, przetwarzania i filtrowania na równi z nowoczesnymi rozwiązaniami dostępnymi w komputerach stacjonarnych. Takie możliwości zapewnia Web Audio API [12], którego pierwsza wersja pojawiła się w 2011 roku i nad którym prace trwają do dnia dzisiejszego. Podstawowym paradygmatem dla tej technologii jest graf routingu audio, w którym wiele obiektów *AudioNode* jest połączonych ze sobą, aby zapewnić renderowanie dźwięku. Web Audio API jest wspomagany przez szereg bibliotek JavaScript ułatwiających realizację bardziej skomplikowanych zadań, np. Tone.js.

2. OPIS TECHNOLOGII

Podstawowe funkcje, jakie spełnia WebAudio API są następujące [12]: modularny routing dla prostych lub złożonych architektur i efektów dźwiękowych, miksowanie i efekty dźwiękowe, 32-bitowe przetwarzanie zmiennoprzecinkowe zapewniające wysoki zakres dynamiki dźwięku, odtwarzanie dźwięku z niskim opóźnieniem do zastosowań muzycznych wymagających bardzo wysokiego stopnia precyzji rytmicznej, takich jak automaty perkusyjne i sekwencery, elastyczna obsługa kanałów w strumieniu audio, umożliwiająca dzielenie i łączenie strumieni, przetwarzanie źródeł dźwięku z obiektu multimedialnego `<audio>` lub `<video>`, przetwarzanie strumienia audio generowanego na żywo, przetwarzanie dźwięku odebranego od zdalnego użytkownika i wysyłanie przetworzonego strumienia, a także obsługę dźwięku przestrzennego, efektywna analiza muzyki w dziedzinie czasu i częstotliwości w czasie rzeczywistym, wydajna filtracja dolnoprzepustowa, górnoprzepustowa i za pomocą innych popularnych filtrów, kształtowanie fali i inne efekty nieliniowe.

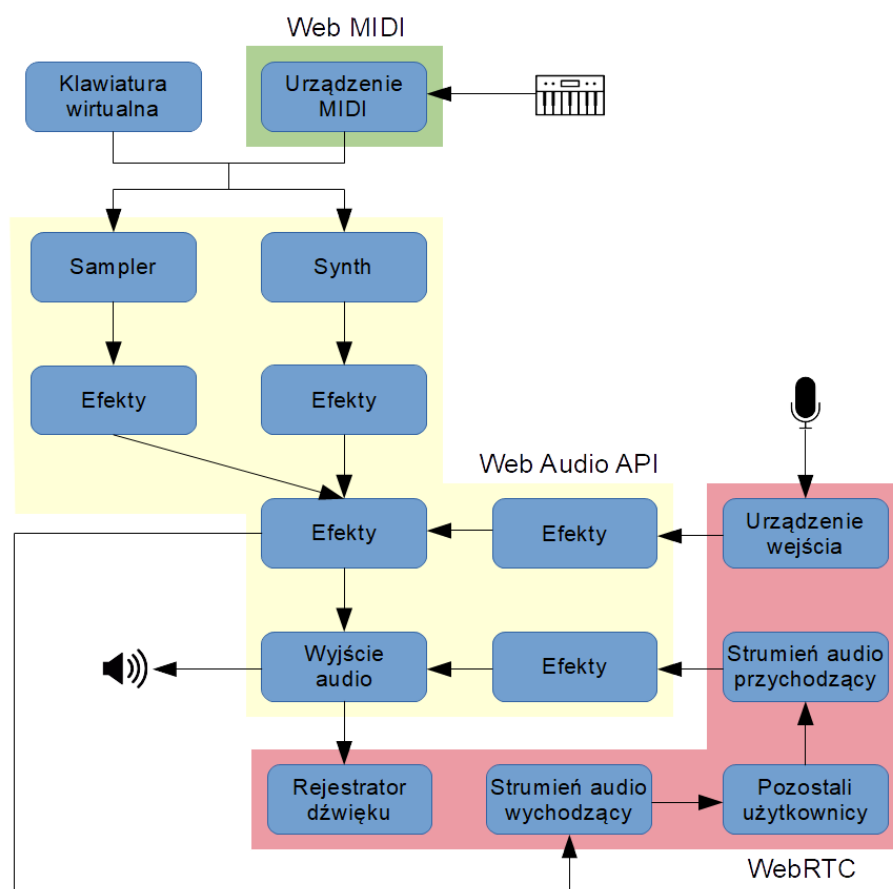
Web Audio API umożliwia również integrację z WebRTC [7]. WebRTC (ang. *Web Real-Time Communication*) jest technologią umożliwiającą aplikacjom internetowym przechwytywanie i przesyłanie strumienia audio i wideo oraz innych danych bezpośrednio między urządzeniami metodą *peer-to-peer*.

WebRTC obejmuje trzy zasadnicze API [10]:

- `MediaStream API` zapewniający dostęp do strumieni danych audiowizualnych z wielu źródeł, jak np. mikrofony, i kamery,
- `RTCPeerConnection` do tworzenia i zarządzania połączeniami *peer-to-peer*,
- `RTCDataChannel` obsługujący komunikację *peer-to-peer* danych.

Architektura WebRTC jest oparta na trapezoidzie SIP (ang. *Session Initiation Protocol*) [7]. W modelu trapezoidalnym WebRTC dwie komunikujące się przeglądarki uruchamiają aplikacje sieciowe, pobrane z osobnych serwerów. Za pośrednictwem serwerów sygnalizacyjnych, wstępnie wymieniają się danymi w celu wynegocjowania połączenia *peer-to-peer*. Dane sygnalizacyjne mogą być przesyłane dowolną metodą, jak np. protokołem HTTP lub WebSocket. Po nawiązaniu połączenia, dalsza komunikacja pomiędzy przeglądarkami następuje według modelu *peer-to-peer*, bez udziału serwerów.

Protokół MIDI (ang. *Musical Interface for Digital Instruments*) jest powszechnie stosowanym w przemyśle muzycznym protokołem umożliwiającym komunikację pomiędzy kompatybilnymi urządzeniami muzycznymi. Celem Web MIDI API [13] jest komunikacja pomiędzy przeglądarką, a urządzeniami MIDI, zarówno wejścia jak i wyjścia.



Rys.1. Przykładowy schemat przepływu strumieni dźwiękowych wewnątrz aplikacji wraz z oznaczeniem elementów wykorzystujących technologie Web Audio API, WebRTC i Web MIDI. Opracowanie własne

3. REALIZACJA SYSTEMU

3.1 SZCZEGÓŁY IMPLEMENTACYJNE

Celem opisanego w artykule projektu była budowa prototypowego systemu umożliwiającego generowanie i modyfikację przez użytkownika ścieżek dźwiękowych z wykorzystaniem przeglądarki internetowej zarówno strumieni wejściowych, jak i wyjściowych. Strumienie dźwiękowe podlegają konfiguracji oraz modyfikacji w czasie rzeczywistym poprzez filtry i inne elementy zmieniające strumień dźwiękowy. Możliwa jest również rejestracja sygnałów dźwiękowych pochodzących z urządzeń wejścia (jak np. mikrofon) i generowanych przez aplikację strumieni dźwiękowych. Zarejestrowane w ten sposób ścieżki dźwiękowe można odtworzyć i wykorzystać w aplikacji. Dodatkowo dostępna jest funkcja eksportu zarejestrowanej ścieżki dźwiękowej w formie wybranego przez użytkownika formatu audio. Sterowanie warstwą dźwiękową odbywa się za pomocą wirtualnej klawiatury dostępnej w oknie przeglądarki, fizycznej klawiatury urządzenia oraz urządzenia MIDI. Poza warstwą dźwiękową aplikacja umożliwia komunikację sieciową pomiędzy użytkownikami. Użytkownik ma możliwość tworzenia i zarządzania wirtualnymi pokojami pozwalającymi innym użytkownikom na dołączenie do nich. Osoby korzystające z tego samego pokoju mogą komunikować się między sobą poprzez połączenia *peer-to-peer* [8]. Schemat przepływu strumieni dźwiękowych wewnątrz aplikacji wraz

z oznaczeniem elementów wykorzystujących technologie Web Audio API [12], WebRTC [10] i Web MIDI API [13] przedstawiono na rysunku 1.

Podstawowym zadaniem serwera jest pośrednictwo przy komunikacji między klientami aplikacji. Nawiązanie połączenia *peer-to-peer* za pomocą technologii WebRTC wymaga kanału komunikacyjnego do negocjacji właściwego połączenia. Rolę tę przejmuje przygotowany w tym celu serwer.

Zaimplementowano i skonfigurowano serwer STUN w celu pozyskania przez aplikacje klienckie swoich globalnych adresów IP. Umożliwiło to poprawne ustanawianie bezpośrednich połączeń między użytkownikami w przypadku, gdy co najmniej jeden z nich znajduje się wewnątrz sieci NAT.

Zgodnie z przedstawionymi wymaganiami aplikacja powinna pozwalać na uzyskanie połączenia *peer-to-peer* pomiędzy użytkownikami przy różnych konfiguracjach sieciowych. Ustanawianie połączeń pomiędzy użytkownikami może zostać utrudnione m.in. z uwagi na zastosowanie przez sieć lokalną klienta NAT lub ograniczenia w ustanawianiu połączeń nałożone przez lokalną zaporę sieciową.

W przypadku, gdy nie jest możliwe nawiązanie połączenia *peer-to-peer* z uwagi na zabezpieczenia sieci, zastosowano serwer TURN. Został on wdrożony jako zastępcza forma komunikacji w sytuacji blokowania przez sieć połączenia *peer-to-peer* z innym klientem aplikacji. Działa on wtedy jako pośrednik, przesyłając dane pomiędzy użytkownikami. Z uwagi na przechodzenie strumieni danych przez serwer TURN zamiast bezpośrednio poprzez połączenie *peer-to-peer*, opóźnienia przy wykorzystaniu tego rodzaju połączenia mogą ulec zwiększeniu.

3.1 TESTY

Przeprowadzono testy aplikacji metodą białej skrzynki. Skupiono się w nich na sprawdzeniu działania poszczególnych elementów aplikacji na wybranych przeglądarkach wspierających Web Audio API i WebRTC.

Testy przeprowadzono na podanych poniżej przeglądarkach:

- Google Chrome w wersji 79.0.3945.88,
- Mozilla Firefox w wersji 71.0,
- Opera w wersji 65.0.3467.69,
- Safari w wersji 13.0,
- Microsoft Edge w wersji 44.18362.449.0.

Przetestowano różne aspekty komunikacji między użytkownikami, generowanie dźwięku z syntezy, obsługę urządzeń wejścia: mikrofonów i urządzeń MIDI, przetwarzanie dźwięku: dodawanie i edycję efektów w strumieniach dźwięku oraz rejestrację i eksport ścieżki dźwiękowej. Wszystkie rezultaty były zgodne z założeniami.

Natomiast przeprowadzone testy aplikacji w sieci wykazały, że w przypadku części sieci nie było możliwe nawiązanie połączenia *peer-to-peer* wyłącznie za pomocą serwera sygnalizacyjnego. Przypadki niemożności ustanowienia połączenia między użytkownikami były spowodowane brakiem dostępu aplikacji klienckich do ich globalnych adresów IP lub blokowanie przez zaporę sieciową połączeń *peer-to-peer*. Powodowało to oczywiście generowanie dodatkowych opóźnień.

PODSUMOWANIE

W artykule została opisana aplikacja internetowa demonstrująca możliwości wykorzystania technologii Web Audio API, WebRTC API oraz Web MIDI. Aplikacja pozwala na ustanawianie połączeń *peer-to-peer* pomiędzy wieloma użytkownikami za pomocą technologii WebRTC. Wdrożono możliwość generowania, odtwarzania, modyfikacji, przechwytywania oraz przesyłania ścieżek dźwiękowych drogą sieciową. Mimo że aplikacja została opracowana jako demonstracja możliwości opisanych technologii, posiada potencjał rozwojowy. W oparciu o już zaimplementowane technologie możliwe

jest wdrożenie nowych funkcji z zakresu generowania i przetwarzania dźwięku, komunikacji między użytkownikami oraz rozwinięcie już istniejących. Aplikacja może zostać rozbudowana o m.in. sekwencjonowanie ścieżek dźwiękowych, graficzną reprezentację cech strumieni dźwięku w czasie rzeczywistym, czy transmisję strumieni dźwięku między wieloma użytkownikami za pomocą serwera mediów. Wtedy należy przeprowadzić badania jakości transmisji, w tym opóźnień systemowych.

Doświadczenie zdobyte przez jednego z autorów przy realizacji przedstawionego projektu stało się podstawą do podjęcia prac w ramach komercyjnego międzynarodowego projektu Branch [3] udostępniającego wirtualną przestrzeń biurową. System pozwala na komunikację audiowizualną między użytkownikami z wykorzystaniem technologii WebRTC [10]. Jedną z cech wyróżniających projekt jest sposób, w jaki użytkownicy komunikują się między sobą. Aplikacja pozwala na dynamiczne nawiązywanie dwustronnych połączeń w topologii siatki między użytkownikami znajdującymi się w swoim "zasięgu wzroku". W przypadku utraty użytkownika z pola widzenia, połączenie między nimi zostaje zamknięte. Takie rozwiązanie umożliwia emulację rzeczywistej interakcji między członkami zespołu pozwalając na dynamiczne tworzenie grupowych spotkań w sposób odpowiadający temu w rzeczywistym życiu. Dodatkową zaletą jest ograniczenie liczby aktywnych połączeń WebRTC [10], co pozwala redukować generowane przez nie obciążenie łącza.

Omówione technologie nie są jeszcze w pełni wspierane przez wszystkie nowoczesne przeglądarki internetowe. Ich standardy i rozwiązania wciąż podlegają rozwojowi i mogą ulec zmianie w przyszłości.

LITERATURA

- [1] ARTSMESH <https://www.artsmesh.com/>,
- [2] BIGBLUEBUTTON <https://bigbluebutton.org>
- [3] BRANCH <https://branch.gg/>
- [4] CARÔT A., Musical Telepresence - A Comprehensive Analysis Towards New Cognitive and Technical Approaches, PhD Thesis, Lübeck, 2009.
- [5] JAMULUS <http://lcon.sourceforge.net/>,
- [6] LOLA <https://lola.conts.it/>
- [7] LORETO S.; ROMANO S.P.: Real-Time Communication with WebRTC: Peer-to-Peer in the Browser. O'Reilly Media, Inc. Sebastopol, 2014
- [8] NAYYEF Z., AMER S., HUSSAIN Z., Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology, Int. J. for the History of Engineering & Technology, 2019, s. 125 – 130
- [9] ROTTONDI C, CHAFE Ch., ALLOCCHIO C., SARTI A.: An Overview on Networked Music Performance Technologies. IEEE Access 4: 8823-8843 2016,
- [10] WEB RTC, Real Time Communication for the Web, webrtc.org
- [11] SOUNDJACK <https://www.soundjack.eu/>,
- [12] Web Audio API, W3C Candidate Recommendation, 11 June 2020, <https://www.w3.org/TR/webaudio/>
- [13] Web MIDI API, W3C Working Draft, March 2015, <https://www.w3.org/TR/2015/WD-webmidi-20150317>

GENERATING, EDITING AND MULTI-SOURCE TRANSMISSION OF AUDIO STREAMS USING WEB AUDIO API, WEBRTC AND WEB MIDI API

In the time of social distancing, there is a special need for reliable low-latency audio/video online communication systems for online music rehearsing, recording and broadcasting. In the paper, first, a review of existing systems used by remote users will be provided. Then the potential of the Web Audio API will be discussed. The Web Audio API introduces extensive functionality related to the synthesis, reproduction and processing of sound. In combination with the WebRTC technology enabling real-time audio streaming and capturing, and with the Web MIDI API technology ensuring communication with MIDI devices, there is a possibility of creating advanced Web audio applications. We will describe prototype implementation and tests of an exemplary system using the above mentioned technologies to form and transmit audio streams from multiple sources.