

Kierunek: **Informatyka techniczna**
Specjalność: **Grafika i Systemy Multimedialne**

PRACA DYPLOMOWA
INŻYNIERSKA

**Wideokomunikator internetowy
implementujący nowoczesne kodeki
video w języku Rust**

**Internet videochat application
implementing modern video codecs
using Rust programming language**

Marcel Guzik

Opiekun pracy
dr inż. Marek Woda

Opracował: Tomasz Kubik <tomasz.kubik@pwr.edu.pl>
Data: grudzień 2021



Tekst zawarty w niniejszym szablonie jest udostępniany na licencji Creative Commons: *Uznanie autorstwa – Użycie niekomercyjne – Na tych samych warunkach, 3.0 Polska*, Wrocław 2021.

Oznacza to, że wszystkie przekazane treści można kopiować i wykorzystywać do celów niekomercyjnych, a także tworzyć na ich podstawie utwory zależne pod warunkiem podania autora i nazwy licencjodawcy oraz udzielania na utwory zależne takiej samej licencji. Tekst licencji jest dostępny pod adresem: <http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>.

Podczas redakcji pracy dyplomowej stronę tę można usunąć. Licencja dotyczy bowiem zredagowanego opisu, a nie samego latexowego szablonu. Latexowy szablon można wykorzystywać bez wzmiankowania o jego autorze.

Streszczenie

Streszczenie w języku polskim powinno zmieścić się na połowie strony (drugą połowę powinien zająć abstract w języku angielskim).

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

Słowa kluczowe: raz, dwa, trzy, cztery

Abstract

Streszczenie in Polish should fit on the half of the page (the other half should be covered by the abstract in English).

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

Keywords: one, two, three, four

Spis treści

1. Wstęp	9
1.1. Wprowadzenie	9
2. Internetowe strumienie wideo	11
3. WebRTC	12
3.1. Co to jest, z czego się składa?	12
3.2. Sygnalizacja nawiązania połączenia	12
3.3. Interactive Connectivity Establishment (ICE)	12
3.4. Opis API WebRTC	12
4. Aplikacja webowa z użyciem WebRTC	13
4.1. Architektura, budowa aplikacji, omówienie kodu	13
4.2. Omówienie działania aplikacji na przykładzie	13
4.2.1. Analiza ruchu sieciowego WebRTC	13
4.2.2. Analiza pakietów protokołu SDP	13
5. Mechanizmy kompresji wideo	14
5.1. Koncepty kompresji	14
5.2. Kontenery	14
5.3. Kodeki	14
5.3.1. H.264	14
5.3.2. H.265	14
5.3.3. VP9	14
5.3.4. AV1	14
6. Architektura projektu Piperchat	15
6.1. Serwer	15
6.2. Klient	15
7. Aplikacja desktopowa w języku Rust	16
7.1. GUI framework	16
7.2. ffmpeg	16
Bibliografia	17
8. Instrukcja wdrożeniowa	18
9. Opis załączonej płyty CD/DVD	19

Spis rysunków

3.1. Stos protokołów w WebRTC - wzięty z HPBN - do zmiany	12
---	----

Spis tabel

Spis listingów

Skróty

OGC (ang. *Open Geospatial Consortium*)
XML (ang. *eXtensible Markup Language*)
SOAP (ang. *Simple Object Access Protocol*)
WSDL (ang. *Web Services Description Language*)
UDDI (ang. *Universal Description Discovery and Integration*)
GIS (ang. *Geographical Information System*)
SDI (ang. *Spatial Data Infrastructure*)
ISO (ang. *International Standards Organization*)
WMS (ang. *Web Map Service*)
WFS (ang. *Web Feature Service*)
WPS (ang. *Web Processing Service*)
GML (ang. *Geography Markup Language*)
SRG (ang. *Seeded Region Growing*)
SOA (ang. *Service Oriented Architecture*)
IT (ang. *Information Technology*)

Rozdział 1

Wstęp

1.1. Wprowadzenie

Pandemia COVID-19 oraz zdobywające coraz większą popularność zdalne formy zatrudnienia obrazują jak ważna jest rola połączeń wideo we współczesnym społeczeństwie. W wielu przypadkach kontakt "twarzą w twarz" jest preferowalny, a nawet niezbędny do realizacji pewnych zadań. W takich wypadkach niezawodność oraz efektywność transmisji wideo stają się bardzo ważnymi problemami.

Celem niniejszej pracy jest analiza połączeń wideo czasu rzeczywistego w każdym ich etapie, badanie procesów składających się na nie, i wreszcie utworzenie internetowego komunikatora wideo wykorzystującego poznane koncepty i rozwiązania. Tworzony komunikator będzie aplikacją okienkową na systemy Linux, będzie nawiązywał połączenia wideo peer-to-peer, a także będzie wykorzystywał kodek AV1 do kompresji wideo.

W rozdziale 2 omówione zostaną koncepty i metody związane ze strumieniowaniem wideo. Przedstawiony zostanie uproszczony opis procesu transmisji wideo w czasie rzeczywistym, od pobrania klatki obrazu przez kamerę internetową, do wyświetlenia tejże klatki na monitorze rozmówcy. Poruszone zostaną problemy związane z transmisją wideo przez sieć internetową, np. problem ustanowienia kanału P2P pomiędzy klientami za siecią NAT, negocjowanie sesji pomiędzy klientami, adaptive bitrate, etc.

W rozdziale 3 omówiony zostanie projekt WebRTC i jego protokoły składowe, czytelnik dowie się również w jaki sposób WebRTC rozwiązuje problemy poruszone we wcześniejszym rozdziale i tym samym niezwykle upraszcza tworzenie multimedialnych aplikacji webowych.

W rozdziale 4 zaprezentowana zostanie aplikacja webowa wykorzystująca WebRTC, nastąpi ogólny przegląd wykorzystywanych technologii oraz oprogramowania, zaprezentowane zostaną fragmenty kodu źródłowego realizujące kluczowe procesy nawiązywania połączenia poruszone we wcześniejszym rozdziale. Wykonana zostanie dokładna analiza ruchu sieciowego pomiędzy hostami i uzupełniony zostanie proces nawiązywania połączenia WebRTC poruszony we wcześniejszym rozdziale, zobrazowany konkretnym przykładem.

W rozdziale 5 omówione zostaną problemy, generalne mechanizmy i koncepty związane z kompresją wideo, kontenery, strumienie oraz ich muxowanie do kontenerów, rodzaje kodeków wideo, proces enkodowania/dekodowania, sprzętowe oraz programowe implementacje koderów.

W rozdziale 6 poruszone zostaną problemy i rozwiązania związane ze strumieniowaniem AV1. Zrealizowane zostanie "zejście na niższy poziom", co pozwoli na głębsze zapoznanie się z krokami realizowanymi celem przygotowania strumienia wideo do transmisji oraz wyświetlenia przychodzącego wideo na ekranie monitora. Zrealizowane zostanie min.: zastąpienie całości lub części stosu WebRTC rozwiązaniami natywnymi, oferowanymi przez sprzęt lub system operacyjny, wyeliminowanie abstrakcji oferowanych przez przeglądarkę, wybór optymalnego koder

oraz jego parametrów, wybór technologii GUI, zaplanowanie i omówienie procesu strumieniowania z rozdziału 1szego, wraz z elementami które będą go realizowały.

W rozdziale 7 zaprezentowana zostanie wykonana aplikacja desktopowa wykorzystująca biblioteki oraz inne rozwiązania udostępniane przez system operacyjny. Omówienie architektury, wykorzystywanych technologii i fragmentów kodu aplikacji okienkowej. Zaprezentowanie wybranych etapów procesu strumieniowania realizowanych przez aplikację.

Rozdział 2

Internetowe strumienie wideo

Jakieś wprowadzenie o wideo ogólnie, w jakich kontekstach występuje (pliki wideo, streaming np. Youtube/Netflix, strumień czasu rzeczywistego np. Twitch/Discord/Teams), czym te konteksty się różnią.

Zidealizowany obraz rozmowy wideo w internecie:

1. Komputery są publicznymi hostami w internecie i program komunikatora słucha na danym porcie
2. Strona nawiązująca połączenie łączy się do hosta odbiorcy po tym porcie, sygnalizuje chęć nawiązania połączenia
3. Strona odbierająca akceptuje
4. Kamera nadawcy przechwytuje najlepszy obraz jaki może, klatka po klatce i przesyła go do komputera
5. Komputer wysyła przechwycone klatki wcześniej ustanowionym kanałem

Natomiast pojawiają się problemy:

1. Komputery znajdują się w sieciach domowych, za NATem, nie można się do nich bezpośrednio połączyć
2. Nieskompresowane wideo jest zbyt duże by wysłać je przez internet, potrzebny jest jakiś mechanizm kompresji
3. Komputery mogą mieć różne możliwości przetwarzania wideo: znajdować się w sieciach o znacząco różnej szybkości, mieć różniące się szybkością procesory, kamery zapisujące klatki w różnych formatach

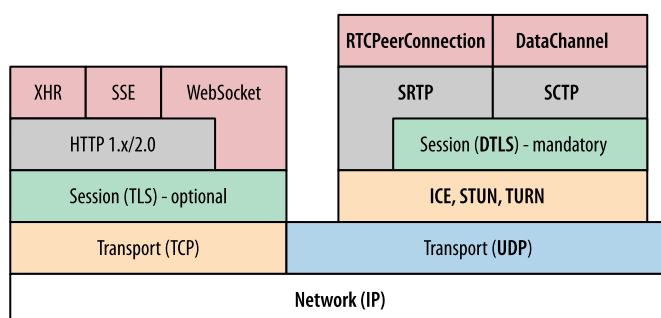
Rozdział 3

WebRTC

3.1. Co to jest, z czego się składa?

Beniz [1] eniz

WebRTC jest projektem zapewniającym przeglądarkom możliwości komunikacji czasu rzeczywistego, dzięki czemu możliwe jest budowanie komunikatorów internetowych w całości w obrębie zapewnianego przez przeglądarkę javascriptowego API. Aby funkcjonalność ta była możliwa do zrealizowania, niezbędne było użycie wielu protokołów sieciowych, które pokazano na rysunku 3.1.



Rys. 3.1: Stos protokołów w WebRTC - wzięty z HPBN - do zmiany

3.2. Sygnalizacja nawiązania połączenia

3.3. Interactive Connectivity Establishment (ICE)

3.4. Opis API WebRTC

Rozdział 4

Aplikacja webowa z użyciem WebRTC

4.1. Architektura, budowa aplikacji, omówienie kodu

4.2. Omówienie działania aplikacji na przykładzie

4.2.1. Analiza ruchu sieciowego WebRTC

4.2.2. Analiza pakietów protokołu SDP

Rozdział 5

Mechanizmy kompresji wideo

5.1. Koncepty kompresji

5.2. Kontenery

5.3. Kodeki

5.3.1. H.264

5.3.2. H.265

5.3.3. VP9

5.3.4. AV1

Rozdział 6

Architektura projektu Piperchat

6.1. Serwer

6.2. Klient

Rozdział 7

Aplikacja desktopowa w języku Rust

7.1. GUI framework

7.2. ffmpeg

Bibliografia

- [1] I. Grigorik, “High performance browser networking”, en, w Sebastopol, CA: O’Reilly Media, wrz. 2013, rozdz. WebRTC.

Rozdział 8

Instrukcja wdrożeniowa

Jeśli praca skończyła się wykonaniem jakiegoś oprogramowania, to w dodatku powinna pojawić się instrukcja wdrożeniowa (o tym jak skompilować/zainstalować to oprogramowanie). Przydałoby się również krótkie how to (jak uruchomić system i coś w nim zrobić – zademonstrowane na jakimś najprostszym przypadku użycia). Można z tego zrobić osobny dodatek,

Rozdział 9

Opis załączonej płyty CD/DVD

Tutaj jest miejsce na zamieszczenie opisu zawartości załączonej płyty. Należy wymienić, co zawiera.