

AGH SSM WebRTC

laboratorium 2



Pytania

- Czy WebRTC się skaluje?
- Jak tworzyć systemy wideokonferencyjne?
- Jak rozwiązać problem różnych warunków sieciowych u różnych odbiorców?
- Jak określić przepustowość sieci między dwoma hostami?
- Co i jak można zoptymalizować?



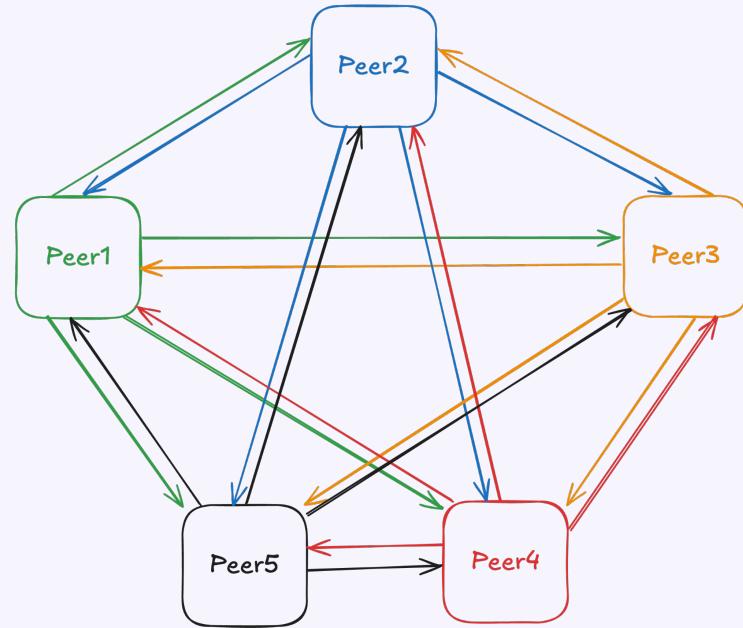
Zadania

1. Wprowadzenie do simulcastu.
2. Wpływ packet lossu na jakość transmisji.
3. Implementacja strony klienckiej prostego systemu wideokonferencyjnego.



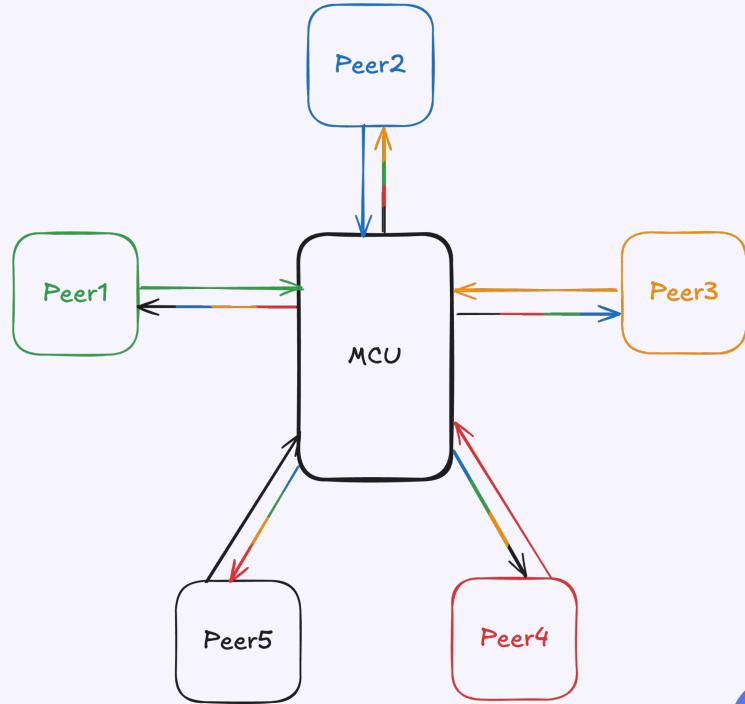
Mesh

- Architektura połączeń “każdy z każdym”
- Duża liczba połączeń:
 - w sumie $n * (n-1) / 2$
- Duże zużycie pasma:
 - Upload i download wymagany od każdego uczestnika rosną liniowo z ich liczbą
- Tania w utrzymaniu
 - koszty przerzucone na userów, wystarczy serwer do signalingu



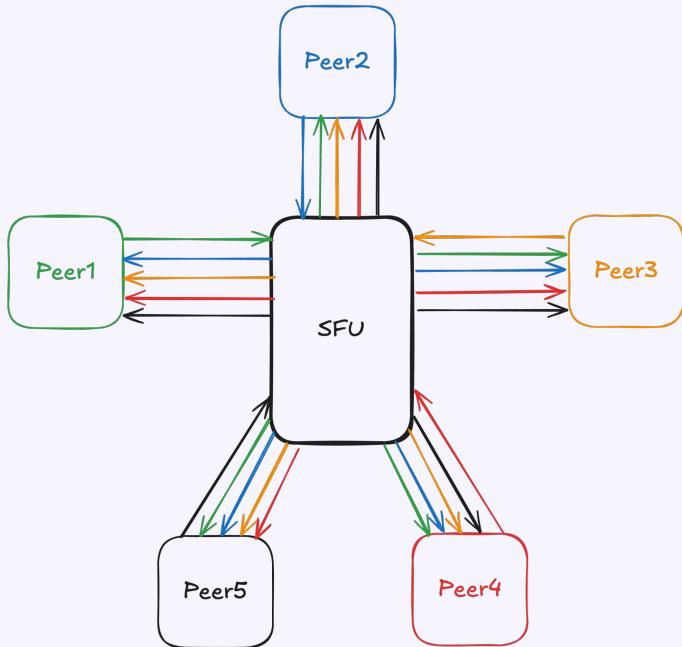
MCU - Multipoint Conferencing Unit

- Znana z telekomunikacji alternatywa
- Sygnał od użytkowników zbierany na serwerze, gdzie generowane jest wyjście specyficzne dla każdego z nich
 - dla audio - zmiksowane źródła z wycięciem audio danego usera
 - dla wideo - wideo aktywnego mówcy, grid (wszyscy widzą to samo!)
- Jedno połączenie (2-kierunkowe)
- stałe wymagania pasma
- Wymaga bardzo kosztownego obliczeniowo przetwarzania multimedii w czasie rzeczywistym (horrendalne koszty infrastruktury przy wideo)
- Umożliwia łatwe tworzenie nagrań w 100% oddających to, co widzą odbiorcy bez dodatkowego processingu
- Pozwala dodawać efekty po stronie serwera (np. blur tła)



SFU - Selective Forwarding Unit

- W większości przypadków złoty środek
- Media przekazywane są bez dekodowania
 - Może to być wada, gdy potrzebna jest analiza/obróbka po stronie serwera
 - Stworzenie nagrania wymaga dodatkowej pracy
- 1 upload, $n * \text{download}$
- Stałe wymagania dla uploadu, liniowo rosnące dla downloadu
 - akceptowalne przy łączach asymetrycznych
 - dany klient może ograniczać odbieranie niektórych sygnałów (np. wideo usera, którego nie wyświetla na ekranie)

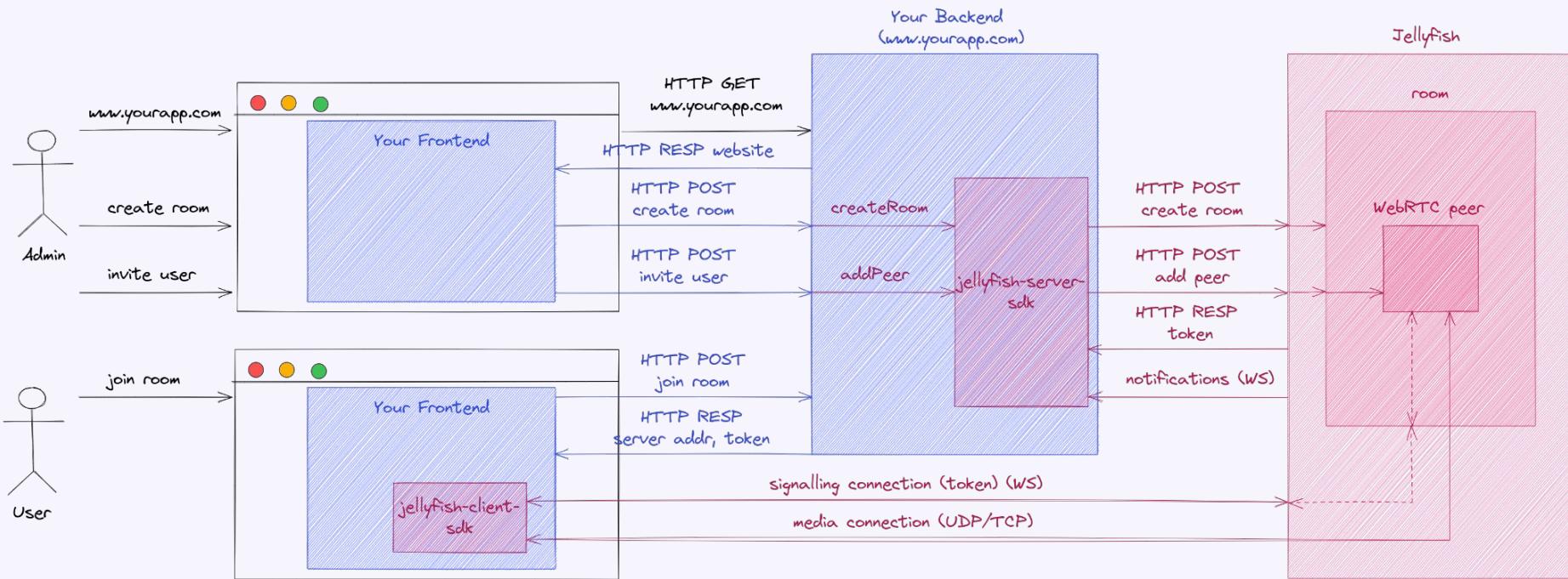


Architektura SFU

- Serwer:
 - wystawia API do komunikacji - HTTP, WS, Webhook, GraphQL, Protobuf
- Serwer SDK:
 - służy do zarządzania serwerem (zarządzanie pokojami, użytkownikami, serwerowymi funkcjonalnościami np. nagraniami)
 - ukrywa serwerową część API serwera
- Client SDK - wykorzystywane przez aplikację kliencką do połączenia się z serwerem
 - ukrywa kliencką część API serwera
 - ukrywa API WebRTC



Architektura SFU

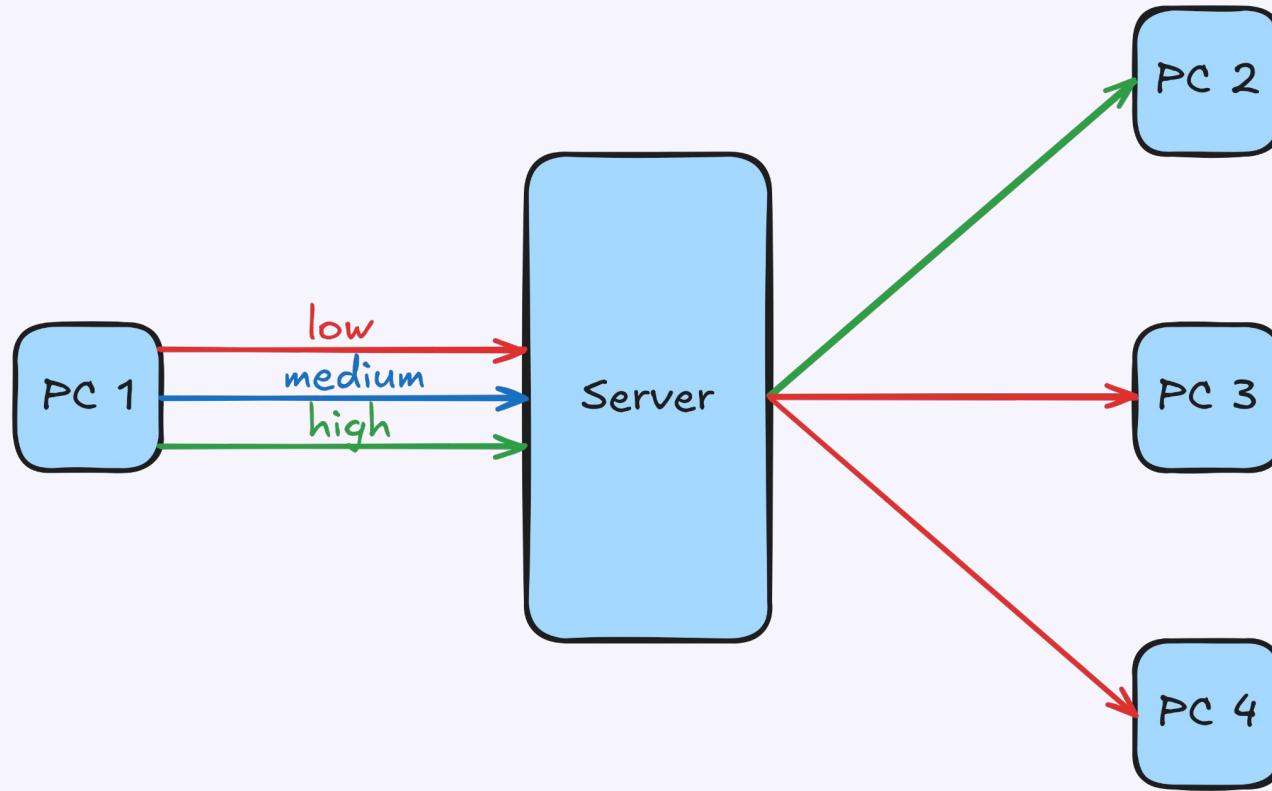


Jak rozwiązać problem różnych warunków sieciowych u różnych odbiorców?

- simulcast
- SVC



Simulcast



Simulcast

- decyzję o wyborze konkretnej warstwy podejmuje serwer na podstawie:
 - warunków sieciowych
 - wielkości kafelków
 - preferencji użytkownika
- trzeba zachować płynność transmisji - przełączanie następuje tylko na keyframe'ie
- odbiorca nie jest świadomy simulcastu - zawsze odbiera pojedynczy stream RTP
- serwer musi sprowadzić numery sekwencyjne i timestampy RTP, oraz część pól kodeków z RTP payloadu do tej samej przestrzeni - RTP munging
- RTP munging nie jest ustandaryzowany



Określanie przepustowości sieci

- probing - wysyłanie nadmiarowych danych do momentu aż zaobserwujemy packet loss lub zwiększyony delay
- wykorzystuje RTP padding pakiety lub RTX
- nie jest ustandaryzowany



Optymalizacje w WebRTC

- wysyłanie wideo max 50-100 użytkowników - paginacja
- wysyłanie audio tylko najgłośniejszych osób
- przesyłanie rozdzielczości dopasowanej do wielkości okienka
- zatrzymywanie wysyłania wideo kiedy użytkownik przełącza kartę przeglądarki



Dziękujemy!





v0.1.0

