Web Application for Hybrid Communication Integration

Vasko Mykhailo [1], Gregor Rozinaj[1]

[1] Slovak Technical University, FEI, 84104, Slovakia

vasko.mike.dev@gmail.com

Abstract - **This project will be implemlenting a method of hybrid communication into a web application, that will allow to communicate with every person on the call more comfortably. (*Abstract*)**

Keywords – **Mediasoup, Hybrid Communication, SFU**

I. Introduction

In the modern world video calls have become a common reality. We got used to them during the pandemic, working and studying from home. For me, it was just studying, and I've experienced a frustration of bad quality communication during online classes and meetings. In our university, for example, classrooms usually have a projector and a computer, but they aren't really made for hybrid use. Often, online students feel like observers instead of active participants. This happens because the focus stays mostly on what's going on in the room, and the tools used (like Zoom or Meet) don't fully connect both groups.

My project aims to improve that by creating a custom web app that works better with real classrooms. The goal is to use existing classroom equipment — like cameras and screens above the whiteboard — to make remote students more visible and more connected to what's happening on-site. The app lets each participant, whether remote or local, be displayed in the classroom as if they were there and makes it easier for teachers to interact with everyone equally.

This paper briefly explains how the system works, how it was built.

II. Related Background

Hybrid communication isn't a new idea, and during the pandemic, tools like Zoom Rooms, Microsoft Teams Rooms, and Google Meet became very popular. These apps made it possible to run classes online and had some versions of advanced videoconferencing features. They mainly focus on screen sharing and video calls, collaboration during the calls, combining software with physical spaces.

There are also some projects that used special hardware or AI to switch camera views automatically or show different people depending on who is talking. But again, these systems are often not affordable or flexible for smaller classrooms. The main problem with their approach (apart from using proprietary technology) is that their main goal is to make calls more convenient from the perspective of classic videoconferencing meeting, addressing some problems, like dividing two people using single camera into two separate tiles in a call, but not addressing the main issue of feeling separated from the class during a lesson.

My project takes a different approach. Instead of using expensive tools, it focuses on making a simple web app that connects to the equipment most classrooms already have. It's built using open web technologies like **WebRTC** for real-time video and audio, and **Angular** for the front-end. On the server side I decided to use **Mediasoup** SFU for handling video streams between users, efficiently managing the resources.

Why did I choose SFU implementation from Mediasoup?

After researching multiple libraries for videoconferencing, I've decided to use Routing s multiparty topology approach with SFU. SFU stands for Selective Forwarding Unit, and it acts like a router for media. SFU receives incoming media from all participants and then decides to whom which media would route. This implementation reduces the server load, making users able to process many incoming media streams, having high scalability and low infrastructure cost in comparison to MCU and Mesh topology.

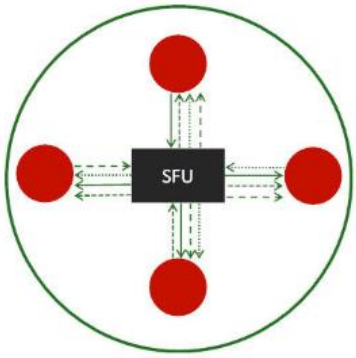
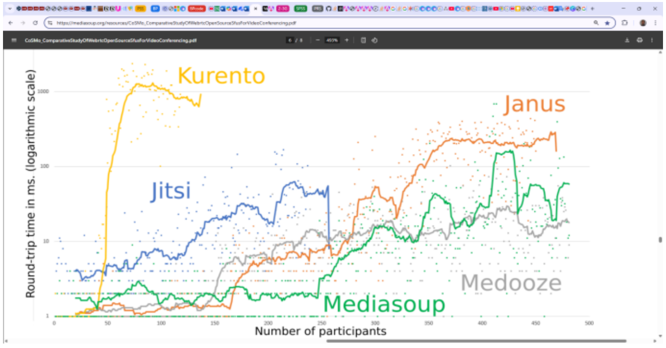
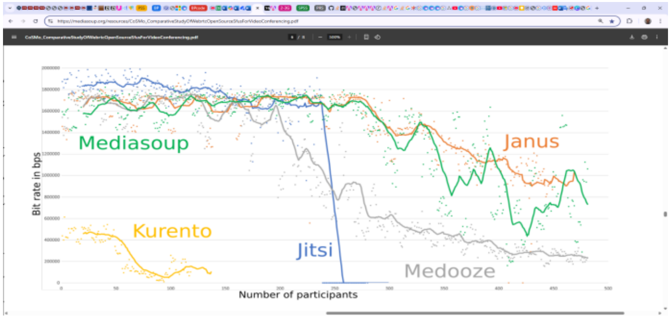
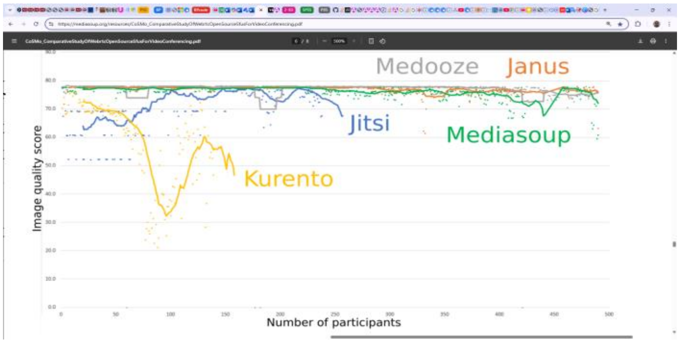


Figure 1. Routing topology

The library with SFU implementation was the hard choice, since there are many of them in open source. After exploring difference between them, the choice was made in favor of Mediasoup:







On these graphs we can see that Mediasoup library is one of the best performing SFUs, having balance between round-trip time, bit rate and image quality score [1].

III. System Architecture

My app is divided into a front-end part and a back-end part, which communicate through Web-sockets. Client after establishing a connection to server through socket starts communicating with a server, creating room, a send/receive transports for participants, and so on. The main stages for creating an audio and video stream to the server and connecting devices are:

- Creating a router(room) on the server side or joining to existing one
- Creating a device object on client side for each participant and loading router parameters onto it
- Creating WebRTC transport for sending to the server using device methods, then start producing data from cameras/microphones to the server
- Send request to all sockets in the belonging to the same room (same router) to create receive transports
- Parameters of these transports return to users, using them they should create Receive transport and consume data from server.

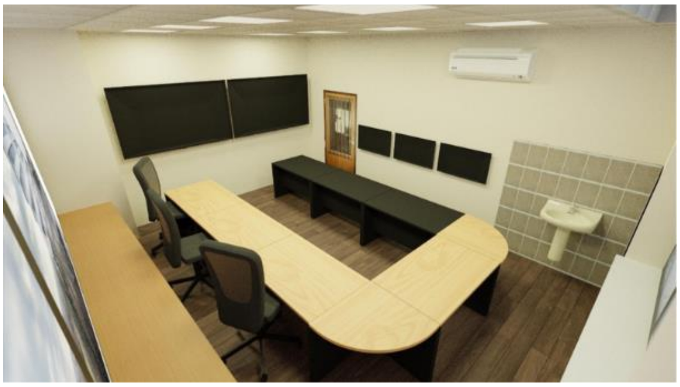


Figure 2. Example of a room eqipped with displays and cameras

The hybrid communication is achieved by accessing displays and cameras assigned to them, displaying each participant on a separate display. With this approach the edge between virtual and real meetings becomes thinner, creating an illusion of participants being present in the room.

IV. Implementation

My implementation of front-end is made using Angular framework. Brief description of UI features:

- Local video is displayed on the bottom right corner of the screen, so you will be able to see yourself.
- Other participants are displayed in the grid, with the ability to resize every video for convenience to be able to highlight main speakers/demonstrators.
- Client has an ability to choose main speaker, and make his video full screen, having other's participants video at the bottom line.
- Another option is to detach participant's video from the grid to separate window, and move it freely to other screens, or sides of display. This will help the viewer to focus better on the presenter.
- By default, the user gets a video feed from the camera assigned to display, but user can choose the video feed from any camera in the conference room, to see the whole environment in the room or remote participants.
- As addition you have an ability to request screen sharing of any user, or share your own, which is helpful for presentations.

This logic allows us to use any setup either its well-equipped room or simple classroom with few people and laptops, changing it to system with hybrid communication.

V. RESULTS

As a result, I have a working video-conferencing application, that allows to connect people and provide better communicating experience. The logic behind it is that users that are in the classroom provide their devices as screens for setting up virtual meetings, where remote users would be displayed.

VI. CONCLUSION AND FUTURE WORK

My application helps to improve videoconferencing experience by allowing to replace the need for physical attending of person with display and camera – setup, usually available everywhere. This way, the feeling of being present in the room and comfortably discussing any topic. Also, my app leaves room for improvement in functionality, such as adding virtual drawing board workspace into every room, or projecting selected remote user to the real-world whiteboard. In the perfect world, this system will allow remote participants to display in real-life classrooms as they were in there and give remote users a 3D view of the classroom with people.

A. *Abbreviations and Acronyms*

SFU – Selective Forward Unit

MCU – Multipoint Control Unit

ACKNOWLEDGEMENT

This paper was supported by DISIC (09I05-03-V2), NEXT (ERASMUS-EDU-2023-CBHE-STRAND-2), and EULiST (ERASMUS).

REFERENCES

[1] Emmanuel Andre´ ∗ , Nicolas Le Breton∗§, Augustin Lemesle∗§, Ludovic Roux∗ and Alexandre Gouaillard, "Comparative Study of WebRTC Open Source SFUs for Video Conferencing", page 6 https://mediasoup.org/resources/CoSMo_ComparativeStudyOfWebrtcOpenSourceSfusForVideoConferencing.pdf

[2] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[3] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.