Exploring the roads to multipath live streaming in WebRTC

Group: 14

Presentation By:

Arnav Gupta 2021236

Karan Gupta 2021258

Shivesh Gulati 2021286

Outline

- This project aims to explore the possibility of implementing and designing a multi-path WebRTC framework capable of live-streaming manipulated frames.

- Currently, it supports single-path streaming.

- As part of the project, we will first extend the WebRTC codebase to enable multipath streaming of video-feed.

- Afterwards, a brief performance study of the impact of different schedulers and various frame-splitting techniques on streaming latency and quality will be conducted.

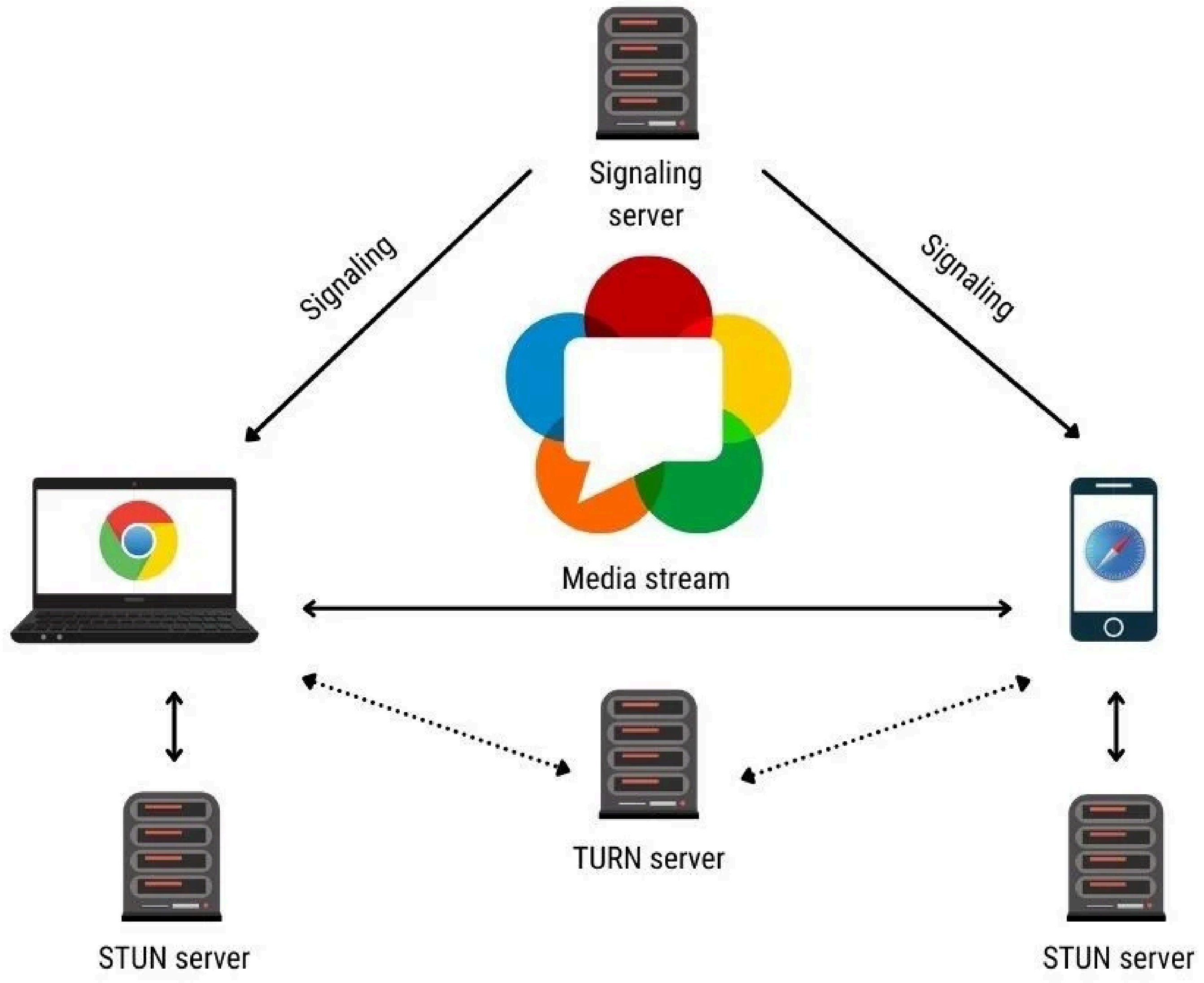
About Webrtc

- WebRTC enables you to integrate real-time communication features into your application using an open standard. It supports video, audio, and general data transmission between peers, allowing developers to create robust voice and video communication solutions.

How webrtc works

Components of Webrtc

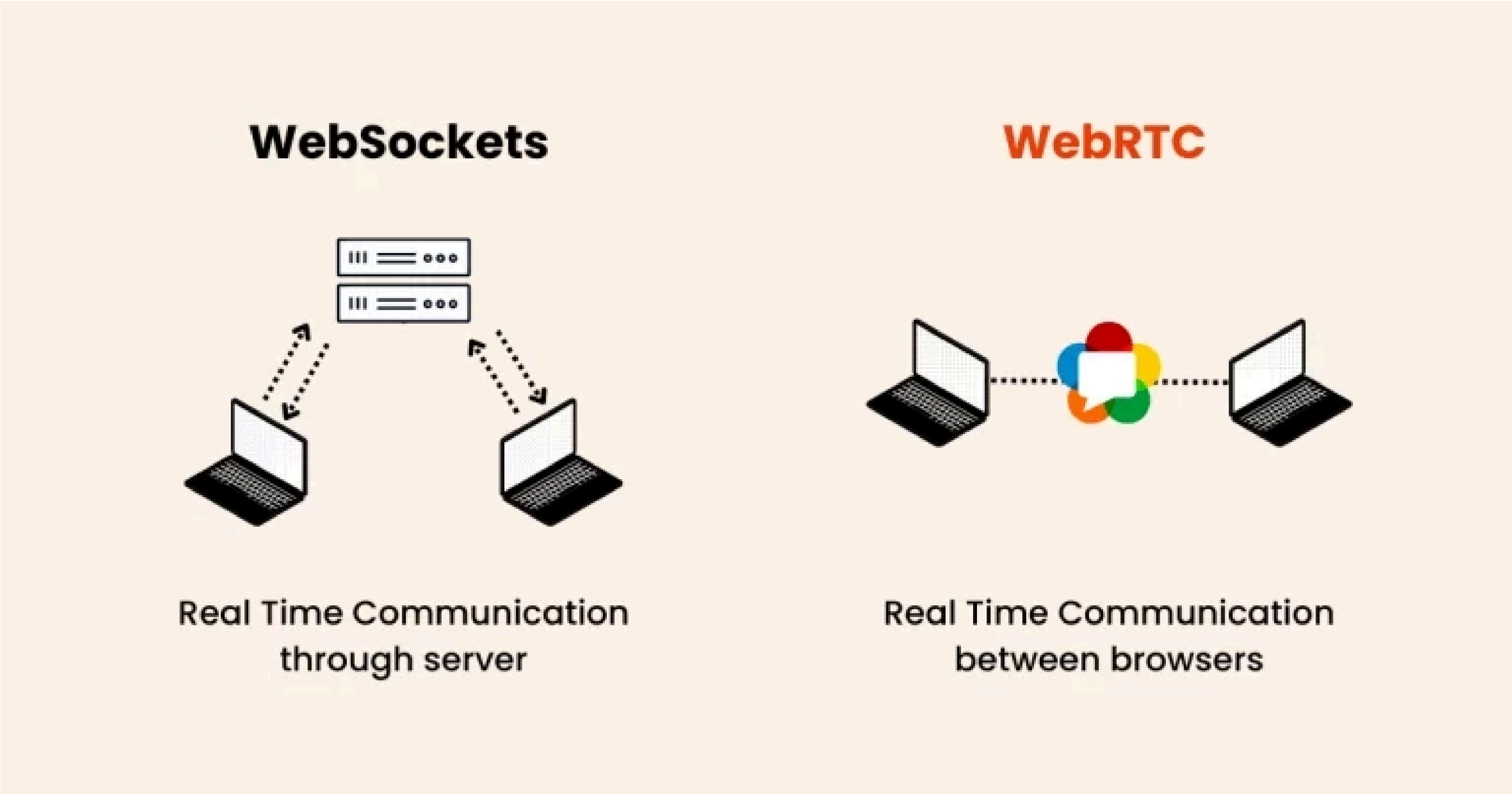
- Peers
- SDP
- Stun Server
- Signalling Server
- Turn Server



Peers

Peers are the client browsers which directly communicate through webrtc without the need for a server for real time communication.

Webrtc is a peer -to- peer protocol



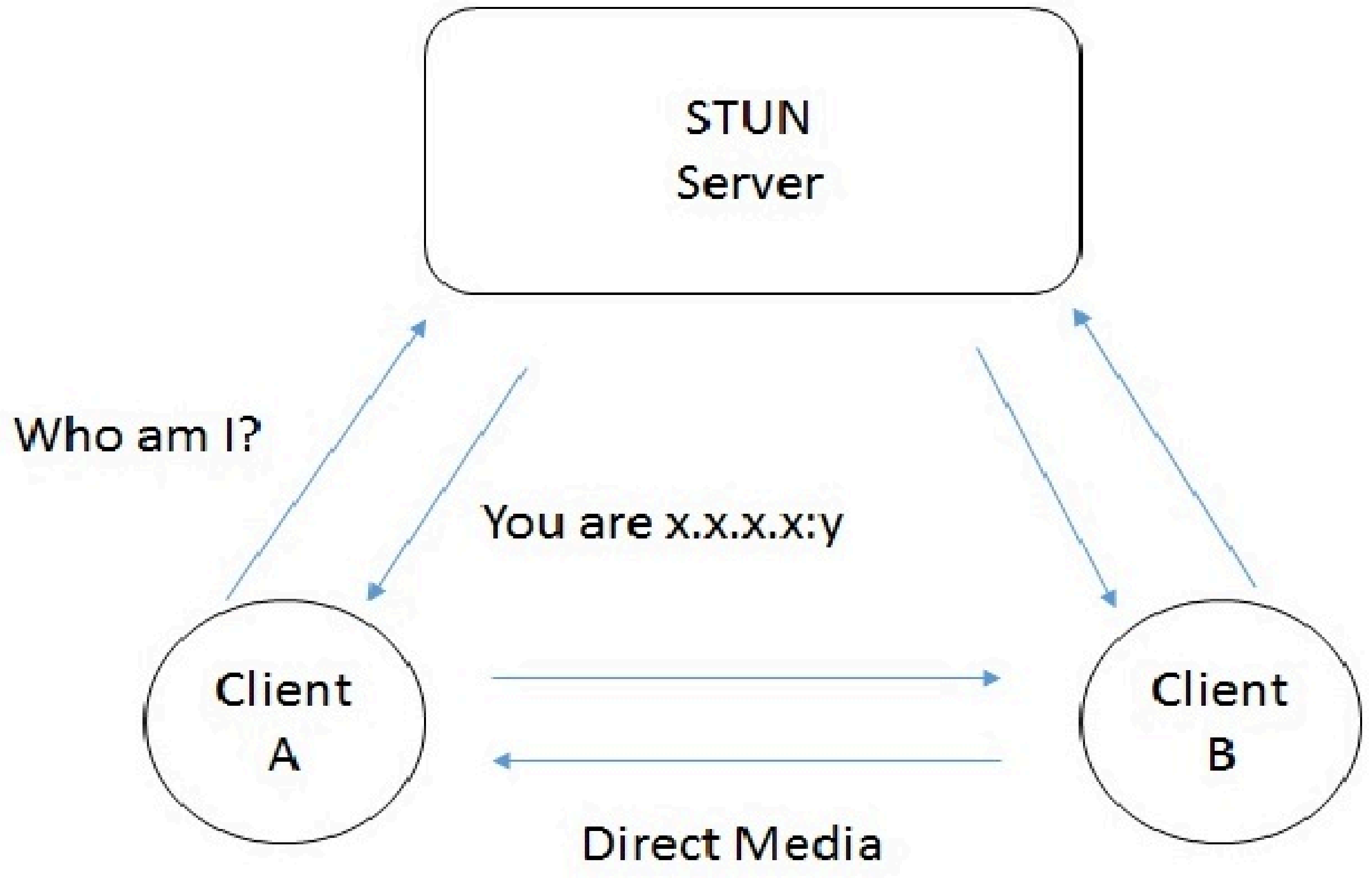
Software Development Protocol (SDP)

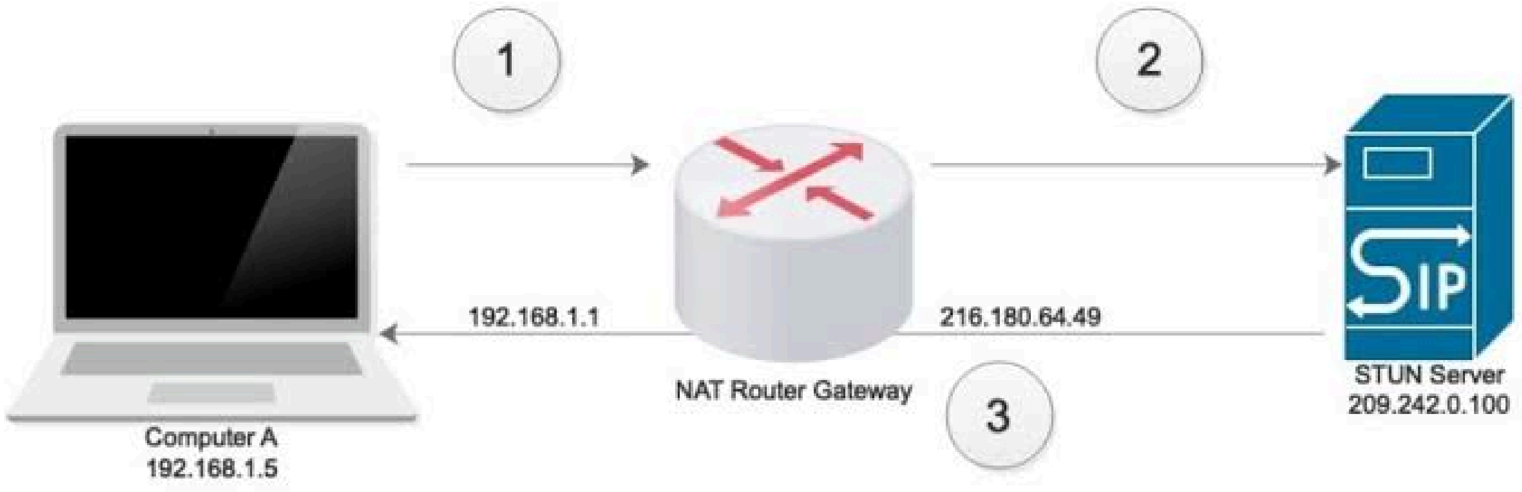
Session Description Protocol (SDP) is a text-based format used to describe multimedia sessions in WebRTC. It serves as a standard for establishing peer-to-peer connections by exchanging SDP messages between devices. SDP is essential for WebRTC because it enables devices to negotiate key details for a successful connection, such as:

- **<u>Media formats:</u> The types of media streams to be sent and received**
- **<u>Transport protocols:</u> The communication protocols to use**
- **<u>Codec:</u> The specific codec for encoding and decoding media**

Stun Server

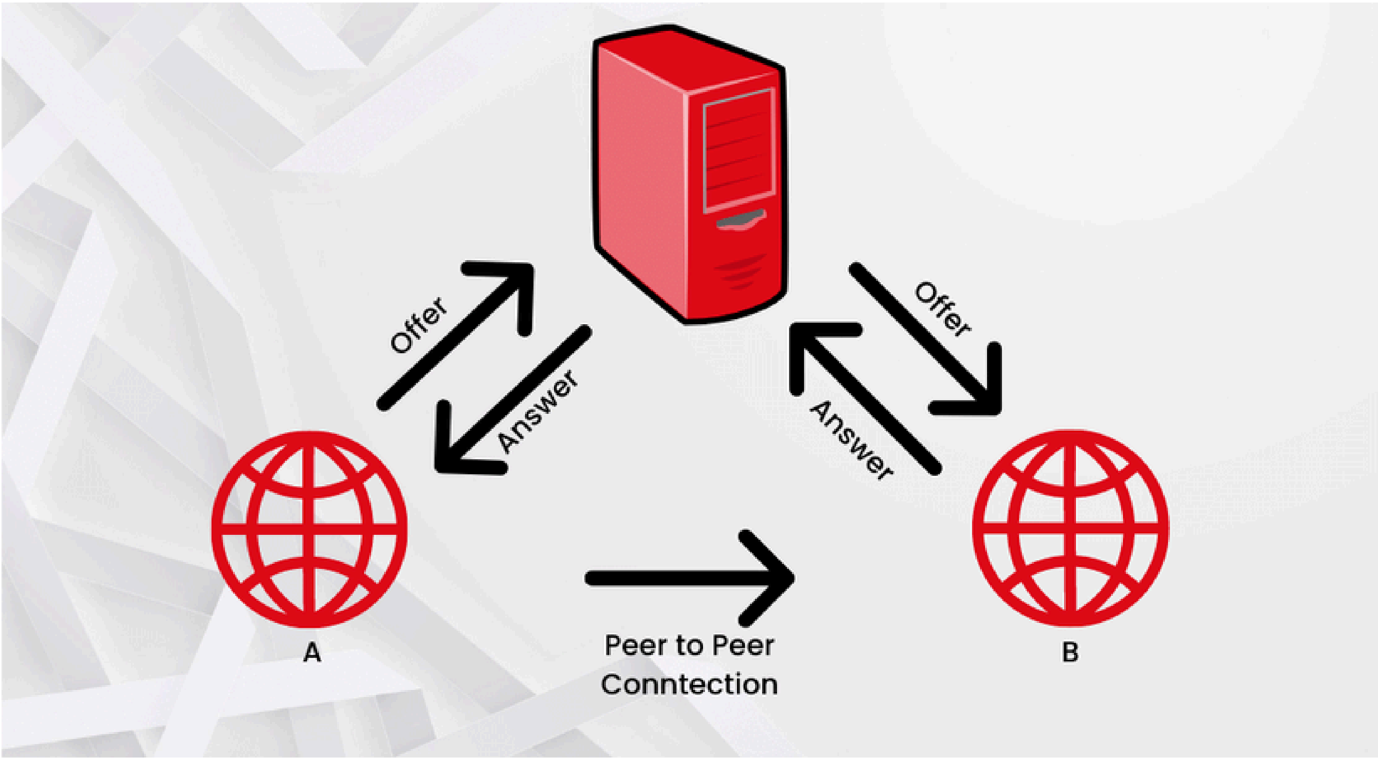
STUN (Session Traversal Utilities for NAT) works alongside ICE to help navigate NATs using the UDP protocol. It enables applications to detect the presence and type of NATs and fire-walls between them and the public Internet. Devices can use STUN to identify the IP address and port assigned to them by a NAT.

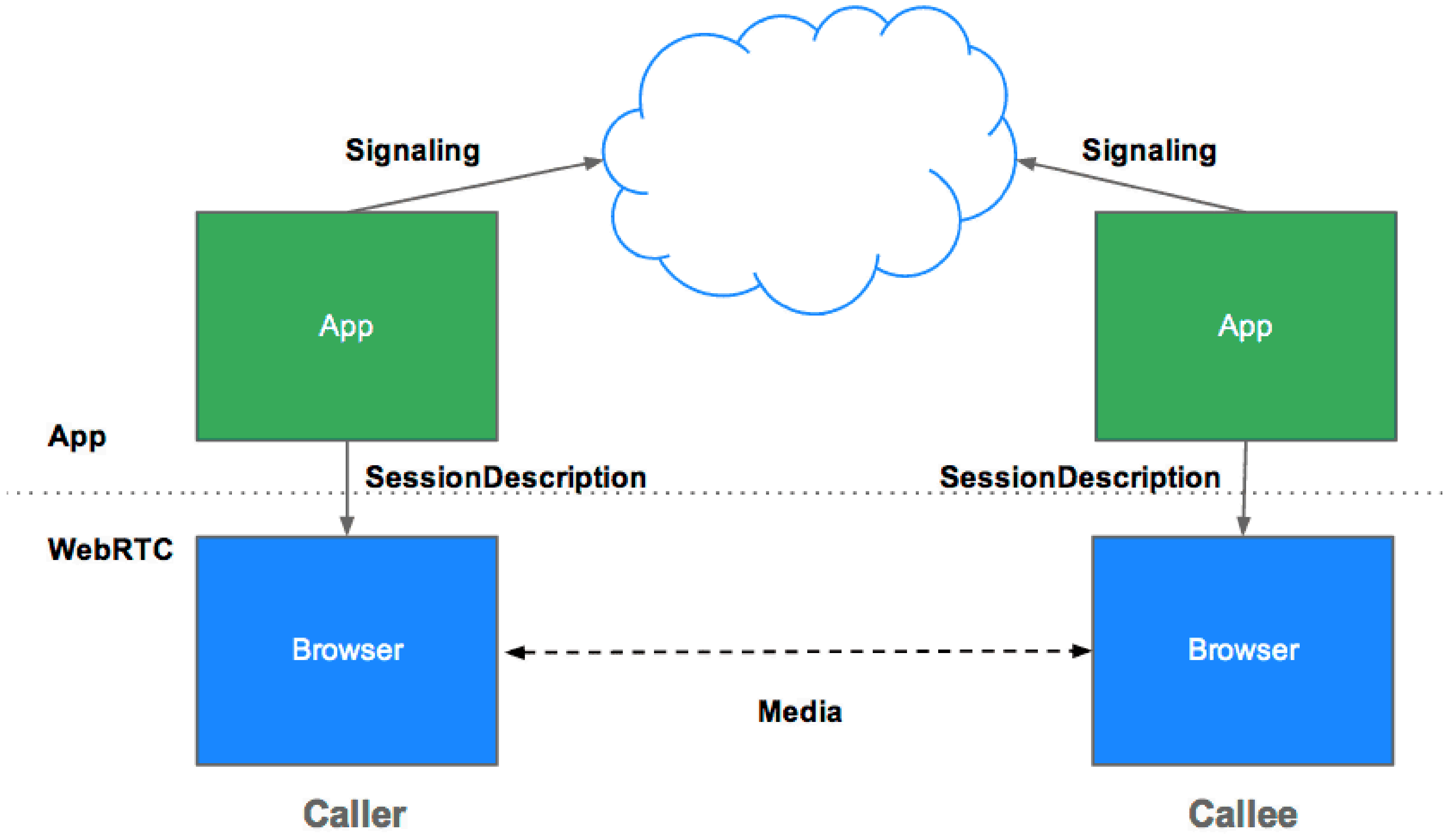




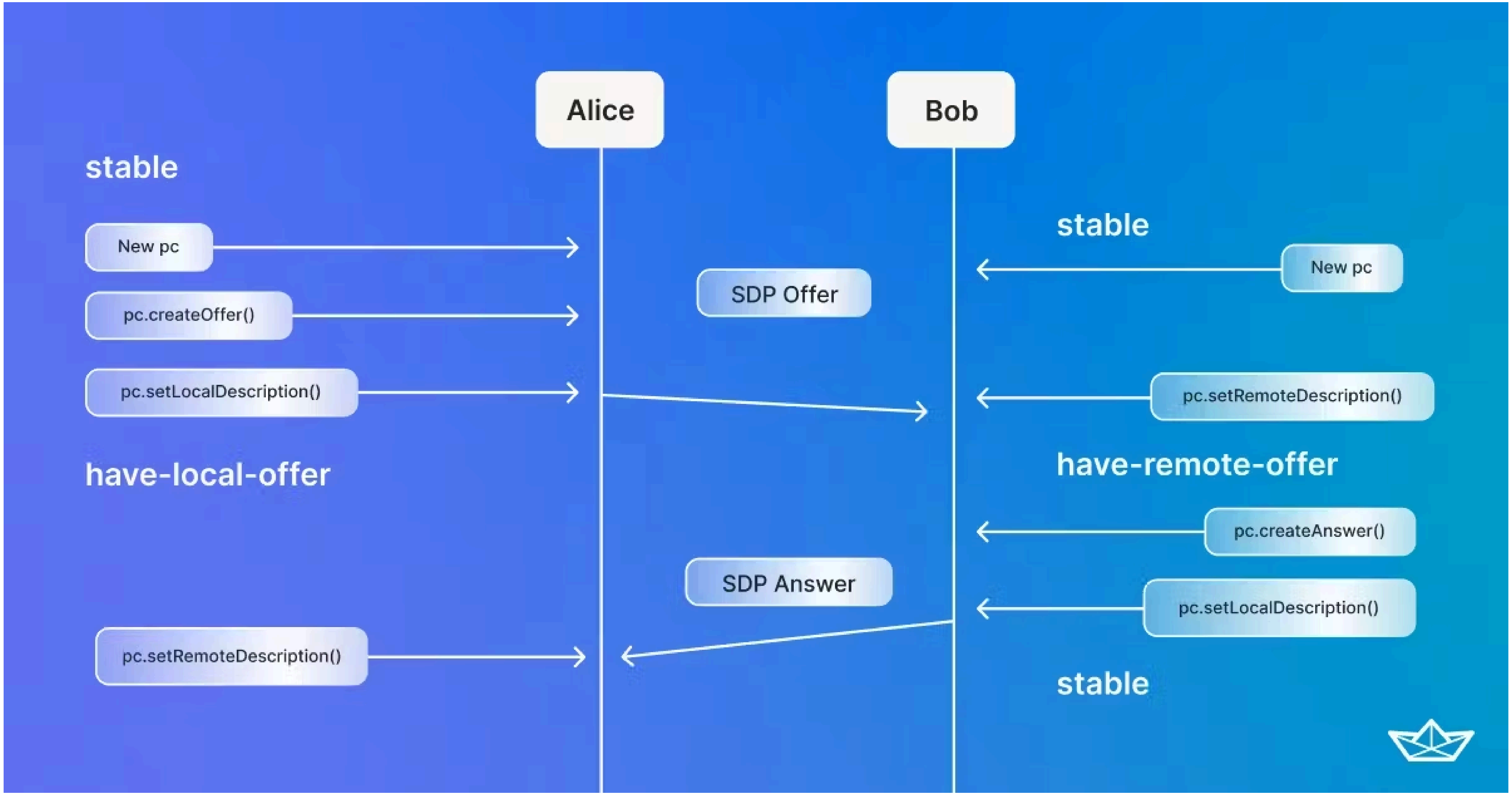
Signaling Server

The signaling server is essential for facilitating and establishing connections between peers, allowing them to communicate while reducing the risk of exposing private information. This is used for transmitting offers and answers between the peers.

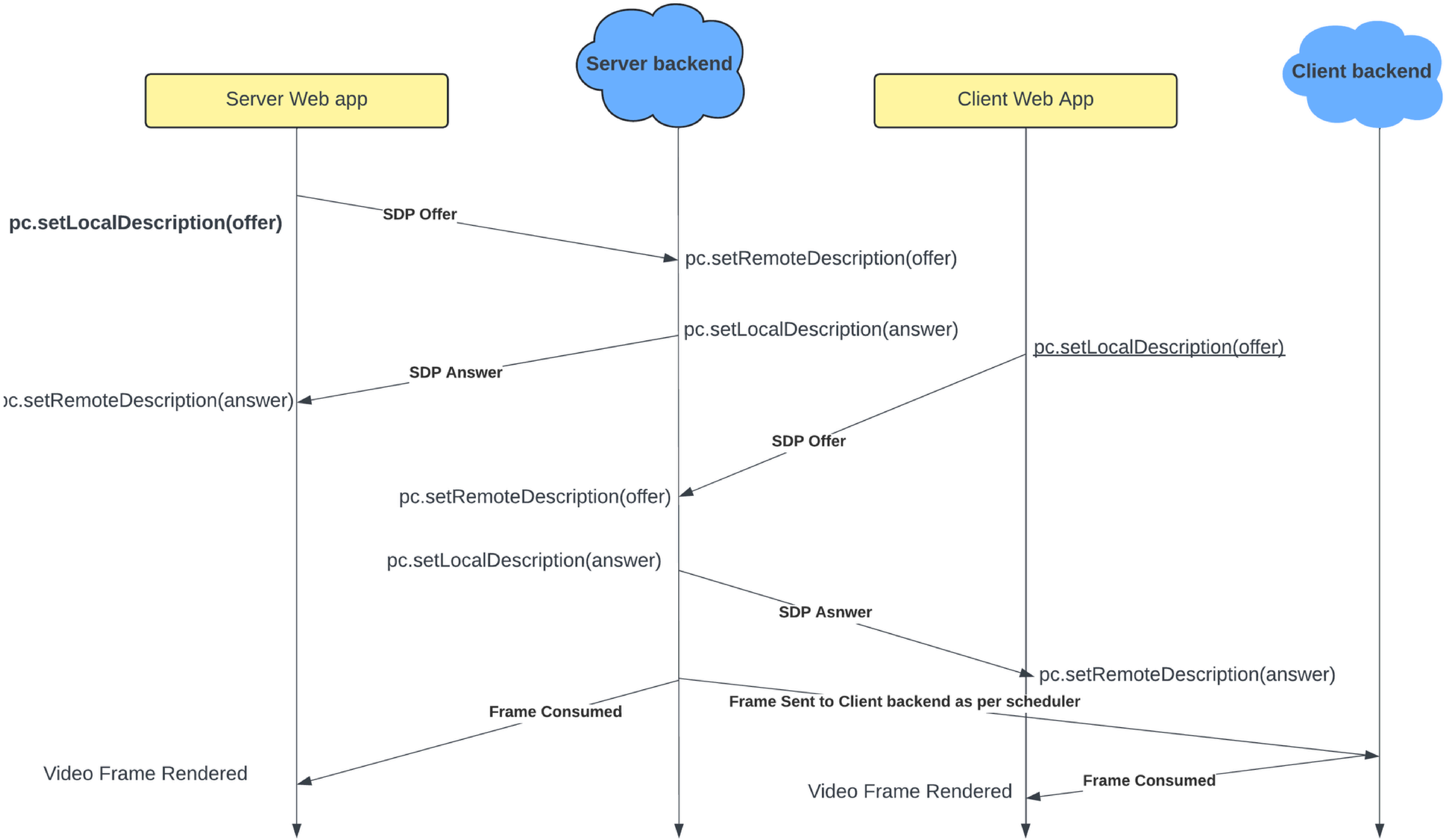




Connection Setup Basic



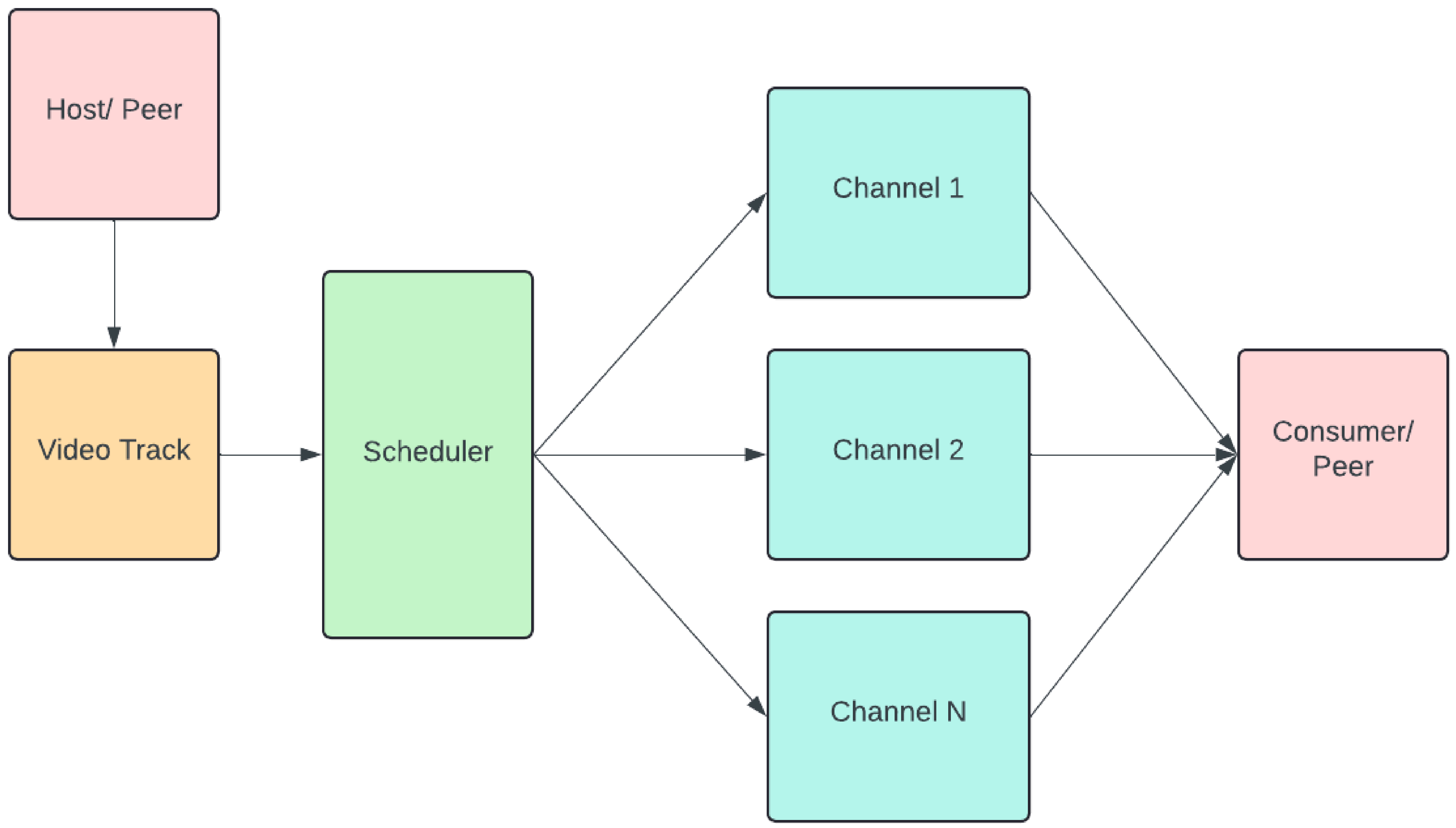
Our Setup



Server Web app
Server backend
Client Web App
Client backend
pc.setLocalDescription(offer)
SDP Offer
pc.setRemoteDescription(offer)
pc.setLocalDescription(answer)
SDP Answer
ɔc.setRemoteDescription(answer)
pc.setLocalDescription(offer)
SDP Offer
pc.setRemoteDescription(offer)
pc.setLocalDescription(answer)
SDP Asnwer
pc.setRemoteDescription(answer)
Frame Consumed
Frame Sent to Client backend as per scheduler
Video Frame Rendered
Video Frame Rendered
Frame Consumed

Multipath Setup

- <u>Video Track</u>- The video stream from the camera
- <u>Scheduler</u>- Round Robin Scheduler, redirects frames from track to each consumer
- <u>Channel</u>- each channel receives some part of the original video

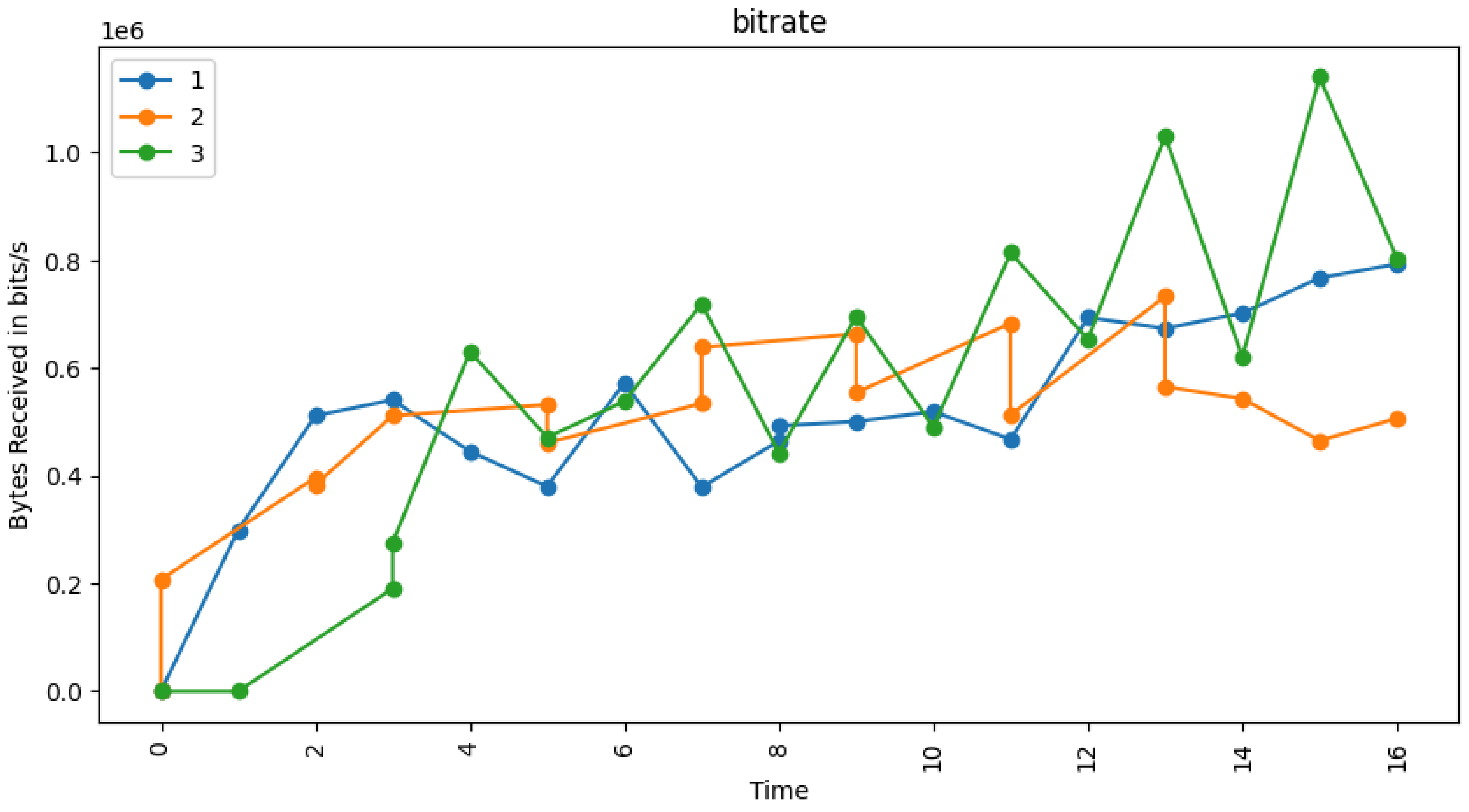


Bit-Rate Analysis

- The graph shows an analysis for the bytes received v/s time for varying number of connections.

- Initially we observe that both 2-connection setup and 3-connection setup have a lower bitrate than the 1-connection setup, which is mainly due to the time it takes to initialize the connections at the start which results in sending of a few empty frames.

- Once the connections are up, it can be seen that the bit rate of 3-connection setup and 2-connection setup are higher than the one-connection setup on an average which is exactly what we expected.

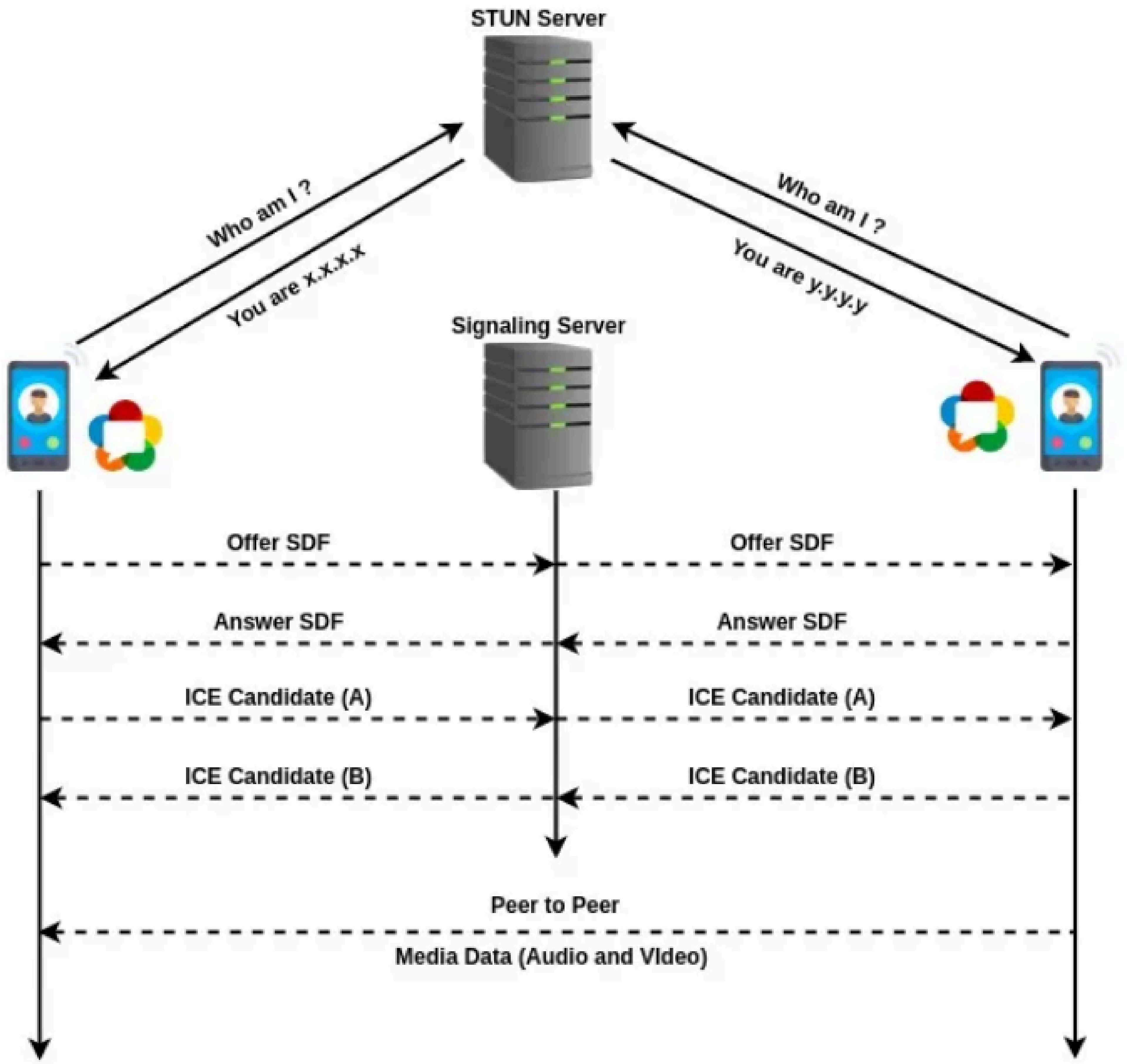


Milestones achieved

- Setup basic connection between 2 peers
- Setup multipath setup by building wrappers around the library function.
- Implemented basic Round Robin Scheduling to send frames across connections and analyzed the bit-rate.
- Built Basic UI of the web-app.

Future Milestones

- In our current setup our peers are connected to each other via a tunnel and so they can ping each other directly being connected to a NAT.
- In real life however the connecting peers may be connected to different networks and hence cannot simply directly ping each other.
- To fix this, generally STUN and signaling servers are used in combination with WebRTC applications that we wish to implement in our design in the next half of the semester.



Future Milestones (Contd)

- Implement different schedulers to handle multipath setup such as min RTT,minCT and compare performance with single path WebRTC

- Use additional techniques such as frame splitting and get the performance metrics.