WebRTC Lab Assignment

> Audio/video streaming cloud applications
>
> **Surbhi Sonkiya** (EIT Digital Master School)

Introduction

This document presents observations to the various experiments performed as part of WebRTC lab assignment.

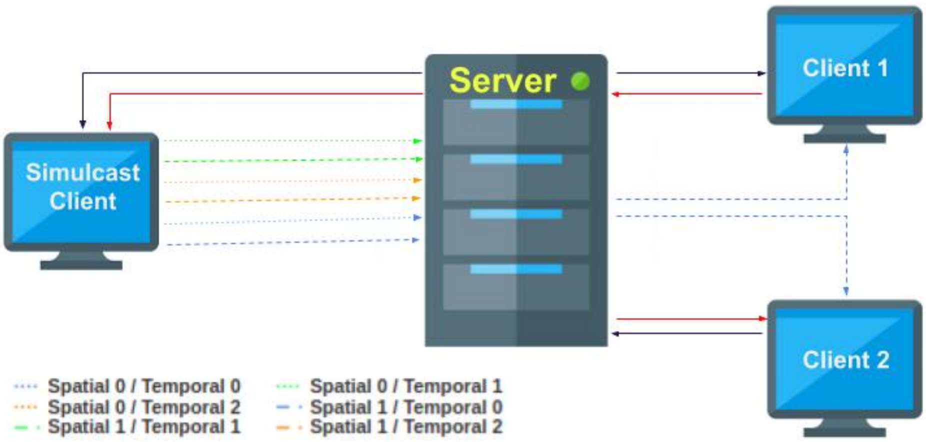
Q1. Represent in a diagram the current architecture from the media point of view (ignore the control and signalling components). Include the quality_layers.

Launch simulcast client with URL (https://192.168.56.101:3004/?simulcast=true) and launch two regular client with URL (https://192.168.56.101:3004).

Architecture Diagram

Simulcast client sends six streams with different combinations of spatial layer [0,1] and temporal layer [0,1,2] while the regular clients send only one stream with spatial layer as 0 and temporal layer as 0.

- **Spatial layer**: captures the differences between pixels within a frame because nearby pixels within a frame maybe similar and therefore, only those different pixels could be loaded.
- **Temporal layer**: captures the differences between the adjacent frames because adjacent frames could be similar and therefore, only those differences between the frames could be loaded.



Quality_layers

As observed from the following **quality_layers** that both of the clients are able to receive the maximum bandwidth with temporal layer as 2. Further, both the regular clients are sending streams with spatial and temporal layer as 0,0 respectively. Moreover, the observation shows that spatial and temporal layers overlap. It is possible to see all three temporal layers [0,1,2] while it is difficult to distinguish the spatial layer [0,1] combinations with these three temporal layers.

Additionally, The "**Current Received**" line shows the bandwidth of both video and audio streams. However, the layers line represents bandwidth of only video streams.

Q2. Represent in a diagram the new scenario. Explain the changes with your own words. Include a screenshot of the new quality_layers graphics.

To lists the ids of the streams available in the room

```
room.remoteStreams.keys()
```

Modify spatial and temporal layers with the following command and observe the changes in the quality_layers shown below.

```
room.remoteStreams.get(id)._setStaticQualityLayer(spatialLayer, temporalLayer)
```

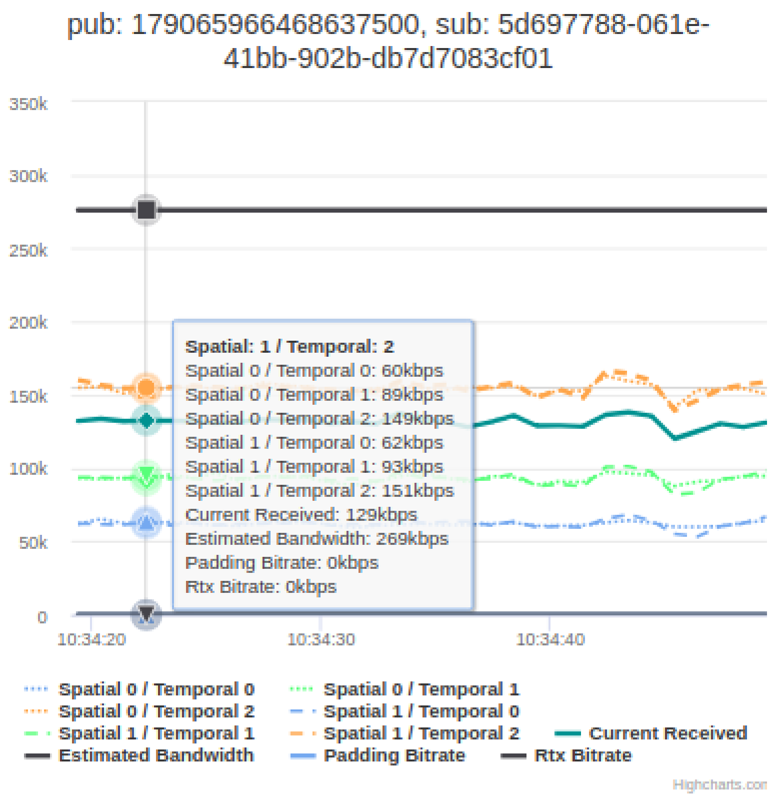
Output

It can be observed from the below screenshot that the first command displays 3 remote stream ids. The ids are in the order of the client joining the room. Since, simulcast client joined after one regular client, hence, it's id is the second one.

```
> room.remoteStreams.keys()
<· ▶ (3) ["164379682288576320", "179065966468637500", "268309573759258800"]
> room.remoteStreams.get(179065966468637500)._setStaticQualityLayer(1, 1)
  DEBUG:  MaxVideoBW Changed to  undefined                          erizo.js:100
  DEBUG:  MinVideo Changed to  undefined                            erizo.js:100
  DEBUG:  SlideShowMode Changed to  undefined                       erizo.js:100
  DEBUG:  muteStream changed to  undefined                          erizo.js:100
  DEBUG:  Video Constraints undefined                               erizo.js:100
  DEBUG:  Will activate slideshow when below layer undefined        erizo.js:100
  INFO:  Sending message ▶ {type: "updatestream", config: {…}}      erizo.js:100
  179065966468637500 179065966468637500
```

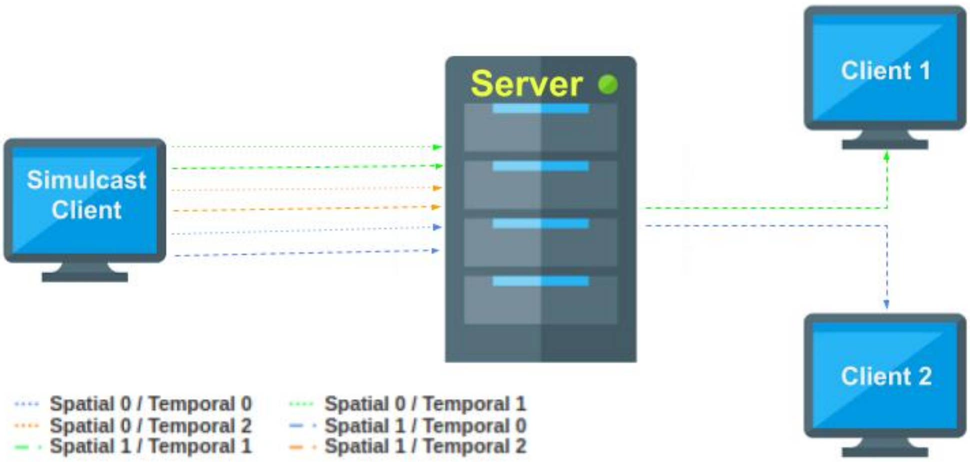
Quality_layers changed for the client

From the quality_layers, it can be observed that one of the clients is able to receive the stream that has bandwidth with spatial layer 1 and temporal layer as 1.



Modified Architecture Diagram

The change in "Current Received" stream for Client 1 has been represented in the architecture diagram.



Q3. Which is the maximum video bandwidth a client could receive with this configuration? Explain it.

As it can be seen from the `licode_config.js` file, the below two video bandwidth attributes have 300 kbps value. These are default and maximum bandwidth parameters to be used by clients for both published and subscribed streams.

`licode_config.js`

```
config.erizoController.defaultVideoBW = 300;
config.erizoController.maxVideoBW = 300;
```

Also, the same can be observed from the quality_layers (shown in Q1) that the maximum video bandwidth in general that a client can receive is 300 kbps.

However, after modifying the spatial and temporal layer for one of the clients and setting it to 1,1 respectively, the maximum video bandwidth for that client reduced from 300 kbps to 93 kbps (see quality_layers shown in Q2).

Q4. Explain the changes in the environment after bandwidth modification.

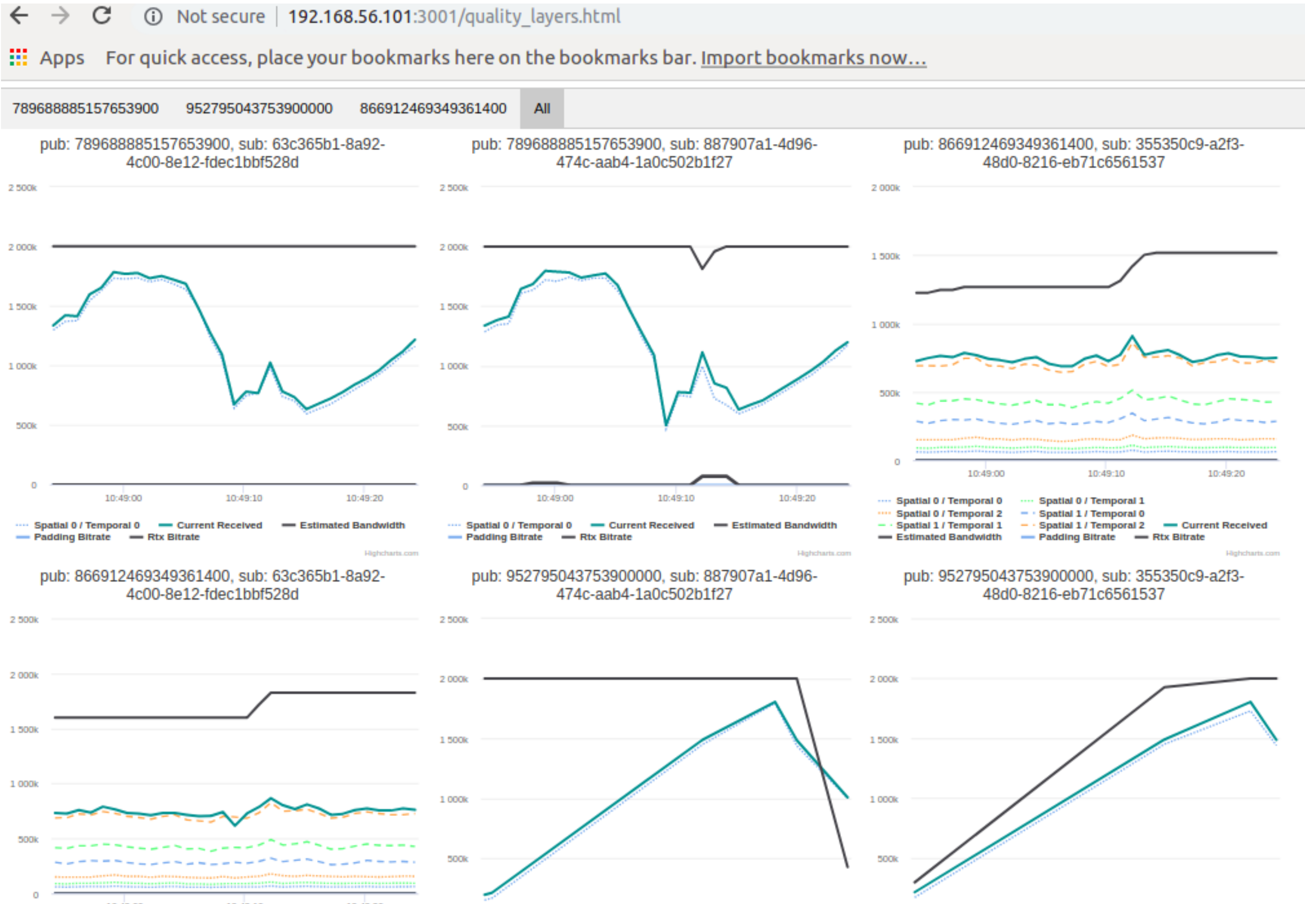
The below two attributes are modified to 2 Mbps in the `licode_config.js` file.

`licode_config.js`

```
config.erizoController.defaultVideoBW = 2000;
config.erizoController.maxVideoBW = 2000;
```

The changes in the quality_layers is prevalent. The maximum video bandwidth available for the two regular clients is now 2 Mbps.

Quality_layers with 2 Mbps bandwidth



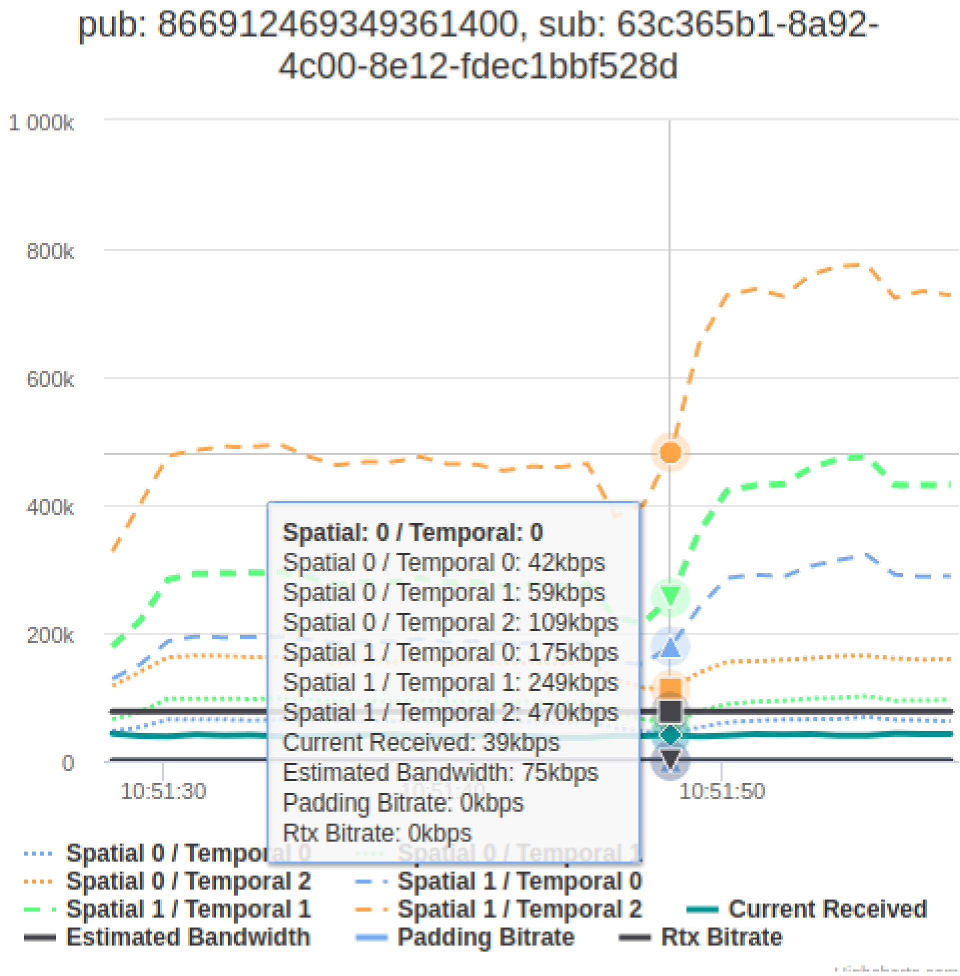
Check the new stream id's and modify spatial and temporal layer for one client

It can be observed from the below screenshot that the first command displays 3 remote stream ids. The ids are in the order of the client joining the room. Since, simulcast client joined after two regular clients, hence, it's id is the last one.

```
> room.remoteStreams.keys()
<· ▶(3) ["789688885157653900", "952795043753900000", "866912469349361400"]
> room.remoteStreams.get(866912469349361400)._setStaticQualityLayer(0, 0)
  DEBUG:  MaxVideoBW Changed to  undefined                       erizo.js:100
  DEBUG:  MinVideo Changed to  undefined                         erizo.js:100
  DEBUG:  SlideShowMode Changed to  undefined                    erizo.js:100
  DEBUG:  muteStream changed to  undefined                       erizo.js:100
  DEBUG:  Video Constraints undefined                            erizo.js:100
  DEBUG:  Will activate slideshow when below layer undefined     erizo.js:100
  INFO:  Sending message ▶{type: "updatestream", config: {…}}    erizo.js:100
  866912469349361400 866912469349361400
```

Quality_layers changed for the client

From the quality_layers, it can be observed that one of the clients is able to receive the stream that has bandwidth with spatial layer 0 and temporal layer as 0.

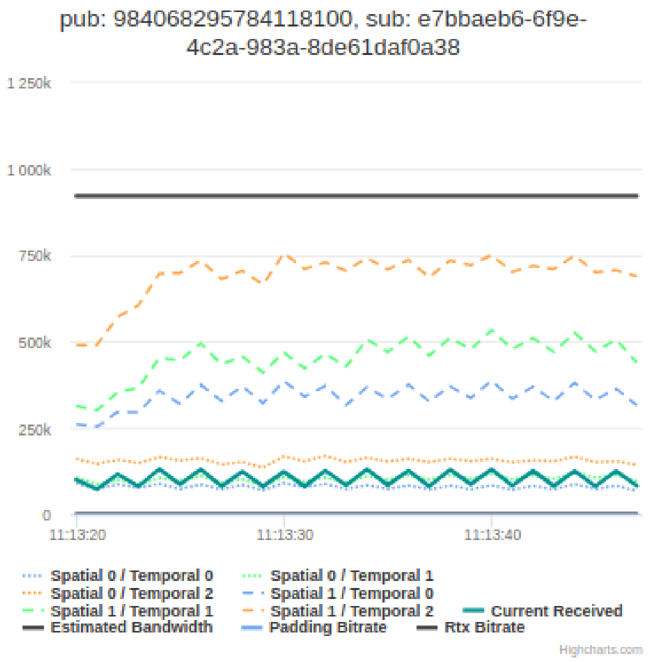


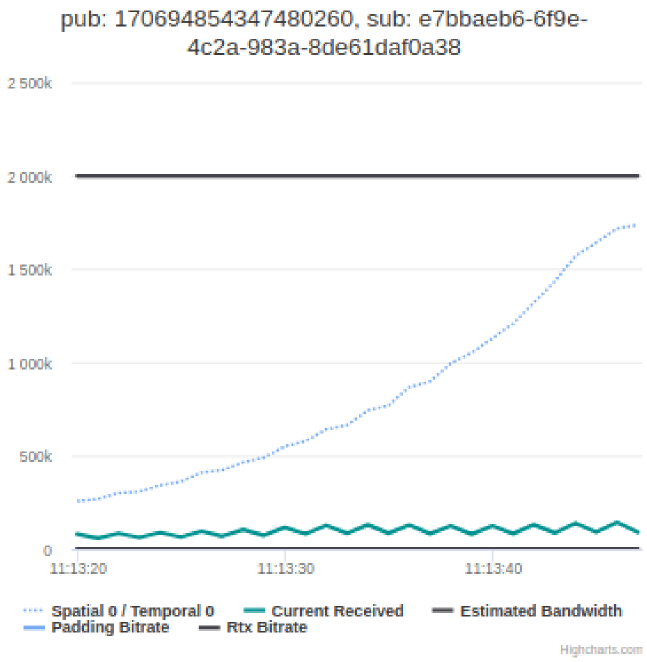
Q5. Test "Toggle Slide Show Mode" functionality and explain the differences between this mode and the simulcast feature. Explain in which use cases (thinking on real videoconferencing applications) you would use each of these features.

Differences between Toggle Slide Show Mode and the simulcast feature is explained below.

Quality_layers when `Toggle Slide Show Mode` On

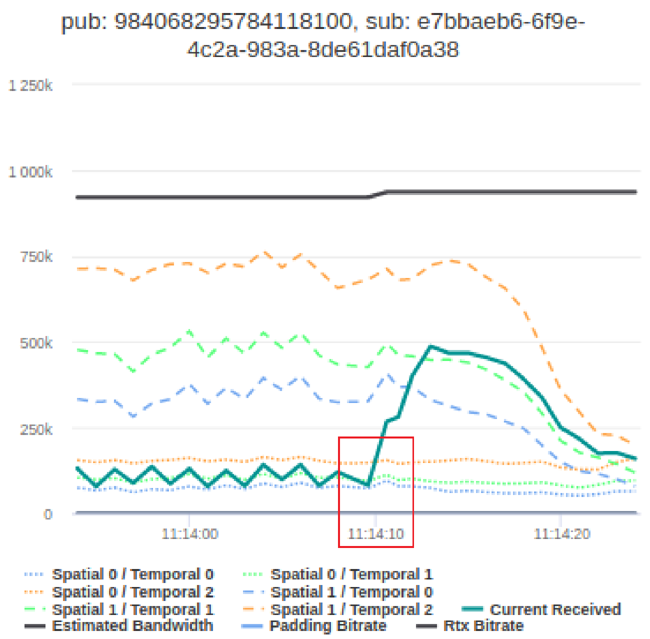
As observed from the below quality_layers that the **Current Received** bandwidth drops to the video bandwidth where spatial layer is 0 and temporal layer is also 0. This shows that the streams **send I-frames** with an interval of certain milliseconds in this case **instead of B-frames or P-frames**.

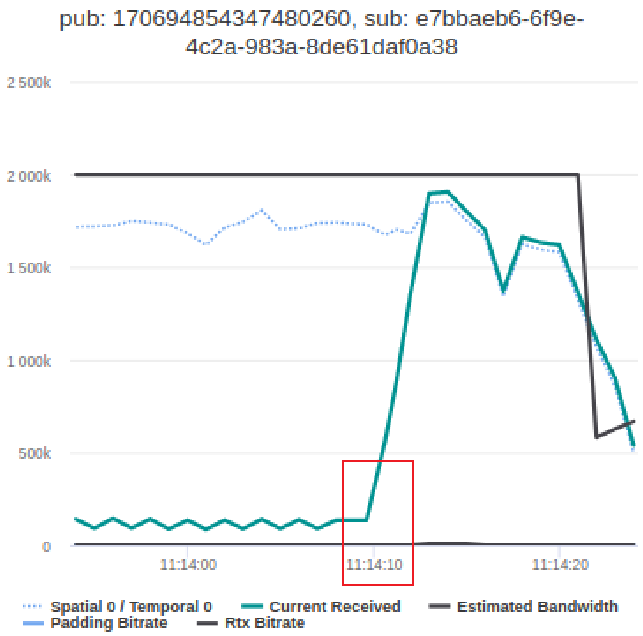




Quality_layers when `Toggle Slide Show Mode` Off

Highlighted in the below quality_layers is the change that occurred in the quality_layers at the moment when the `Toggle Slide Show Mode` was turned off.





Use Case: Real Videoconferencing Application

In the online class of 50 students, when the professor is delivering lecture, it is not required for the professor to see each student clearly and as a regular stream. It is sufficient for the professor to see a frame every few seconds just to make sure that the student is following the lecture. In such scenarios, this mode serves handy and efficient.

Conclusion

Licode and WebRTC concepts has been put into practise through this assignment. Licode is a open source WebRTC communications platform where as WebRTC is an open framework for the web that enables Real Time Communications in the browser. WebRTC is much more than building a simple video chat application. It's about finding ways to improve an experience between people over the Internet. Three clients were connected to the same room and it was possible to see the media flow and the changes in the quality of stream sent and received by each of the clients by modifying bandwidth, spatial and temporal layers.

References

[1] https://licode.readthedocs.io/en/stable/, "Licode Documentation"

[2] http://home.ku.edu.tr/~mtekalp/globecom2017.pdf, "WebRTC"