

Written report

For this Portal Assignment 4, Sofia decided to focus on the audio set up part and tried to learn more about routing going through the MOTU and RME fireface, Arvid worked on the spatialization/automation using Reaper and the IEM plugins, Kristian focused on the monitoring of Pd patches that each of us had to use in order to play together and create a collaborative composition, and I decided to focus on the JackTrip connections.

With Oliver in charge of the JackTrip connection for Team B, we tried different settings such as the peer2peer and hub connections in order to receive signals sent from the portal to the video room, and vice versa.

We started by setting the following parameters:

- 1) Enabling the public firewall and turning off the other ones (private and domain) on the LOLA computers
- 2) Launching the **Jack Audio Connection Toolkit** (QjackCtl) with the right setup (sampling rate & buffer size) being careful to set the same setup in JackTrip, Reaper and the RME fireface buffersize. For JackTrip, we could either specify these parameters through the Jack Audio Connection Toolkit or through the command line:

- `--srate` (default: 48000)
- `--bufsize` (default: 128)

- 3) After that, I used the Windows PowerShell, change the directory to the `jacktrip.exe` file location `cd "C:\Program Files\JACK2\jacktrip"` and create a connection by sending the IP address of the video room to Oliver in the Portal and using the following commands:

Get the internet protocol addresses in the command line: `ipconfig`

Portal IP address: 129.240.238.21

Videoroom IP address: 129.240.238.20

Peer2Peer

Jacktrip Server (running locally) to Client mode (Peer to peer).

- Joachim - Server: `./jacktrip.exe -s -n8`
- Oliver - Client: `./jacktrip.exe -c 129.240.238.20 -n8`

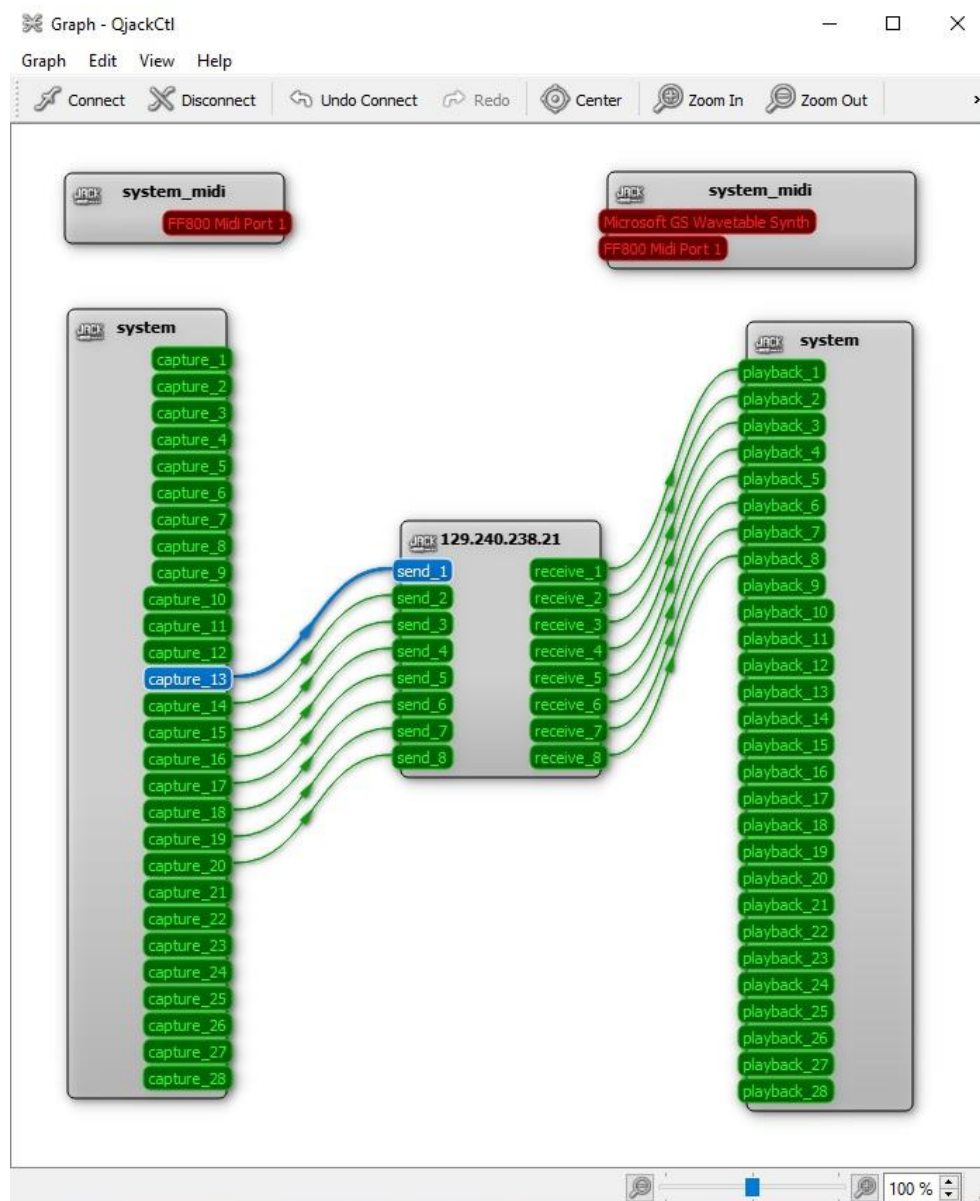
Hub

Jacktrip Hub mode (running locally) to multiple clients. Using the Jacktrip -S mode allows multiple clients to connect using the -C command (each additional client

needs to run a port offset in order to connect). But because we were only two, we didn't add any offset number.

- Joachim - Server: `./jacktrip.exe -S -n8`
- Oliver - Client: `./jacktrip.exe -C 129.240.238.20 -n8`

We tried several settings, as peer2peer or hub and had to specify the number of channels we wanted to send by adding the command `-n8` (8 channels). After that, we had to check the *QjackCtl Graph* option in order to see if the connections were enabled. The audio routing in the Video room was a bit tricky because we first had to send audio from the Reaper session on the MAC mini to the channels 17-24 however, the channels were routed differently in the LOLA PC. Therefore, to send them back to JackTrip we had to figure out that the correct routing was from 13 to 20.



We could also just run JackTrip using our system's default sound system instead of Jack Audio Connection Toolkit using the following command line:

```
./jacktrip.exe -S --rtaudio --srate 44100 --bufsize 128
```

Furthermore, we enabled arguments to display the IO statistics by specifying the interval in seconds (here 5 seconds) for the IO stat report and using the following command line:

```
Joachim - Server: ./jacktrip.exe -S -n8 --iostat 5
```

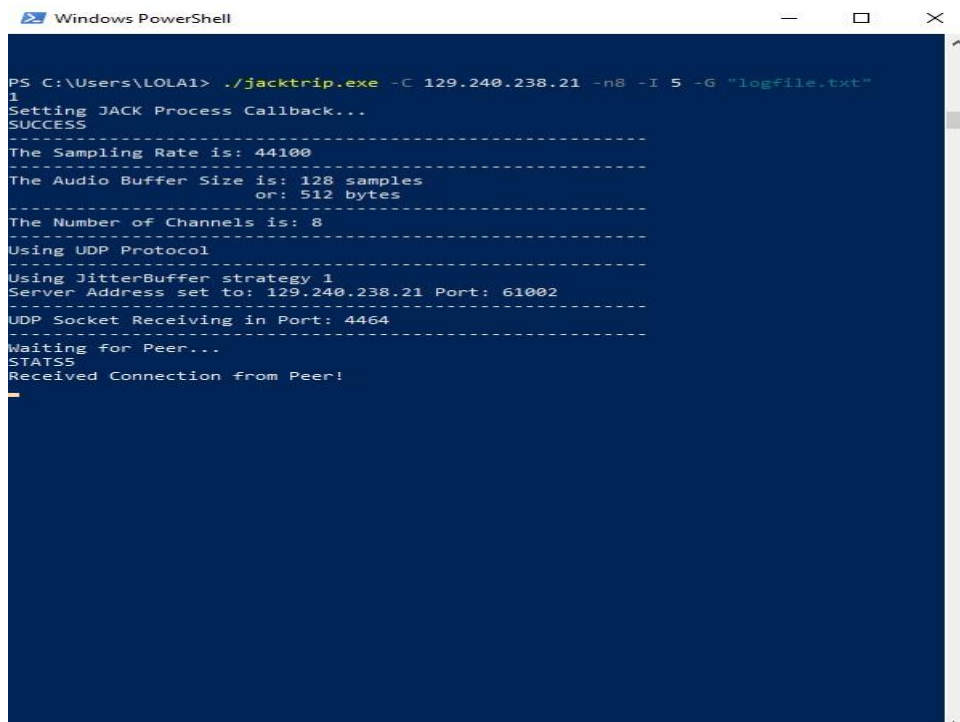
Additionally, we could save the statistics report to a `logfile.txt` by doing the following

```
Joachim - Server: ./jacktrip.exe -S -n8 --iostat 5 --iostatlog  
"logfile.txt"
```

On the other hand, another handy option was to test the audio delay statistics which printed the roundtrip audio delay statistics for the highest-numbered audio channel every `printIntervalSec` seconds, including an ASCII latency histogram if `printIntervalSec` is 1.0 or more. We could access this by using the following command (5 seconds for the interval):

```
Joachim - Server: ./jacktrip.exe -S --rtaudio --examine-audio-delay 5
```

Finally, we received a **connection from Peer!** validation in JackTrip indicating that the connection was enabled. We could therefore send the 8 audio channels encoded in 3rd order ambisonics in order to be decoded through the [AllRAD IEM decoder](#) plug-in which enabled us to decode the signals for any desired loudspeaker layout using the AllRAD approach.



```
Windows PowerShell

PS C:\Users\LOLA1> ./jacktrip.exe -C 129.240.238.21 -n8 -I 5 -G "logfile.txt"
1
Setting JACK Process Callback...
SUCCESS
-----
The Sampling Rate is: 44100
-----
The Audio Buffer Size is: 128 samples
                        or: 512 bytes
-----
The Number of Channels is: 8
-----
Using UDP Protocol
-----
Using JitterBuffer strategy 1
Server Address set to: 129.240.238.21 Port: 61002
-----
UDP Socket Receiving in Port: 4464
-----
Waiting for Peer...
STATS
Received Connection from Peer!
```

