

JACKTRIP AND YOU

with Jackson, Mari, Iggy and Ulrik

IS A NETWORKED MUSIC PERFORMANCE (NMP) POSSIBLE?

Yes, and has been for a while - but in the past composed mostly of OSC messages and sonification techniques.

Only recently has the live transfer of real-time sound been possible and even this was limited by WLAN protocols, consumer hardware, and, of course, a fast internet connection (Internet2)

And even if the technical bandwidth is possible we still have to consider latency of all of these components. This led to the development of NMP techniques that took latency into account (NINJAM)

TELEMATIC PERFORMANCES

A brief history of JackTrip:

- Made by Chris Chafe
- Developed in 2000/2001 for institutional networks
- Over the next decade, tested with musicians and artists in live performances
- However, even up to their major performance, *Pacific Rim of Wire* (2008), a concert between Beijing and Stanford, there were many issues (RTT of 220ms!)

in Europe and ARNET in Australia. It also requires technical expertise to install and run the software, which for this research was deemed prohibitive for performers who otherwise fulfilled the participant criteria (see Chap. 4). However, as Bassist Mark Dresser notes, the team cooperation required to set up a telematic performance with JackTrip is paradoxically one of the most rewarding aspects of using it (Dresser [2008](#)).

Mills R. (2019) Telematics, Art and the Evolution of Networked Music Performance. In: Tele-Improvisation: Intercultural Interaction in the Online Global Music Jam Session. Springer Series on Cultural Computing. Springer, Cham.
https://doi-org.ezproxy.uio.no/10.1007/978-3-319-71039-6_2

TO SUM UP...

However, the performance demonstrated the inherent difficulties of streaming multi-site live audio and video, as well as how to ameliorate technical problems associated with it, e.g., network latency, video encoding and streaming high amount of data required for transmission. It also showed how beholden, networked performers are to the vagaries of speed and bandwidth of multiple networks, network architectures, and differences in audio-visual streaming applications. While these problems are lessened using high-speed research networks, they remain a significant issue for audio-visual streaming of live NMP.

A POST-JACKTRIP WORLD?

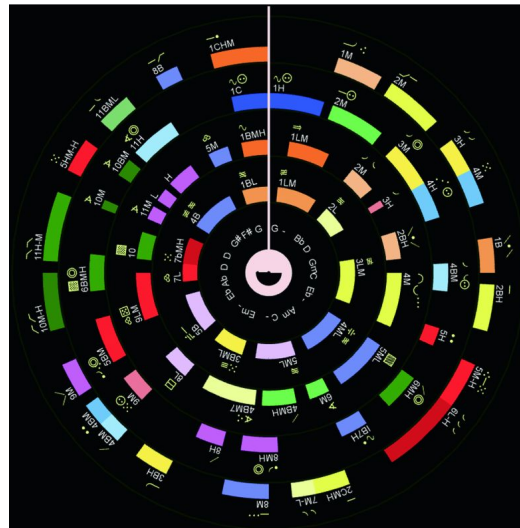
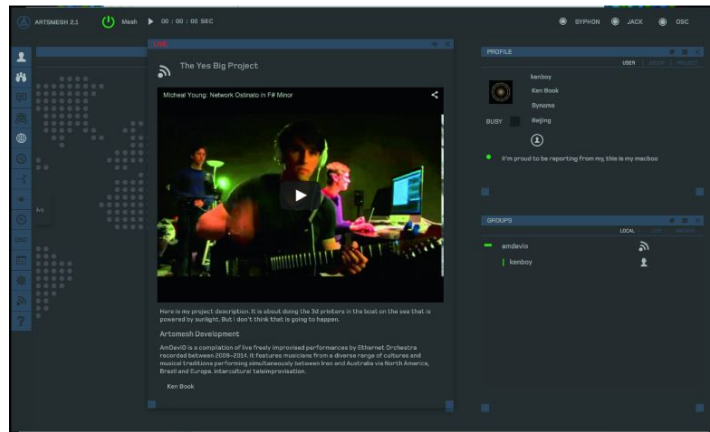
After the rise of JackTrip, a number of people wanted to modernize the application - but many real-time audio projects are still based on JT

- Ken Fields' ArtsMesh
- Ian Whalley's Graphic Networked Music Interactive Scoring System

Non-JackTrip projects? (WebRTC?)

- Jamulus
- JamKazam
- SoundJack
- Sofasession
- NinJam
- Jammr
- eJamming
- and the list goes on...

[A descriptive blog by Anna Xambó](#)



A BRIEF ON JACKTRIP

```
JackTrip: A System for High-Quality Audio Network Performance
over the Internet
Copyright (c) 2008-2015 Juan-Pablo Caceres, Chris Chafe.
SoundWIRE group at CCRMA, Stanford University
VERSION: 1.1
```

Jacktrip is a command line program developed at CCRMA at Stanford for low-latency audio transmission between multiple clients. Jacktrip employs the Jack protocol, a high definition API for audio communication between applications. Jacktrip appears to be one of the few viable tools that, in theory, works on every operating system (Windows, macOS, Linux) which makes it a potentially worthwhile investment from the angle of usability and access alone.

It is also both the glimmer of hope for lossless musicking and the bane of our collective existence.

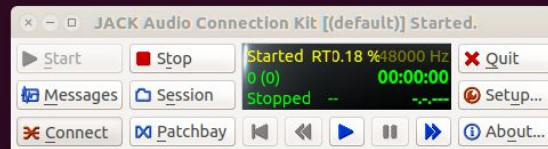
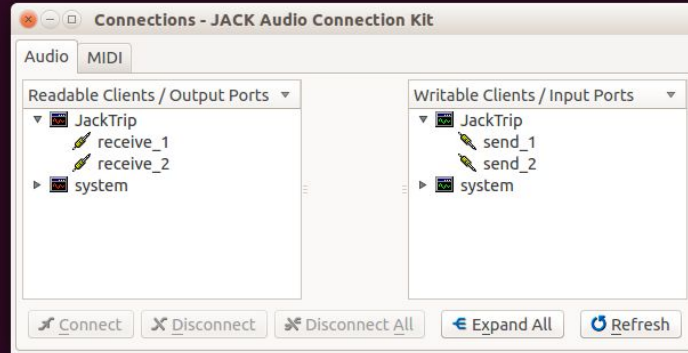
```
jacktrip -C 129.241.8.239
```

 Hub server

```
jacktrip -c 129.241.8.239
```

 Normal pair-to-pair server

```
~ $ jacktrip -s
SETTING ALL PORTS
Setting JACK Process Callback...
SUCCESS
-----
The Sampling Rate is: 48000
-----
The Audio Buffer Size is: 512 samples
                        or: 2048 bytes
-----
The Number of Channels is: 2
-----
Using UDP Protocol
-----
Waiting for Connection From Client...
█
```



WHAT IS QJACKTRIP

In the slightly bewildering, wonky world of JackTrip there is a new entry in the field, **QJackTrip**, who offers a more UI friendly take on JackTrip (built in Qt), therefore no terminal required!

You can find downloads to it here.

<https://www.psi-borg.org/other-dev.html>

<https://github.com/jacktrip/jacktrip/tree/qjacktrip> - Repo

Scrolling down on the main page provides packages for Windows and OSX (Linux is a bit trickier). These listed packages include the Jack driver - nice for us!

1. Installation

QJackTrip runs on Windows, OS X, and Linux, and the exact installation process will depend on your operating system. For Windows and Mac, the easiest (and recommended) way of going about it is to install the prebuilt binaries provided on this page. For Linux, you'll need to build from source.

Important: The OS X binaries provided here will only work on High Sierra (10.13) and later. If you're running an earlier version of OS X you'll either need to update your operating system or build the software (with the exception of the Jack server package) yourself. (Refer to the Linux instructions below for more detail on that.)

OS X

Download and install the following packages:

- [The Jack Server](#) – You'll need to restart your computer after running the installer package.
- [QJackCtl](#) – This replaces the very outdated version shipped with the JackTrip package. (Optional, but highly recommended.)
- [QJackTrip](#)

Windows

Download and install the following packages:

- [The ASIO4ALL driver](#) – This makes it easier to use Jack on Windows.
- [The Jack Server](#)
- [QJackTrip](#) – The installer is a little basic at the moment and doesn't give much user feedback. If you see a QJackTrip folder in the start menu once it's finished then you'll know it's done the job.

Linux (Building from source)

SETTING UP JACK (AND QJACKCTL)

The unique audio driver/server that is employed to achieve this low-latency communication is Jack. It can be installed, for our purposes, using the links found on the QJackTrip page we can install the Jack server.

Note: For Mac users, you may need to right click on the .pkg file and “Open” to bypass MacOS’s security.

Afterwards, we are required to reboot. Then we can install a utility QJackCtl (yes, also built in Qt) so we can control the Jack server manually without having to go into the terminal.

PORT FORWARDING (ONLY IF HOSTING)

More details on psi-borg.org - we will just try to connect to a pre-existing server

Port Forwarding / Port Triggering

Please select the service type.

Port Forwarding
 Port Triggering

Service Name Internal IP address

FTP 192 . 168 . 1 . + Add

#	Service Name	External Starting Port	Internal Starting Port	Internal IP address
<input type="radio"/> 1	JackTrip	4464-4471	4464-4471	192.168.1.4

SETTING UP QJACKCTL

Finally, the last piece is QJackCtl, a GUI for the jack command line interface built in Qt. It will serve as our utility to finally connect with a peer using their IP.

The download link and files are located on the same page.



QJACKTRIP

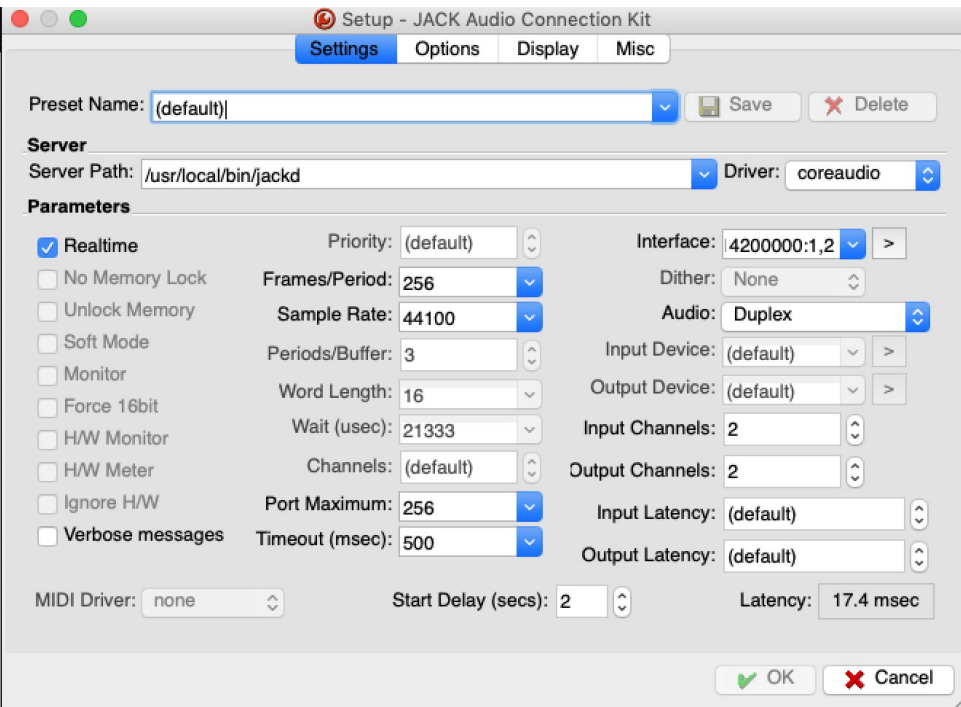
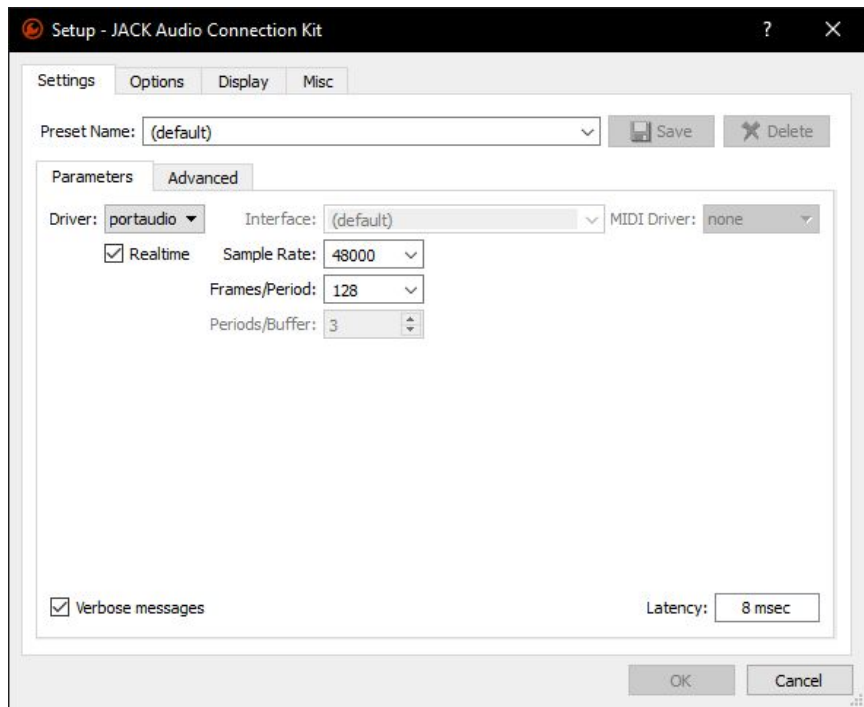
So - why couldn't this QJackCtl and QJackTrip just be bundled as a single app?
Great question!

Moving on:

Attempting to connect to the server (hosted by Anders & Co.) through QJackCtl requires a little fine tuning on our side.

1. Sample rate and buffer size of all participants (and servers) must be identical - in our case we change it to 48Hz and 128 frames/sec
2. Make sure there are no applications using port 4464 (default)
3. Plug an ethernet cable in! Somewhat essential if we are attempting to create a bit perfect stream (though it **is** possible over WiFi - sounds like crap)

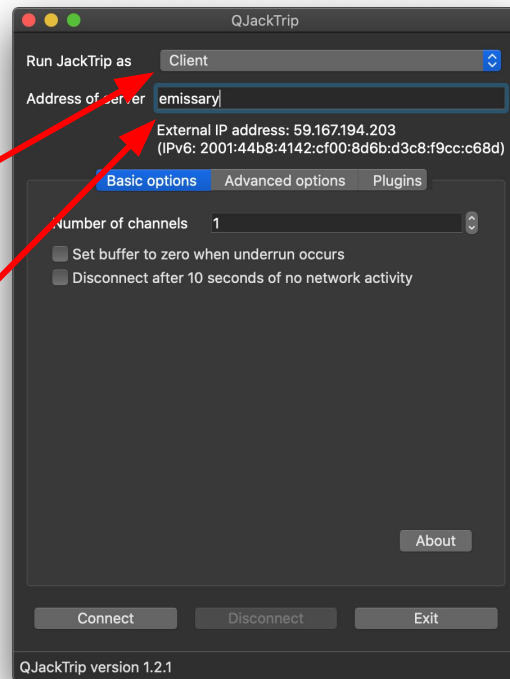
QJACKCTL SETTINGS



CONNECTING

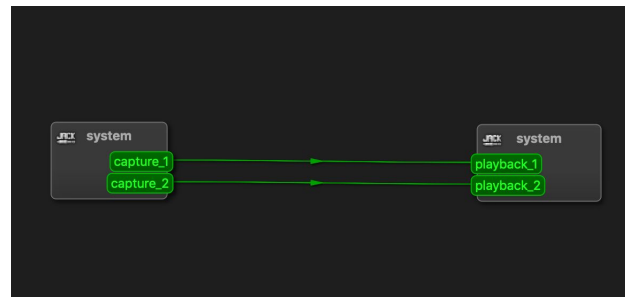
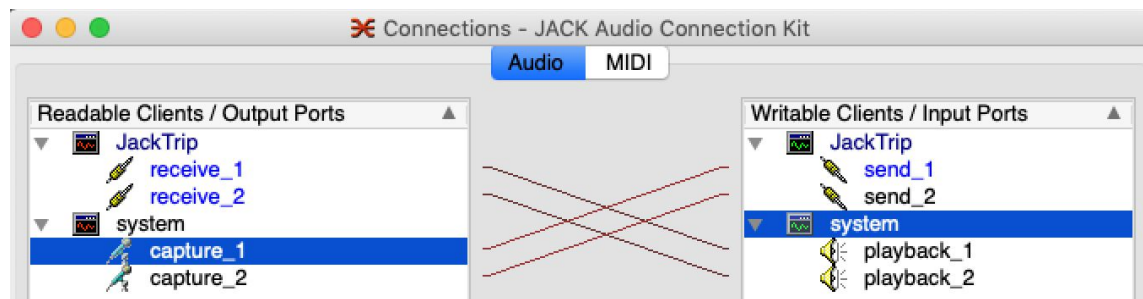
Depending on whether you're connecting with one peer (P2P) or connecting to a live instance of a server (Hub) you can select your type

Then just type in the IP, make sure Jack is running and wait to be connected



ONCE CONNECTED - TIME TO PATCH!

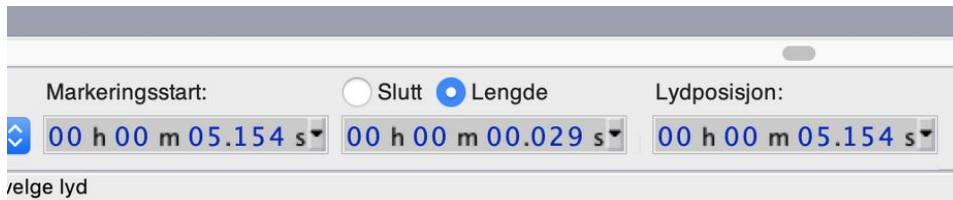
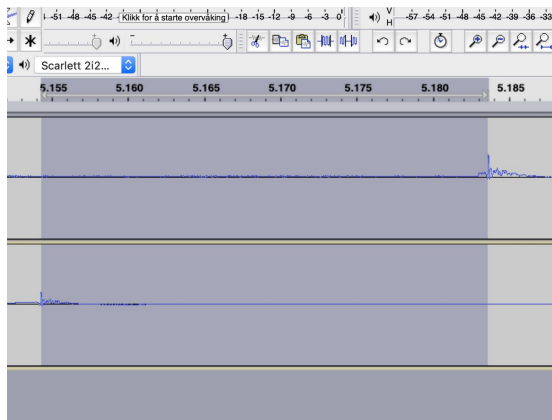
Yes, even once we are connected there's still more work to do. Patching the microphone inputs to the server out, and the server outs to your ins... Here's a picture.



HOW DOES IT PERFORM?

Well, if anyone is on WiFi, the experiences is quite hit or miss as network bandwidth is generally distributed across the computer - meaning JackTrip drops out.

However, if all participants are on ethernet the results show that we can achieve round-trip latency (from your voice back to you) sub 30ms!



HOW DOES IT SOUND?

Again, if both parties are on ethernet cables the sound quality is at the highest bitrate possible and sounds incredibly transparent

- No compression, lossless quality
- Subject to dropped packets
- Experience may vary very across operating systems
(Windows audio might suffer if on ASIO - see [FlexASIO](#))

WHAT COULD GO WRONG?

Quite a lot - ranging from client to host issues. Here are the the top mishaps during our experience.

- Cannot get Jack to run (from QJackCtl)
 - Configuring audio driver (for Windows, Linux)
- Jack is running - but incorrectly
 - It is not obvious when you've assigned Jack the correct settings (sound driver, etc. especially on Windows)
- Building on Linux - tricky, audio devices may not have native support
- QJackTrip is not running on any old OS X versions (E.g. El Capitan)
- Windows is a poor system to working with real-time audio in general