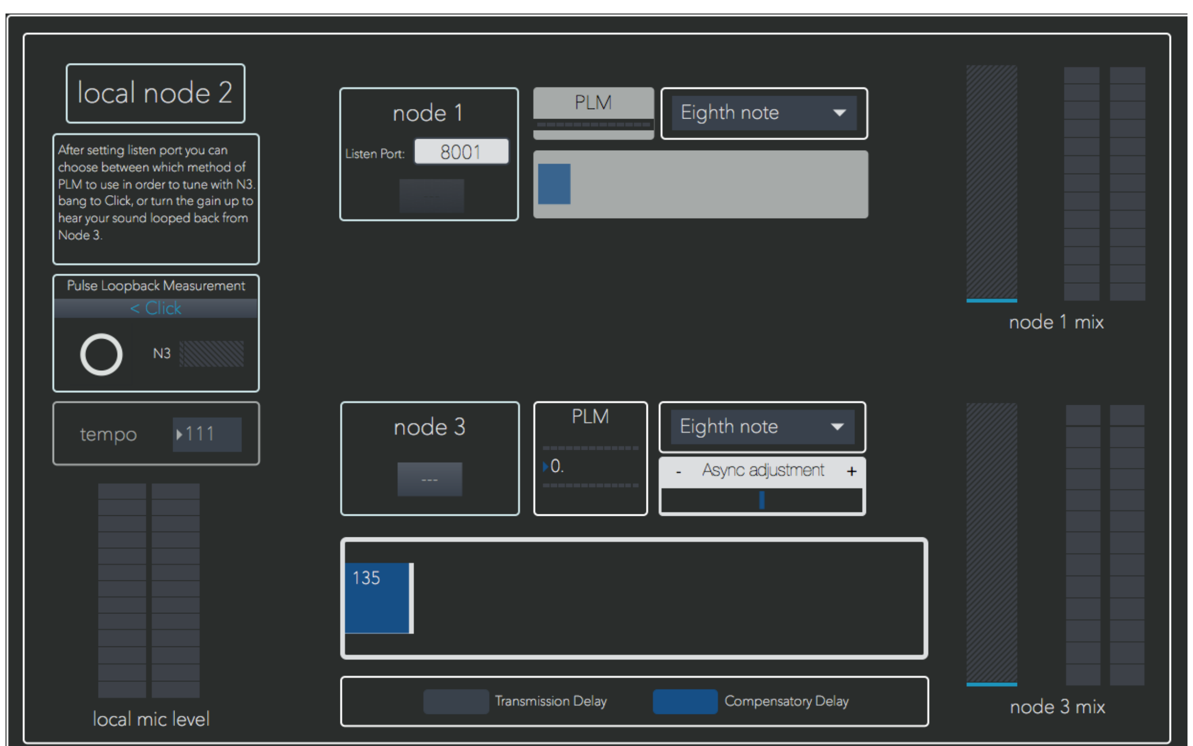
Toporhythm Software Tutorial

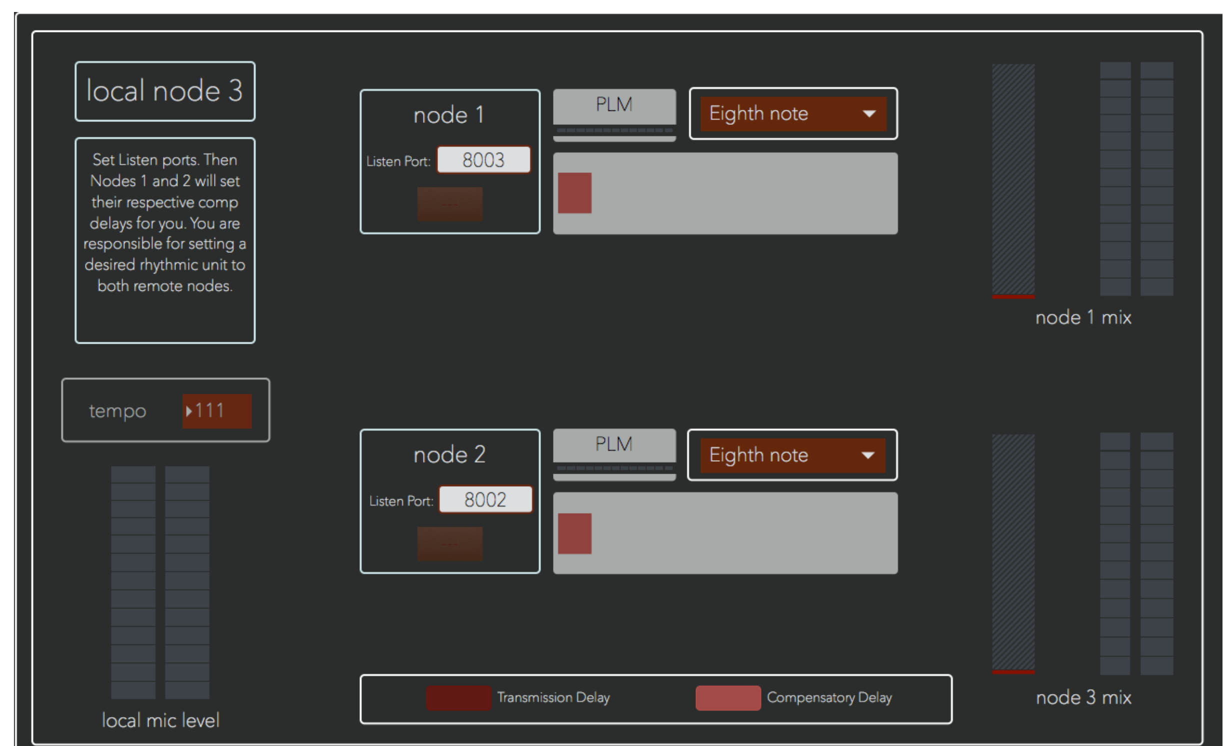
for use with the latency calibration Max/MSP patches

Introduction

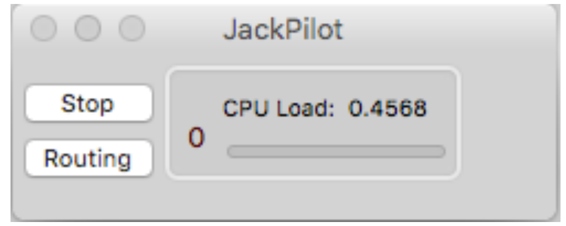
This tutorial is meant to aid in using the Max/MSP patch built in conjunction with this research. Audio routing and connecting to a remote computer are done in separate software (such as Artsmesh, qjackctl, or JackPilot) and will not be described here. However, instructions for routing audio in and out of Max/MSP will be provided.

The primary function of this software is to calibrate the latency between each node (or location) in a network music performance to a desired rhythmic unit of time at a chosen tempo. To do this, the software measures the latency in the network and applies additional delay to each node's audio signal by the appropriate times (in milliseconds). Each node requires a unique version of the patch because not all nodes need to control every variable in the calculation. Node 1 has the most controls and we will mainly focus on Node 1's functions. But first, an overview of the audio routing is necessary.

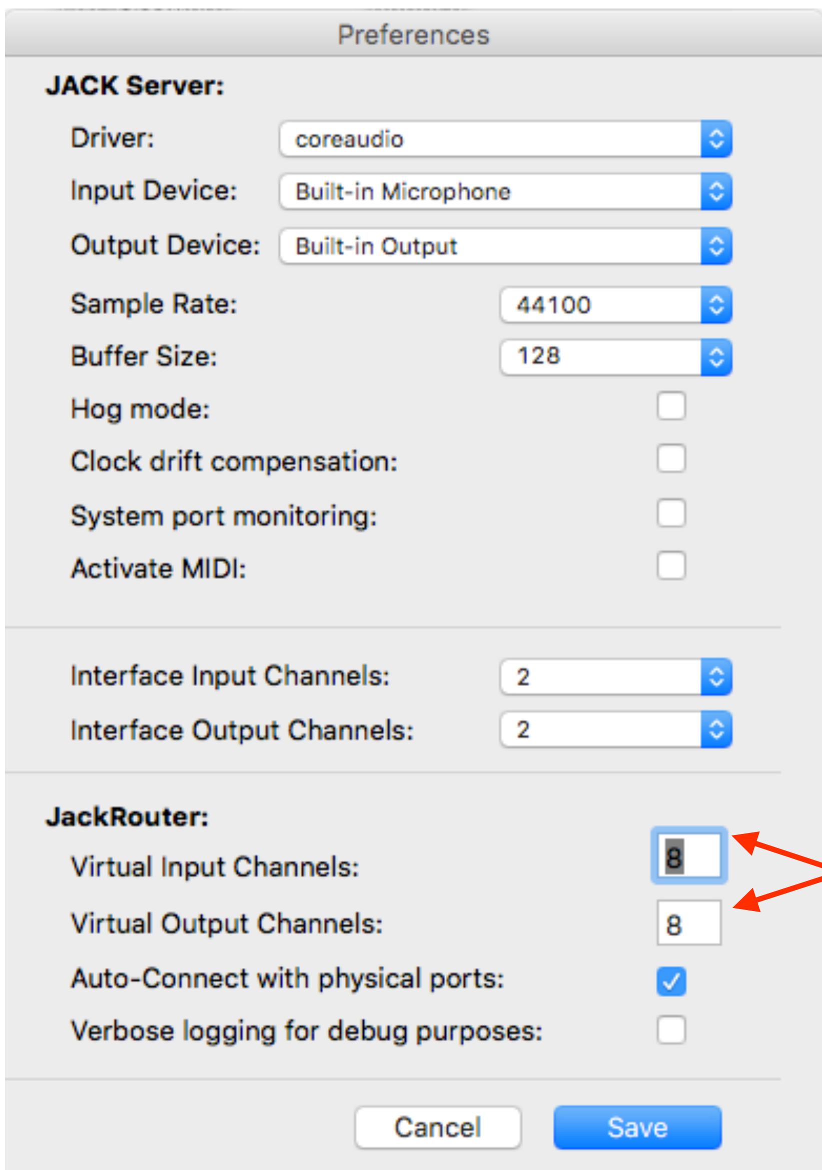




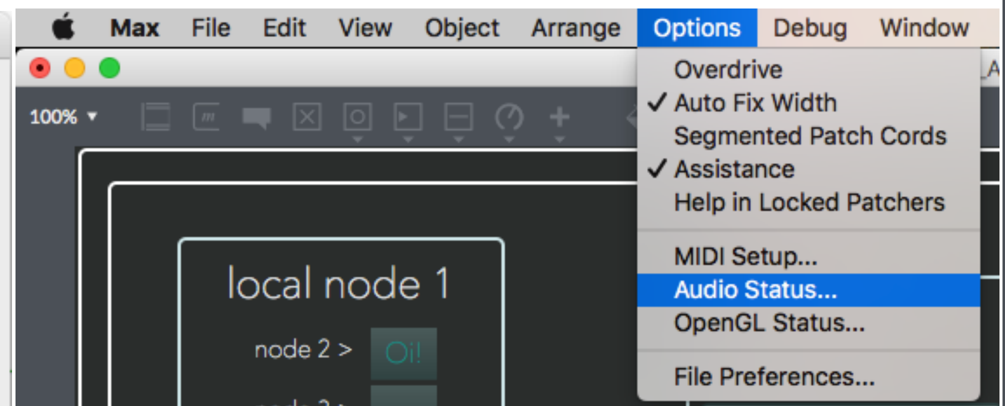
First things first…



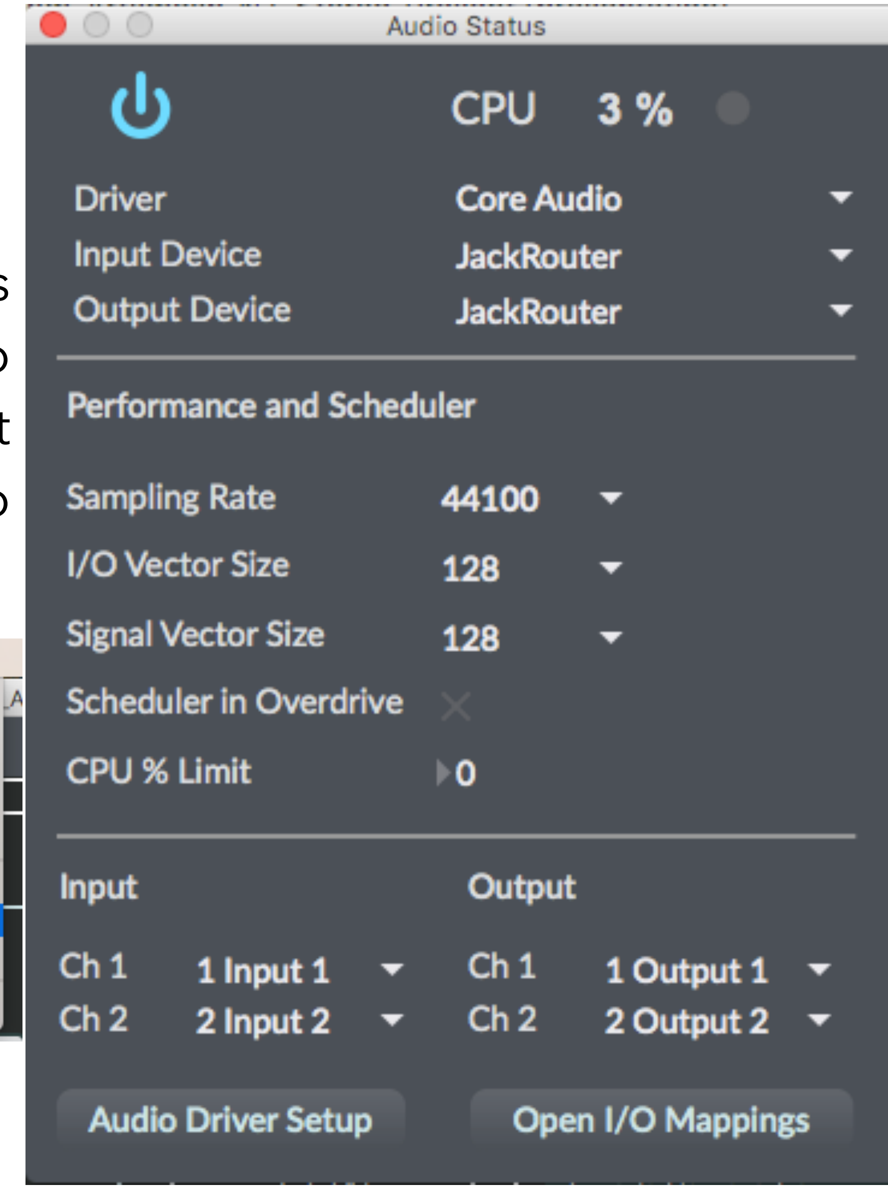
Again, this tutorial will not be addressing how to make audio connections in software such as qjackctl or Artsmesh, but will be using JACK audio server (JackRouter) to route audio to and from Max/MSP. Therefore, it should be addressed that the JACK server needs to be running prior to opening Max/MSP. This is demonstrated here using JackPilot.



make sure you have 8 virtual input and output channels, you'll need these later.



Once the JACK Server is running, open the Max patch, navigate to the "Options" tab, and select "Audio Status…"



The above window will appear. Here, you set JackRouter as your Input and Output device. Then, choose all of the desired preferences, such as Sampling Rate and Vector Sizes, and make sure that they match those set in the JACK preferences. Then, turn on the audio in the top left.

Routing Instructions

Below are the routing instructions for stereo audio to and from every node in a tri-located network music performance using this patch. It includes the PLM (Pulse Loopback Measurement) channels, which will be addressed later in this tutorial. This routing can be done in Artsmesh, qjackctl, or JackPilot.

Node 1

Routing (done in Artsmesh):

Max Recieves (adc)
1 - system send 1
2 - system send 2
3 - node 2's send 1
4 - node 2's send 2
5 - node 3's send 1
6 - node 3's send 2
7 - node 2's send 3*
8 - node 3's send 3*

Max Sends (dac)
1 - system recieve 1
2 - system recieve 2
3 - node 2's recieve 1
4 - node 2's recieve 2
5 - node 3's recieve 1
6 - node 3's recieve 2
7 - node 2's recieve 3*
8 - node 3's recieve 3*
(*=PLM channels)

Node 2

Routing (done in Artsmesh):

Max Recieves (adc)
1 - system send 1
2 - system send 2
3 - node 1's send 1
4 - node 1's send 2
5 - node 3's send 1
6 - node 3's send 2
7 - node 1's send 3*
8 - node 3's send 3*

Max Sends (dac)
1 - system recieve 1
2 - system recieve 2
3 - node 1's recieve 1
4 - node 1's recieve 2
5 - node 3's recieve 1
6 - node 3's recieve 2
7 - node 1's recieve 3*
8 - node 3's recieve 3*
(*=PLM channels)

Node 3

Routing (done in Artsmesh):

Max Recieves (adc)
1 - system send 1
2 - system send 2
3 - node 1's send 1
4 - node 1's send 2
5 - node 2's send 1
6 - node 2's send 2
7 - node 1's send 3*
8 - node 2's send 3*

Max Sends (dac)
1 - system recieve 1
2 - system recieve 2
3 - node 1's recieve 1
4 - node 1's recieve 2
5 - node 2's recieve 1
6 - node 2's recieve 2
7 - node 1's recieve 3*
8 - node 2's recieve 3*
(*=PLM channels)

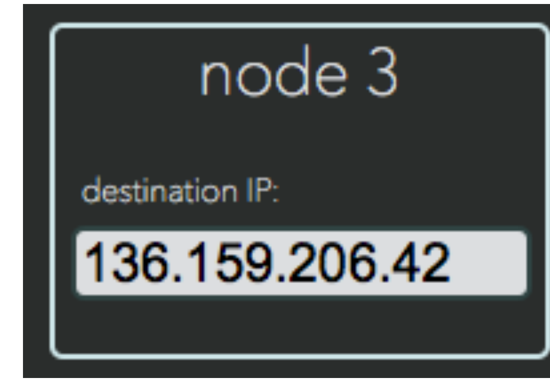
- Latency Calibration

- UDP functionality

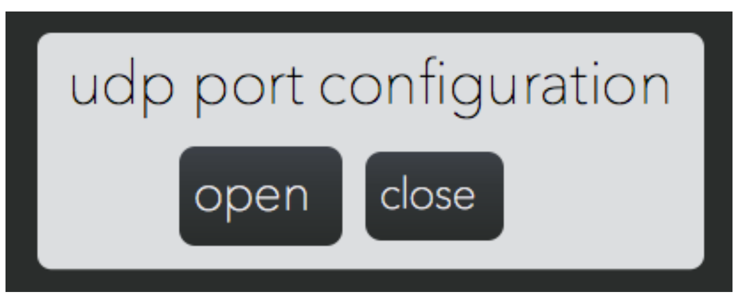
Two main components of the patch are its use of UDP for sending and receiving controls and calculating the compensatory delay

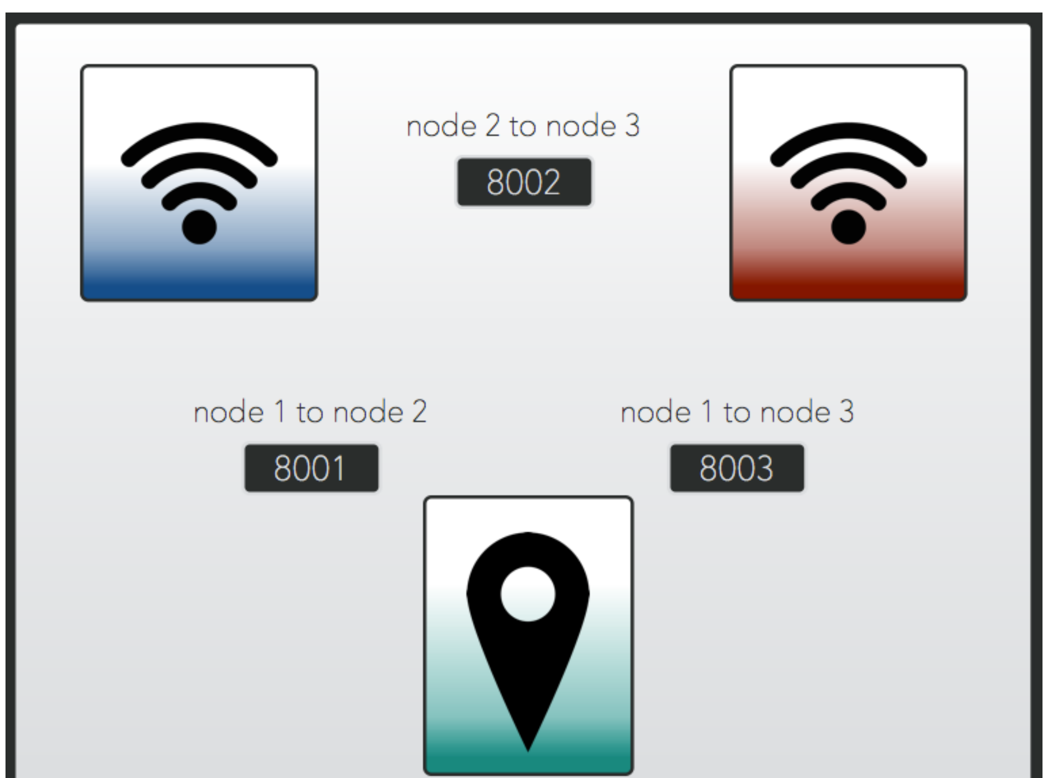
UDP Functionality

In network music, communication can be difficult. In most cases, an external tool (such as Skype) is used to roughly communicate prior to connecting audio. This is done in order to ensure that everyone's settings are the same and there won't be any conflicts with sampling rate or vector sizes. Aligning the latency using this software presents another potential conflict, if each location can control their tempo and compensatory delay independently, there is more opportunity for miscalculation, resulting in unwanted temporal relationships. Therefore, a sort of hierarchy is built into the system that allows Node 1 to control the tempo for all nodes, and to set the compensatory delay for two of the three connections. The remote nodes need only to set the appropriate udp ports.

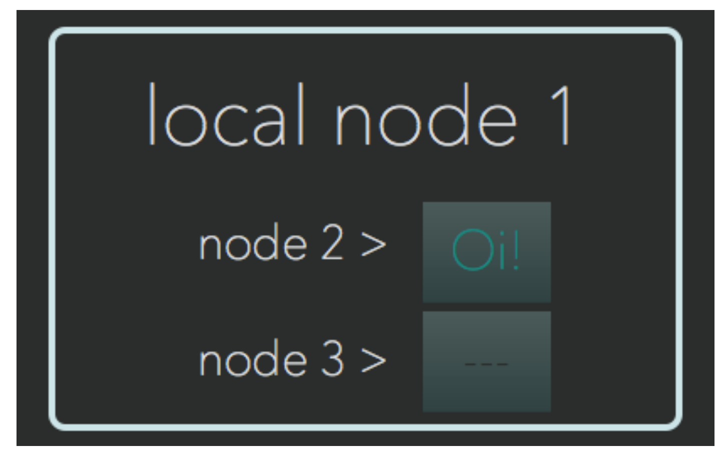


1) Node 1 enters the IP addresses of the remote nodes.

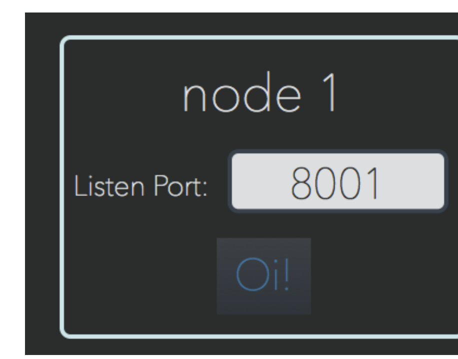




2) Node 1 opens the "udp port configuration" window and sets all the udp ports between each node.

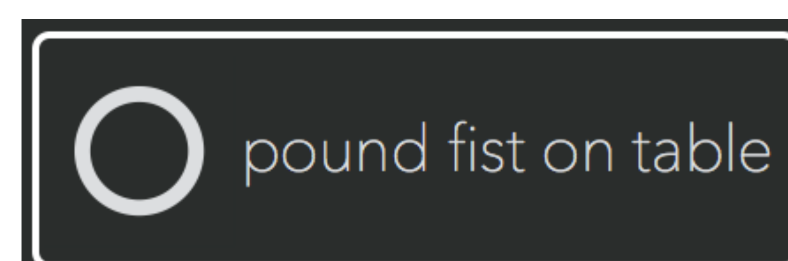


3) Node 1 can check the connection with the remote nodes by digitally yelling, "Oi!"



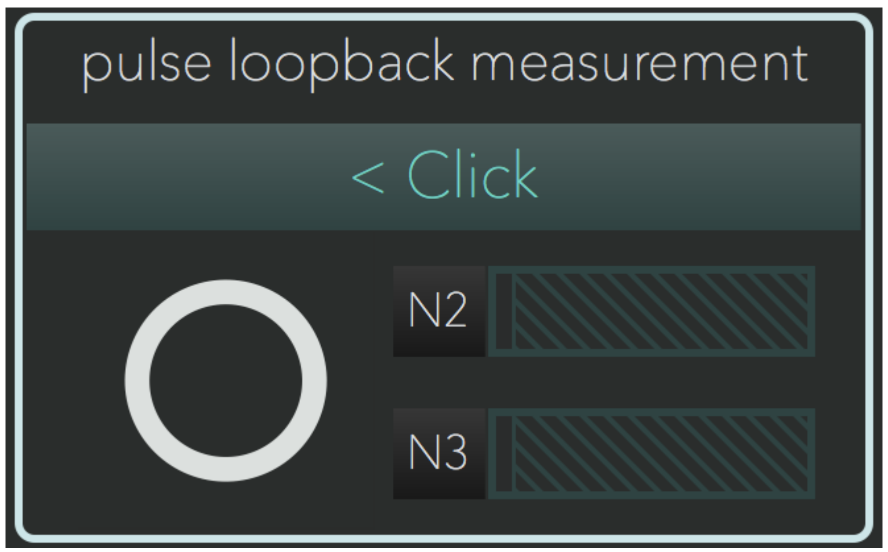
* If the remote node hears it, we've got a connection

** If not, there may have been a number entered out of order somewhere and digitally pounding your fist on the table can help.

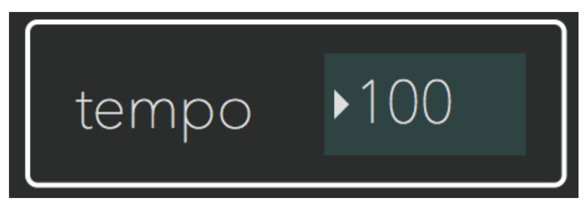


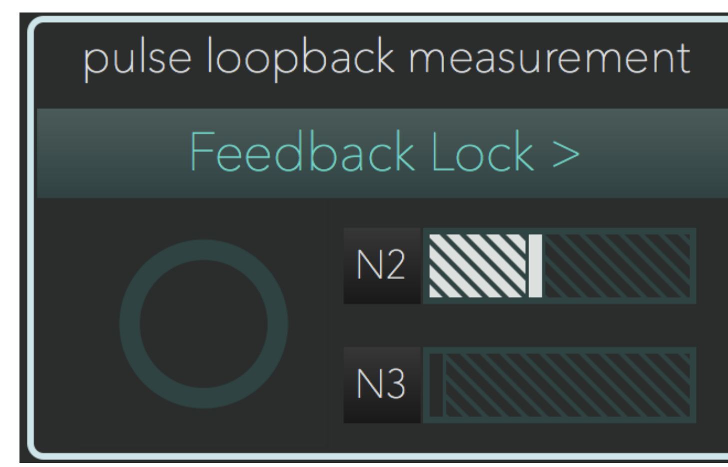
Latency Calibration

There are two ways to calibrate the latency to a rhythmic value using the PLM (Pulse Loopback Measurement), the *Click* and the *Feedback Lock*. I find it works best to use a combination of both methods in order to get the tightest sync.



Once we have a PLM number in place, choose a desired rhythmic value and a tempo and the patch will run the calculation. However, this calculation does not consider the latency added through A-to-D and D-to-A conversion by your audio interface and speakers. This is where Feedback Locking comes in handy.





If the routing has been done correctly, hitting the button in the pulse loopback measurement section while in *Click* mode, should send a click to all nodes, which is routed directly back. The time elapsed during the click's journey is gives us the PLM.

In Feedback Lock mode, your audio signal travels the same route that the click travelled before, looping back immediately upon arriving at the remote node. However, it is now delayed locally according to the calculation made using your chosen rhythmic value and tempo.

Latency Calibration
continued

Using *Feedback Lock* mode is a good way to aurally confirm everything is working. Change the rhythmic value while making a sound and you will hear the patch make the timing adjustment. There is still one problem though, at this point we are only considering one A-to-D and D-to-A conversion. In order to integrate all levels of latency into the compensation we need to "temporally tune" with the remote node. To do this we make small adjustments with the Async adjustment.

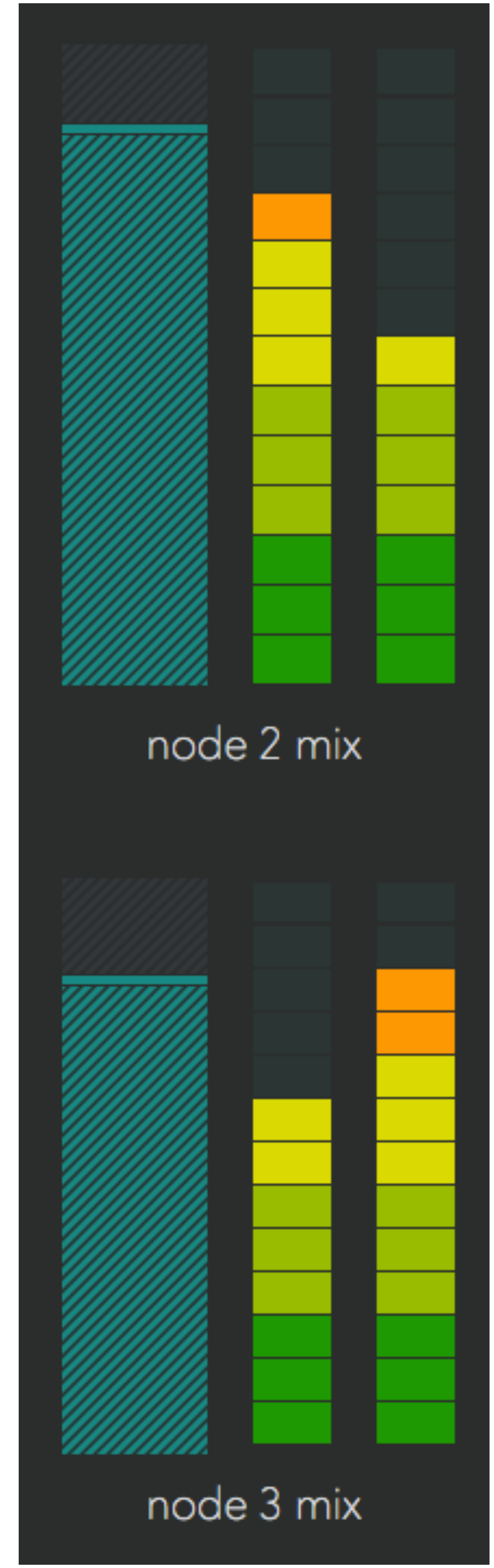
I find the most effective way to "temporally tune" is with the aid of a metronome. Have the remote node point their microphone toward their speaker and bring their gain up for Node 1. At Node 1, set your metronome to the rhythmic value and tempo you've chosen and play it into your mic. Turn up the gain to the remote node and if your unit is set to a quarter note, you should hear the metronome click coming back a quarter note later. The local click from the metronome should fall exactly in time with the sound returning through the speakers. If it is slightly ahead or slightly behind, use the Async adjustment to bring it into tune.

* This should be done with one node at a time as the extra feedback echoes from a third node can cause confusion.

Play music…toporhythmically

The software is designed so that Node 1 can take care of most of the tuning adjustments and other parameters. However, Node 2 must follow the same procedure to tune with Node 3. Also, it is each node's responsibility to choose a desired rhythmic value. This means asymmetrical rhythmic relationships are possible (eighth note from Node 1 to Node 2, but a quarter note from Node 2 back to Node 1). However, these should be chosen after having tuned using a symmetrical rhythmic relationship.

Once all nodes are in temporal tune, the only step left is to set levels and play. The software allows you to set a level for each remote node's incoming signal and it is advised to consider your gain-staging along the way. Gain levels for your microphone should be robust, leaving plenty of headroom in the Max patch.



Useful Terms

Listed below are some common terms found in the practice of network music performance

Audio

Sampling Rate:

44.1kHz is CD-quality, but 48kHz is also common

Bit Depth:

16 bit is CD-quality, but 24 bit is also common

Vector Size (Buffer Size):

a larger vector size (2048) means more latency but less CPU usage, a smaller vector size (64) may cause some interfaces to click but results in very low latency

Network

IP address:

Internet Protocol address used for computers to find each other over a network. IPv4 example: 127.255.255.255
IPv6 example: 2001:0DB8:AC10:FE01::4

Bandwidth:

The more the merrier, but stereo audio at 44.1kHz needs at very least 2Mbps upload and download speed