



Berklee
Global Initiatives

Remote Performance Experiences

Jason Camelio, Ivanna Cuesta, Jim Odgren, Ray Seol

Berklee College of Music



The Story of Remote Performance Experiences

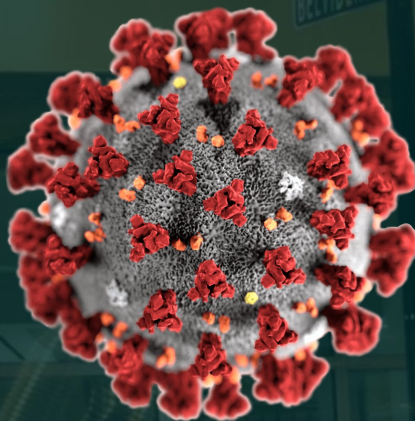
- Challenge
- Research - Experiment
- Collaboration
- Feedback



Berklee
Global Initiatives

Challenge - COVID 19

How can musicians play together remotely?

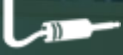




Berklee
Global Initiatives

Research

Look for Free, Open Source, and Easy to Use Software

jamkazam 

 Kompoz

SoundJack
a realtime communication solution



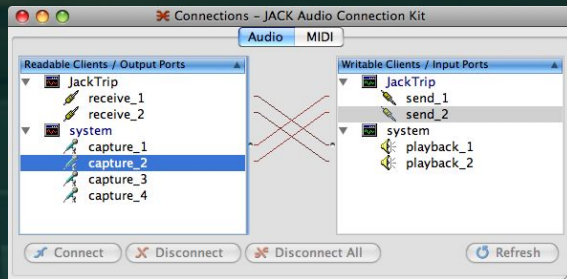
Cockos Ninjam



REAPER
Digital Audio Workstation



JamTaba



Jack Audio



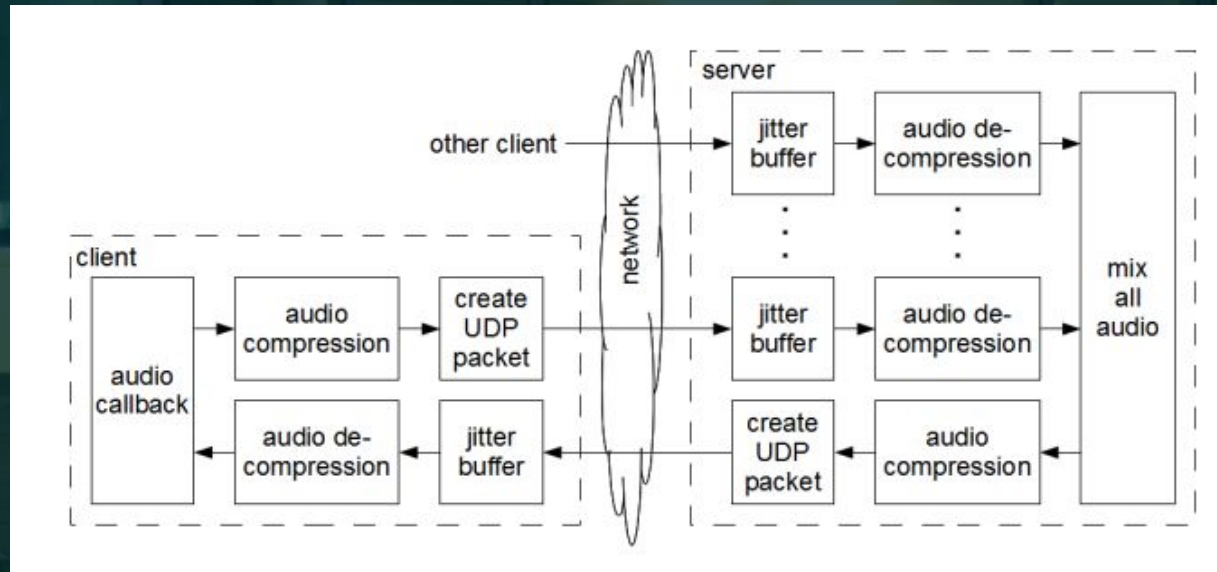
Base Technology:

- Jamulus by Volker Fischer - "a client-server system (no special synchronization is required). All streams are mixed at the server and each client receives the same mix." - Real Time (since 2006)
- Novel Intervallic Network Jamming Architecture for Music (NINJAM) by Cockos - <https://www.cockos.com/ninjam/> - Staggered Real Time (Since 2005)
- Linux Server
- Amazon Web Services

Reference

<https://jamulus.io/PerformingBandRehearsalsontheInternetWithJamulus.pdf>

Server Diagram



Reference:

<https://jamulus.io/PerformingBandRehearsalsontheInternetWithJamulus.pdf>



Berklee
Global Initiatives



Berklee
Global Initiatives

Collaboration

How to build the platform together.

Ivanna Cuesta: Drums Student, Berklee

Jim Odgren: Professor, Woodwind, Berklee

Jason Camelio: Assistant Vice President, Berklee Global Initiatives

Ray Seol: Assistant Director, Berklee Global Initiatives



Basic System Requirements

- Laptop or Desktop Computer (Mac or PC)
- Ethernet (preferred) or WiFi connection
- Audio Interface (e.g. Focusrite, Apogee, Yeti)
- Your Instrument (a mic or line - extra mic for talkback)
- Headphone (not earbuds)
- Software from <https://jamulus.io/>
- Korean Sweet Potatoes (if you hungry)



Set Up Tips

- Laptop or Desktop Computer (Mac or PC)
- Ethernet (preferred) or WiFi connection
- Audio Interface (e.g. Focusrite, Apogee, Yeti)
- Your Instrument (a mic or line - extra mic for talkback)
- Headphone (not earbuds)
- Software from <https://jamulus.io/>
- Korean Sweet Potatoes (if you hungry)



Berklee
Global Initiatives

Software Overview

<https://jamulus.io/>



A screenshot of a digital audio workstation (DAW) mixer interface. The interface is dark grey and features several control elements. On the left, there are buttons for "Delay" and "Buffers", both with green indicator lights. Below these are "L" and "R" channel indicators. A yellow circle highlights a settings menu containing "Settings", "Chat", "Mute Myself", and "Disconnect" buttons. The main mixer area is titled "Personal Mix at: 3: Snayler's AWS" and contains six mixer channels. Each channel has a "PAN" knob, a volume fader, and three buttons labeled "GRP", "MUTE", and "SOLO". Below the mixer are six participant cards: "CG" (UK flag), "Giampiero" (Italy flag), "Serc-bas s" (UK flag), "Smiff" (UK flag), "Tino" (UK flag), and "Wanderso n" (Brazil flag). A "Reverb" control is visible on the far left of the mixer area.



Settings

Soundcard

Device
in: Duet USB/out: Duet USB

Input Channel Mapping

L 1: Mic 1
R 2: Inst 2

Output Channel Mapping

L 1: Line 1
R 2: Line 2

Enable Small Network Buffers

Buffer Delay

2.67 ms (64)
 5.33 ms (128, preferred)
 10.67 ms (256)

Jitter Buffer

Auto

Local Server
Size: 6 Size: 4

Misc

Audio Channels Mono

Audio Quality Normal

New Client Level 100 %

Skin Fancy

Language American English (en)

Custom Central Server Address:

Audio Stream Rate ---

Ping Time ---

Overall Delay ---



Connection Setup

List: Default Filter: Type # for occupied servers Show All Musicians

Server Name	Ping Time	Musicians	Location
Henry's Jamulus Serv	12 ms	0/10	New York, United States
yourServerName	13 ms	0/10	yourCity
JapaneseKaraokeUSA	17 ms	0/10	Massachusetts, United States
ip-10-0-0-158	18 ms	0/10	
WMChoirs	18 ms	0/10	Williamsburg, United States
JamNastyServer	18 ms	0/10	DC, Afghanistan
TSO-Combo	19 ms	0/10	Jacksonville Beach F, United Sta...
ChangeThisServerNa...	19 ms	0/10	NewYork, United States
BlumJamulusServer	21 ms	0/10	Williamsburg, United States
MikeLSJServer	27 ms	0/10	NewYork, United States
MaahsJamulusServer	28 ms	0/10	Montreal, Canada
NES-Jamulus	28 ms	0/10	Virginia, United States
JoelSindelar	28 ms	0/12	Virginia, United States
JamPad	28 ms	0/10	NewYork, United States
Gordon	28 ms	0/10	Montreal, Canada
Mikey's Man Cave	28 ms	0/10	Canada, Canada
ip-172-31-28-78	28 ms	0/10	
TestJamulusServer	28 ms	0/10	Ohio, United States
eCompositor	28 ms	0/10	Yardley, PA, United States
BernaServer	34 ms	0/10	Ohio, United States
jam.eecs.umich.edu	34 ms	0/10	Ann Arbor Michigan
Mixed Bitch Studio	35 ms	0/10	Alexandria, VA, United States

Server Address

BERKLEE COLLEGE OF

Classroom Application

Jim Odgren



- Private Instruction
(one-to-one sessions)
- Performance Studies lecture/playing classes in jazz improvisation
- Small Jazz Ensembles
(6 students, average)
- Performance Studies Recital Preparation class
(average, 10 students)

BERKLEE COLLEGE OF

The Students' View



Berklee
Global Initiatives

Questions & Answers

Case Study: Performing Band Rehearsals on the Internet With Jamulus

Volker Fischer

Abstract

The author of this paper is a member of a rock cover band performing weekly rehearsals on the internet over a period of three years. During that time a lot of practical experience with real time jamming was gained. In this paper the author shares these experiences and gives an overview of the Jamulus software which was used to jam online discussing the most important facts for a successful setup.

1 Introduction

Since the quality and speed of internet connections have improved tremendously in the past years, performing distant online jamming in real time is possible now. There are near real time solutions like Ninjam [Incorporated, 2015] available with a synchronization mechanism based on musical form. It provides a way for musicians to improvise together but real songs cannot be played with this system. Jamulus [Fischer, 2015] is a similar software like LDAS [Saebø and Svensson, 2006], Soundjack [Carot et al., 2006] or eJamming [Blythe and Sewell, 2011] for actual real time networked music performance. It enables musicians to rehearse songs like they would if they were in the same room together.

Online jamming has many advantages, e.g. musicians get the possibility to jam with others residing too far away, where they would not have a chance to practice in a traditional way. Another advantage is that no rehearsal room must be rented and no instruments need to be transported. The virtual rehearsal room is basically moved to the home.

In this paper the long time experience with online jamming is presented. In the first part of the paper the Jamulus software is introduced. In the second part the online band experience is shared.

2 The Jamulus Software

The Jamulus software enables musicians to perform real-time jam sessions on the internet us-



Figure 1: Jamulus software on a typical jam session

ing narrow band networks like a Digital Subscriber Line (DSL) or a Cable Modem (CM). There is one server running the Jamulus server software which collects the audio data from each Jamulus client, mixes the audio data and sends the mix back to each client. Jamulus is Open Source software (GPL, GNU General Public License) hosted at SourceForge and runs under Linux, Windows and MacOS. It is based on the Qt framework and uses the OPUS audio codec [Valin et al., 2012]. A screen shot of Jamulus is shown in Figure 1.

The initial release on SourceForge was in 2006 where Linux was the only supported operating system using a native ALSA audio interface and a simple ADPCM audio coding. In the most recent version, Jamulus is a multi-platform, easy to use software utilizing the JACK audio interface and a high quality audio coding library. Private jam servers can be registered at the central server so that they show up in a list on the Jamulus client server menu.

The simplified core structure of Jamulus is shown in Figure 2 and starts with a callback based audio interface which captures audio sample blocks. These blocks are compressed with

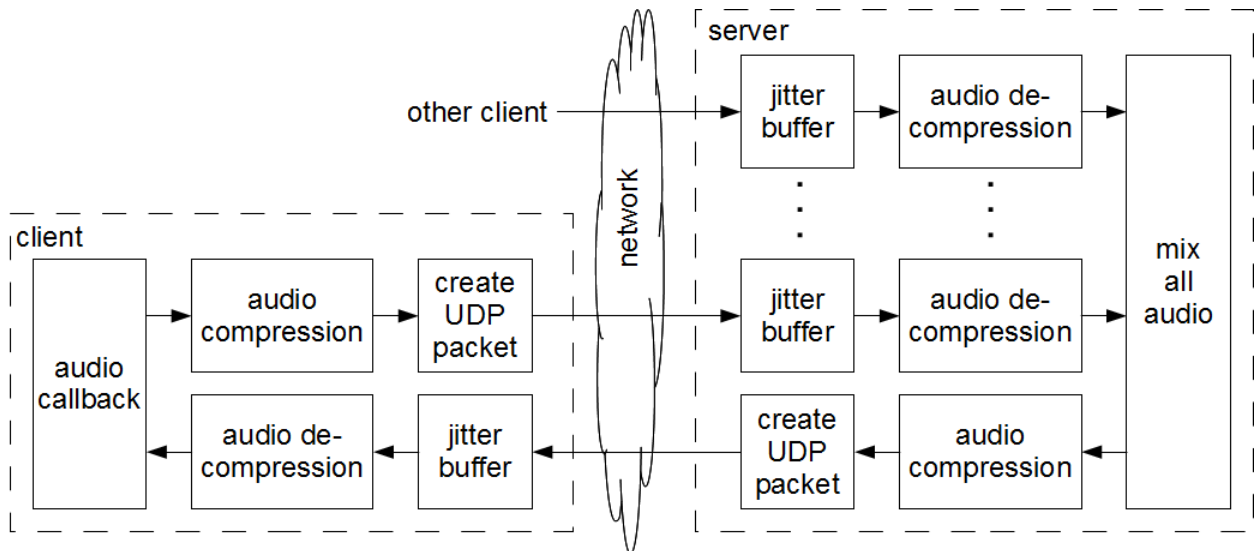


Figure 2: Simplified Jamulus block diagram

the OPUS low-latency audio codec and transmitted through the internet using the connectionless User Datagram Protocol (UDP). At the server a set of jitter buffers collect the asynchronous network packets from all connected clients. In the server processing loop, the data packets from each client are taken from the jitter buffer, decompressed and mixed together. This mix is again compressed with OPUS and transmitted to all connected clients with UDP packets. Back at the client, the received server packets are stored in a jitter buffer. On the next audio interface callback, a network packet is taken from the jitter buffer, decompressed and sent to the sound card for output.

Jamulus follows the *keep it simple* principle. There are some papers about optimizing the network queue by inserting a counter in each network packet so that the stream can be re-assembled correctly at the receiver (e.g. [Sæbø and Svensson, 2006]). Other papers suggest to resample the audio stream to compensate for the different sample rates in the sound cards to avoid buffer overflows due to timing drifts (e.g. [Carot et al., 2009]). None of this is supported in Jamulus since these mechanisms target highly reliable high bandwidth networks and result in a nearly dropout-free audio stream. The experience with the Jamulus software using UDP packet transmission over the internet is very different. Frequent audio dropouts are very likely so that the optimizations of these methods would probably make no noticeable difference. If the network is in trouble, it usually does not make sense to try to reassemble the

packets in the correct order since their delay is already so large that a buffer overrun will occur.

The next sections give more information about the decision for implementing a client-server system and discuss typical issues using Jamulus.

2.1 Client-Server System

The concept of distributed music performance can be implemented in software by using a multi-point connection (i.e. each client connects to all other clients) or as a client-server system. Theoretically the multi-point connection gives lower latencies because only one jitter buffer is required and the direct network path between two connecting clients is usually shorter than routing through a server. But the drawback is that some smart synchronization algorithms must be used for compensating for the different path delays. Also, the network load is high at each client which results in a limited number of clients which can be connected at one time depending on the available internet stream rate.

In Jamulus, a client-server system is implemented because no special synchronization is required. All streams are mixed at the server and each client receives the same mix. This is a very simple concept. The drawback is that a server must be located close to all the musicians jamming together. To solve this problem Jamulus implements a procedure so that every client can start their own server which is then registered on the list of available servers. The central server manages a server list which is distributed to each client. The list is sorted by the ping

time to the server so that it is easy to select one near to the client. At present usually about ten servers are registered which are spread all over the world. The central server is a medium range virtual server which proved to be very reliable and stable even with eight or more connected clients.

2.2 Typical Issues Using the Software

For the concept of online jamming with various different musicians it is indispensable that the servers are crowded. Therefore Jamulus must be intended for end users and it has to be user friendly. With a straightforward GUI design and an auto detection of the jitter buffer size, Jamulus tries hard to get to this goal. Unfortunately for the software to work correctly, a lot of other things have to be setup correctly. It starts with the analog audio setup, i.e. an instrument and/or a microphone has to be mixed and connected to the sound card of the PC. The PC hardware must be prepared for real time processing, i.e. no CPU or network demanding software may be run in parallel to Jamulus and things like energy saving techniques of laptop processors have to be turned off. Correct audio drivers must be installed and the parameters like the buffer period size must be set to a small value to get the lowest latency possible. The internet connection must provide not less than the minimum required bandwidth and, at least equally important, must have a low ping time to the server. If even one of these requirements are not fulfilled, the performance is bad or Jamulus does not work at all.

One typical beginner problem is that the users do not suppress the direct sound of their instrument (e.g. they use a loud guitar amplifier). To play synchronized with the others, one must only listen to the audio mix coming back from the server. All local direct monitor sounds must be muted which is challenging for acoustic instruments like a drum set. One possibility is to wear ear plugs and increase the volume of the headphones.

The reality is that these requirements are so stringent that the Jamulus servers are empty most of the time. Even worse, some users just try out the software and get a bad performance because of an incorrect setup. With a bad first impression they convince others that online jamming in real time would not work. One way to overcome this problem is to provide the hardware together with the software like an im-

plementation on the Raspberry PI [Meier et al., 2014].

But even the software solution on a PC works astonishingly well as will be shown in the next section.

3 Online Band

For about three years the author of this paper is a member of a rock cover band performing weekly rehearsals with Jamulus. The band has four musicians. Instruments are drums, two electric guitars, a bass guitar and vocals. Except for two brothers, the members did not know each other before they met by chance on a Jamulus server, which may be the best way to find people who accept the novel rehearsal conditions and commit to the band for a longer period of time, since it has become apparent that it is really hard to find new musicians. Sometimes you meet people on the servers randomly but usually most of them are just doing a quick test and are not willing to spend a lot of time with Jamulus. Posting a query for a new band member in a public forum for musicians was not successful. People may try it out for a short while but soon discover that the new way of jamming is not what they want.

Jamulus is a totally new way of realizing a band. The traditional way is to find musicians living close to each other, to select/write songs, to rehearse a while and then go to a stage and make a public performance. A gig is the ultimate motivation for all the rehearsals. But as a virtual band using Jamulus it is tough to have a real gig since the musicians are usually living far from each other. In the case of the author's band the average distance is about 300km. In the three years of existence it was not yet possible to organize a real gig. So the journey (rehearsal) is the reward, not the gig. We have not found many others that are willing to accept this.

One of the advantages of online jamming is the flexibility of the location. Traditional bands have their rehearsal room where they meet and practice. If one player is traveling or visiting friends or family members far away, he cannot attend. This is different with Jamulus. If at that place a sufficient internet connection is available, the player can participate from that location assuming he has taken his instrument with him.

3.1 Online Experience and the Issue With the Latency and Audio Dropouts

There are a lot of people arguing that online jamming does not work because the overall delay is too high. They claim that latencies above 10 ms are not acceptable. In Jamulus typical overall delays are in the range of 30 to 70 ms. It is also typical that audio dropouts occur frequently (usually every 2 to 10 seconds per audible pop/interrupt sound).

It has been our experience that, if the audio dropouts are not occurring too often, they are not perceived if the band members concentrate on their playing. If one focuses on the dropouts (and this is what people usually do when they try out the software) it feels annoying.

The influence of the delay is kind of subtle. The lower the latency, the better the jam works out. Considering the drummer, a usual effect is that fast fills do not work anymore with a high delay. The timing of the strokes is not as exact as it is without the delay. To a certain extent it is possible to get used to the delay after a long time so that some fills will work fine even with some audio delay. But other fast fills will never work, so one solution is to either omit the song which requires that fast fill or replace the fast fill by a slower one.

It has to be mentioned that in some rare cases the rehearsal must be cancelled because the connection at that moment is too bad and there is no way to fix the problem. E.g., if an important soccer game of the national team is happening at the same time, a lot of people stream the event over the internet and the ping times can go up. But this can be compared to a traditional band rehearsal where e.g. the car of the main singer is broken and because of this the rehearsal must be cancelled.

A major issue for the stability of the connection are other devices sharing the same internet connection, especially if Jamulus is used over a wireless computer network (WLAN). Nowadays almost everybody has multiple devices like laptops, tablet computers and smart phones which access the internet frequently. E.g., if the online rehearsal is ongoing but other family members want to watch online videos at the same time, this does not work except for internet connections providing a large bandwidth. So usually before the rehearsal starts all people using the local network are advised to reduce internet usage. Since some devices load updates from the

internet at random times, it is wise to just turn them off during the rehearsal. Even the PC on which Jamulus runs should be monitored for update processes or an anti virus software scan.

Another strange effect has to be mentioned. This is the ping time to the servers. The ping time seems to depend on a lot of factors like how the network packets are routed (a route may change due to changes in the core internet network) or if the hardware of the home internet modem has a problem. A band member may have low ping times to the main server but suddenly the ping times change to a higher value without having changed anything on the local hardware or settings. So one is at the internet providers mercy when it comes to the ping time.

Concluding fact is that the better the connection quality, the more fun it is making music using Jamulus.

4 Conclusions

The Jamulus software enabling real time jamming on the internet was introduced. It is a free open source tool which was developed over many years. The main focus is on band rehearsals and online jamming using narrow band networks which are available at most homes. A band where the musicians are distributed all over Germany has used the software for years and has proven that online jamming in real time is a usable alternative to traditional rehearsals. The setup is not trivial but as soon as the initial hurdle is overcome, the fun begins.

The download statistics and the number of private servers which are created by the users show that there is an actual interest of the community in this new technology. Unfortunately the servers are empty most of the time. It is time to spread the information that online jamming works and is a lot of fun, so that the servers are crowded and a lot of interesting and enjoyable jams are happening.

5 Acknowledgements

My thanks go to my bandmates Alex, Philipp and Nobbie for all these great online jams and to Chuck (AA0HW) for reviewing this paper.

References

- Mark Blythe and Trevor Sewell. 2011. Ejamming all over the world. In *ACM Conference on Human Factors in Computing Systems*, Vancouver, Canada.

Alexander Carot, Alain Renaud, and B. Verbrugghe. 2006. Network music performance (nmp) with soundjack. In *NIME 06. New Interfaces for Musical Expression: Performance Workshop*.

Alexander Carot, Torben Hohn, and Christian Werner. 2009. Netjack - remote music collaboration with electronic sequencers on the internet. In *Linux Audio Conference 2009*. Casa della Musica, Parma.

Volker Fischer. 2015. Jamulus - internet jam session software. <http://sourceforge.net/projects/llcon>. Accessed: 2015-01-05.

Cockos Incorporated. 2015. Ninjam. <http://www.cockos.com/ninjam>. Accessed: 2015-01-05.

Florian Meier, Marco Fink, and Udo Zölzer. 2014. The jamberry - a stand-alone device for networked music performance based on the raspberry pi. In *Linux Audio Conference 2014 Proceedings*, pages 31–40. Institute for Music and Acoustics (IMA), Karlsruhe.

Asbjørn Saebø and Peter Svensson. 2006. A low-latency full-duplex audio over ip streamer. In *Linux Audio Conference 2014 Proceedings*, pages 25–31. Zentrum für Kunst und Medientechnologie (ZKM), Karlsruhe.

Jm. Valin, Mozilla Corporation, K. Vos, and T. Terriberry. 2012. Definition of the opus audio codec, rfc 6716. Internet Engineering Task Force (IETF).