Accessible Sound Control and Collaboration over the World Area Network

Kathryn Lovell, Rensselaer Polytechnic Institute

***Abstract*—Musical technology has yet to keep pace with new ideas about design and disability. We aim to develop a holistic musical performance program designed specifically for those with physical disabilities. This paper details a networking solution to allow a musician to seamlessly play with a band across the world, developed in particular for those with low mobility. This functionality is implemented using the websocket npm module, with the client program placed on band members' computers running Ableton Live, and an Amazon EC2 instance running as an echo server. The use of OSC allows flexibility over the kind of audio signal sent.**

I. Introduction

> "Disability is not just a health problem. It's a complex phenomenon, reflecting the interaction between features of a person?s body and features of the society in which he or she lives." [16]

Since the dawn of the computer network, it has gradually shaped and shifted nearly every aspect of modern society. This shift implores that as designers of a nearly universal user experience, engineers must consider the diversity of their users' experience.

Although the features of a person's body are immutable, the features–the design–of their environment, depend the people who design it. Many networked technologies already help further this goal. Teleconferencing and instant messaging programs such as Skype and Facebook Messenger already dramatically improve the lives of many people with disabilities. [5] With the proliferation of this technology, those who are deaf or hard of hearing can finally converse remotely, which wasn't an option when there was only telephones. Additionally, cloud computing technologies have allowed small portable devices to access the computing power needed to execute complex processes, such as processing natural language. As a result, those who are blind can use their smartphone by activating Siri [6], or use Microsoft's Seeing AI [11] to describe the world around them.

Unfortunately, musical technology has yet to leap into this design shift. It is well known that learning an instrument and being able to play music improves cognitive function and quality of life, and those who lose the ability to play or never have it to begin with are unfortunately kept away from pursuing what they love. Playing with a band has particular social benefits. However, although people with limited mobility can talk with others over Skype, it would be difficult to play with one's band over the service. The lag and drop in audio quality would drastically effect the performance. Not to mention it would be nearly impossible for other band members to do anything with this data, like add effects within a DAW.

This paper has been written to propose a holistic performance program designed specifically for people with spinal cord injuries, nervous system disorders, or conditions that would otherwise limit someone from playing music in a traditional fashion. Using a new version of the Jamboxx, created by My Music Machines, Inc. (MMMI), as well as Ableton Live 9 and Max/MSP, we have developed a way for anyone to access the world of and live performance electronic music production, DJing, and performance.

There are several options available for transferring MIDI data over the internet. This paper presents a minimal means for achieving this goal, entirely independent of operating system, requiring no virtual MIDI cables, and using as few dependencies as possible. The use of OSC also allows flexibility over the kind of audio signal sent.

This paper describes how to implement this functionality using Node.js and, correspondingly, a few npm dependencies. Although this is the case, the concepts in this paper are meant to the generally applicable to any web protocol implementation.

II. The Jamboxx

My colleague Robert Smock, in his paper "Assistive Music Production and Performance Using

the Jamboxx, Ableton Live 9, and Max for Live," details how new MIDI controllers and electronic performance technology has tended to follow the physical form of traditional instrumentation. DJ controllers mimic analog record players, and a whole plethora of electronic mimic the form & function of traditional instruments, from MIDI keyboards to the Synthophone MIDI Saxophone used by Eric Klein. [14] These controllers pose the same problem that most traditional instruments always have: they are not very accessible. Individuals with physical disabilities that limit their motor function, especially those with spinal cord injuries or nervous system conditions, have long been locked out of this music scene as nearly every single controller available on the market requires use of one's hands in order to play. But despite the barriers, these technologies still have massive potential to improve the lives of those with disabilities. [13]

Robert Smock, Ian Rios, and I were approached by David Whalen in April 2017 with the request to teach him how to DJ. However, instead of a traditional DJ controller, David works with a unique instrument, the Jamboxx. Experiencing firsthand the effects of noninclusive design, David hatched the idea for the Jamboxx back in 2005. With the help of Mike DiCesare, the Jamboxx developed into a highly capable instrument, played like a harmonica. It has been featured at the International Symposium on Adaptive Technology in Music and Art (ISATMA) in previous years.

Despite the Jamboxx's obvious design strengths, the functionality of its software is limited. Instead of a MIDI controller, the instrument currently functions as a Human Interface Device, similarly to a mouse or game controller. The musical functionality of this device is interpreted entirely by Pro Suite, Using a MIDI cable such as LoopBe Internal MIDI [12], information produced by the Pro Suite software can be routed into any other program that can accept MIDI information and used to play a wider range of instruments, but even with this patch functionality remains limited. Because of the amount of time it takes for received data to be converted by Pro Suite, sent out as MIDI information, taken in by the receiving program, and synthesized, users who took this route experienced high levels of latency that made live performance especially difficult and inhibited the ability of the Jamboxx to serve the purpose of an all-in-one virtual instrument. Currently, I work to transform the next iteration of Jamboxx (the Jamboxx Pro) into a MIDI instrument.

In addition, my colleague Ian Rios created a solution for those who want to play music traditionally. His score reader, which will be integrated with the new Jamboxx Pro, allows one to play while following along with a sheet of music based on the MIDI data within Ableton. As a result, it even allows you to create a score and follow along with data you record yourself. This feature is extremely useful for those who can't turn a physical page of music, as well as band students using a Jamboxx that are learning to read traditional music notation. [15]

III. Formatting Musical Data

A. MIDI

Traditionally instrument data is transferred over the MIDI protocol. Data is sent in the format of 4-byte packets. A single note is triggered by a "note on" packet. This packet contains a value between 0-127 for the pitch data. When the note is subsequently released, a "note off" packet is sent for that same pitch.

In between these two notes subsequent messages can be sent that either turn on other notes or apply an effect, like pitchbend or aftertouch, to the existing note.

B. Open Sound Control

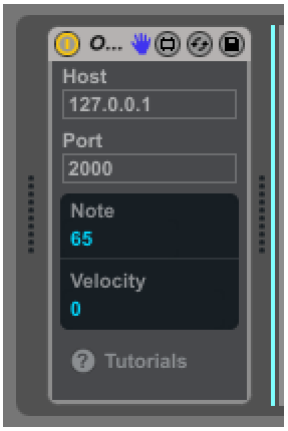
Open Sound Control is another protocol for transferring musical data. Unlike MIDI specifically designed for network communication. With its open-ended naming scheme, it provides the ability to transfer not just note data over your network. But what if we do want to send pitch data? Especially in tandem with other effects? [17]

Open Sound Control cannot replace MIDI entirely. On the lower levels, it behooves firmware developers to communicate without arbitrary tags, and instead by sending a USB packet with four bytes clearly defined. This data needs to be more readable by a computer than a person.

IV. Playing Collaboratively over the Internet

A. Sending Notes as OSC

Sending recorded MIDI notes out of the DAW as OSC is simple. Ableton's Connection Kit has



the MAX MSP Patch MIDISend, which does this exactly. [1]

This patch pairs the note and velocity values with /Note and /Velocity OSC address, along with the number of the voice. The note in the figure would be sent as:
/Note1 65
/Velocity1 0

Finally, the "udpsend" patch will send these values as a UDP datagram to the port specified.

B. UDP Protocol and TCP Forwarding

To communicate over the internet there are two protcols by which two computers can send messages to each other.

TCP stands for Transmission Control Protocol. This protocol is connection-based, meaning that when TCP sends a message, it can be sure that its connection received it.

In contrast, UDP sends messages with no guarantee on their receipt. Typically, UDP the preferred protocol for sending stream data, as well as for broadcasting.

In a video application, the end user cares relatively little if the occasional frame gets lost, but more so if they have to wait for the video to buffer. On the other hand, as with MIDI signals, while we may be able to stand the loss of a few aftertouch chunks, the chunks which contain the "note on" signal and "note off" signal are crucial, and sent only once. If either of the packets containing these signals are dropped, a note might either never appear, or instead be sustained forever. If a "note off" signal arrives for a note that was never turned on, disaster may occur.

The Ableton Connection Kit relies entirely on UDP for its OSC Packet transfer. This is in part because Connection Kit was designed to work solely over the local network, but also because it's largely used for effect data rather than note data. Similarly, effect Over such a short network hop, it's unlikely that many UDP packets will be dropped. However, when playing over long distances, there is a much greater chance of this, as well as of messages shuffling around.

In order to transfer these packets reliably, the relay server grabs these packets Ableton sent over UDP and transfers them over TCP to the other computer.

[9] [8]

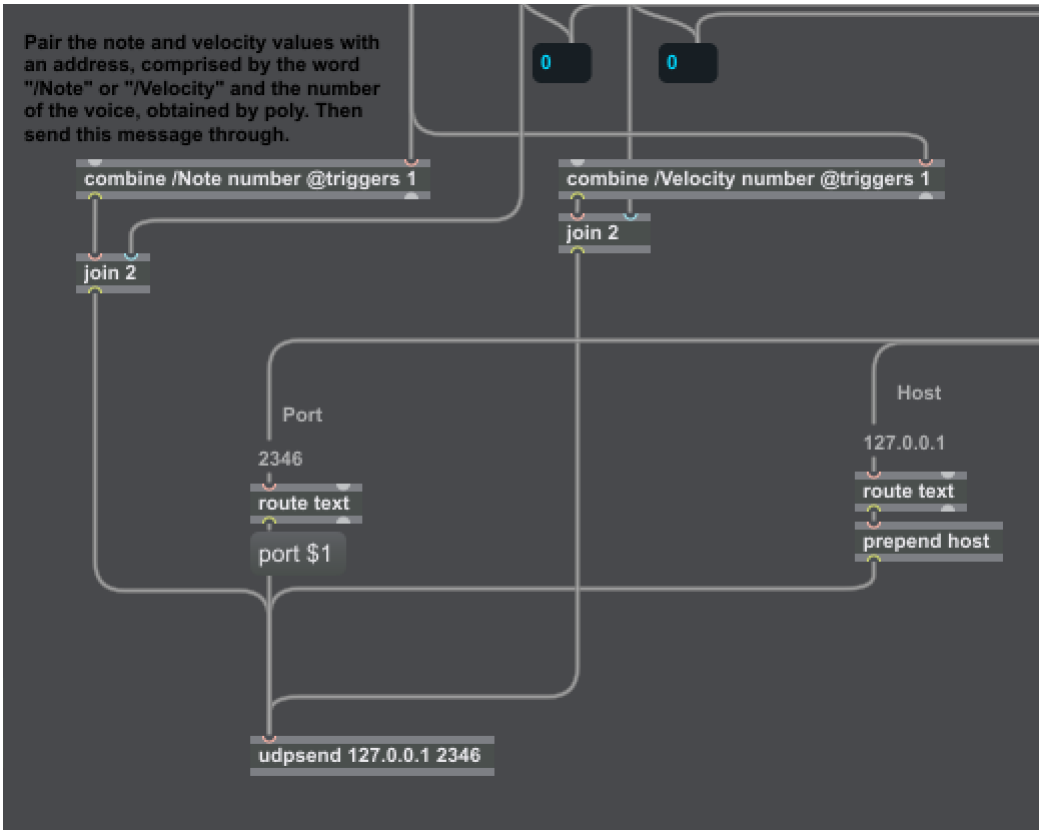
C. Server Functionality

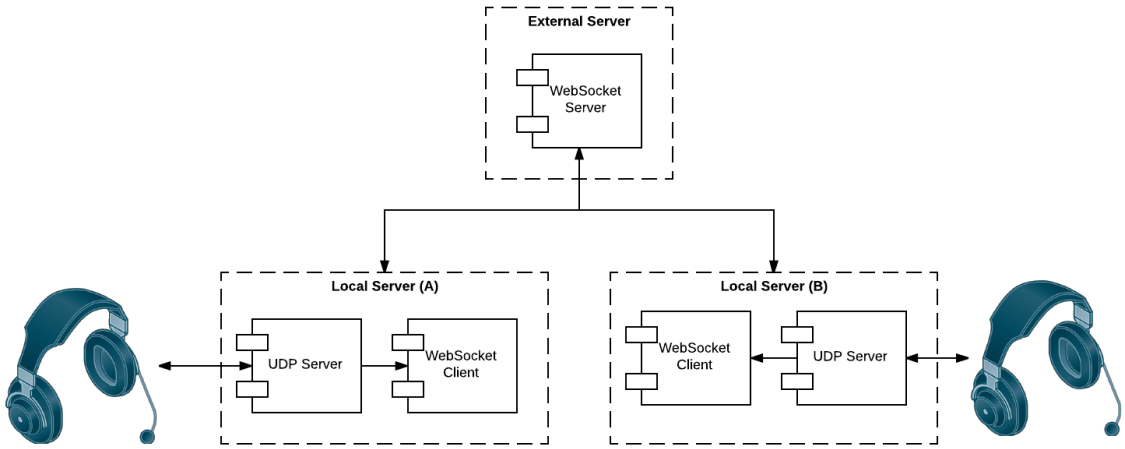
To communicate over the World Area Network, we must set up a web server relay by which the data can be passed from local network to local network. The HTTP protocol rides on top of the TCP protocol, and the websockets will use an HTTP server to open the connection.

> "In order to convert the messages that arrive over the local network via UDP into a TCP package we can send to our server, the code running on a PC has to function as both a client and a server. In order to first receive UDP from Ableton, a datagram socket is opened. This socket is owned by a client websocket that connects to the External Server." [10]

While the websocket client creates a connection to the External Server, outside its local network, since a PC must also listen for messages relayed back to it, it must also open a connection by which it can listen on its own network. Then it must relay that back to the UDP client. This is where the "server" functionality of the Local setup comes in. Upon receipt of a relay message, this server accepts the TCP message and transfers the data back over the UDP socket opened before.

The Remote Server itself is simply a server. As such, it accepts connections keeps track of all the clients currently connected to it. The server simply echos any messages it receives to all its other connected clients. At that point, the Local Server takes care of the rest.





D. Simplify NAT Addressing and DNS Name with Elastic Cloud Instance

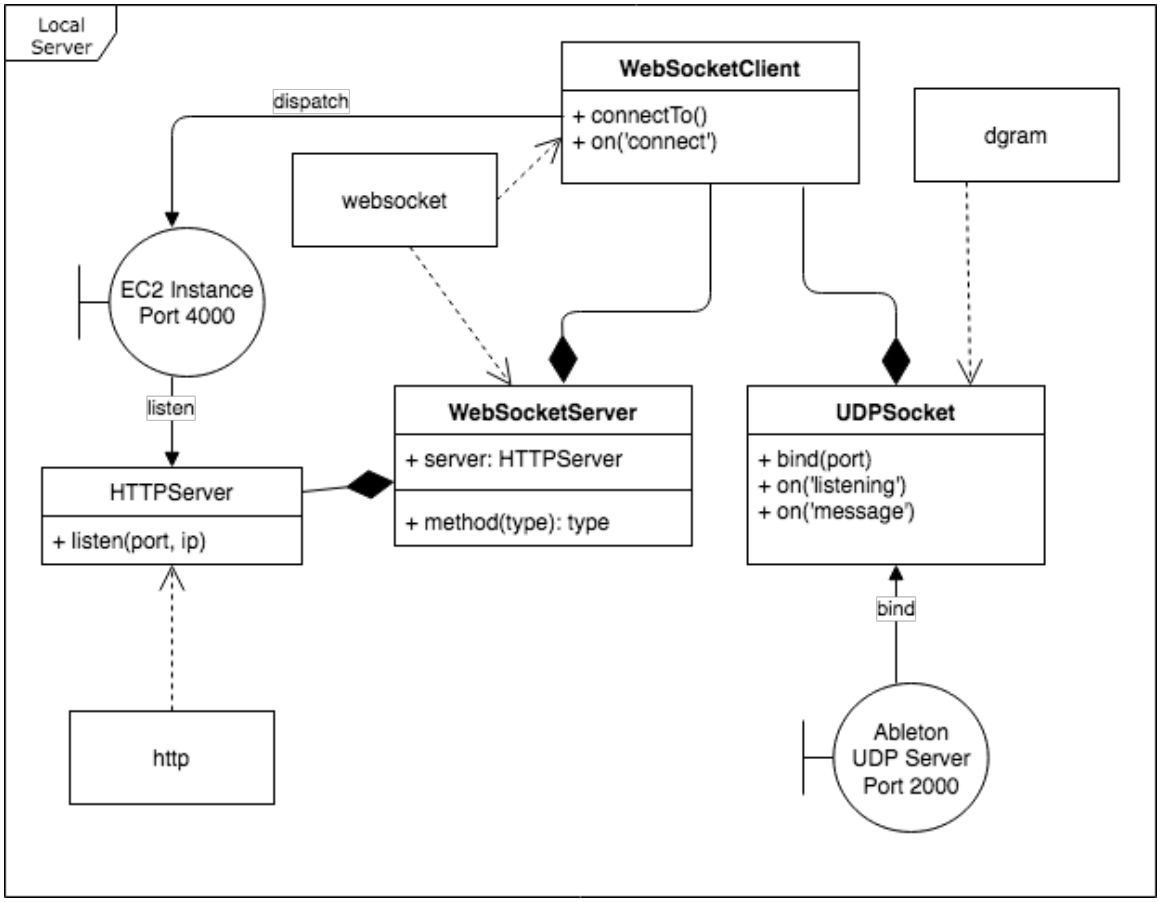
A device is known only by its private IP inside a private network.

With a 1:1 NAT firewall setup, requests from outside the network are translated into requests for a local address...A consequence of this is that from a server inside the network it's no longer possible to access the public IP address. So any HTTP requests for locally hosted websites will fail because a DNS lookup will return the public address which is unreachable. [7]

To get around this we can set up a local DNS name (bind) on the Remote Server, which will forward a request to that name to the Remote Server's private network.

Since Amazon configures a DNS name automatically when you create a computing instance, EC2 is one of the easiest ways to get a WAN server like this to work. [3]

Additionally, the use of a cloud computing in-



stance allows a group's network to persist without any one PC in the band. Once the server is started on the EC2 instance, any client connection can be closed and reconnect. This is opposed to if we had set up the server on one of the band computers, in tandem with the client for that computer. However, it is indeed possible to achieve similar end results this way.

E. Receiving Notes as OSC

Receiving notes and putting them back into a MIDI channel is trickier than sending the data. Ableton's Connection Kit has a MAX Patch for sending MIDI via OSC, and it also has the Touch OSC patch. Like MIDI Send, Touch OSC also contains a patch that opens a UDP socket. This socket will listen to the port we told this computer's Local Server to relay to.

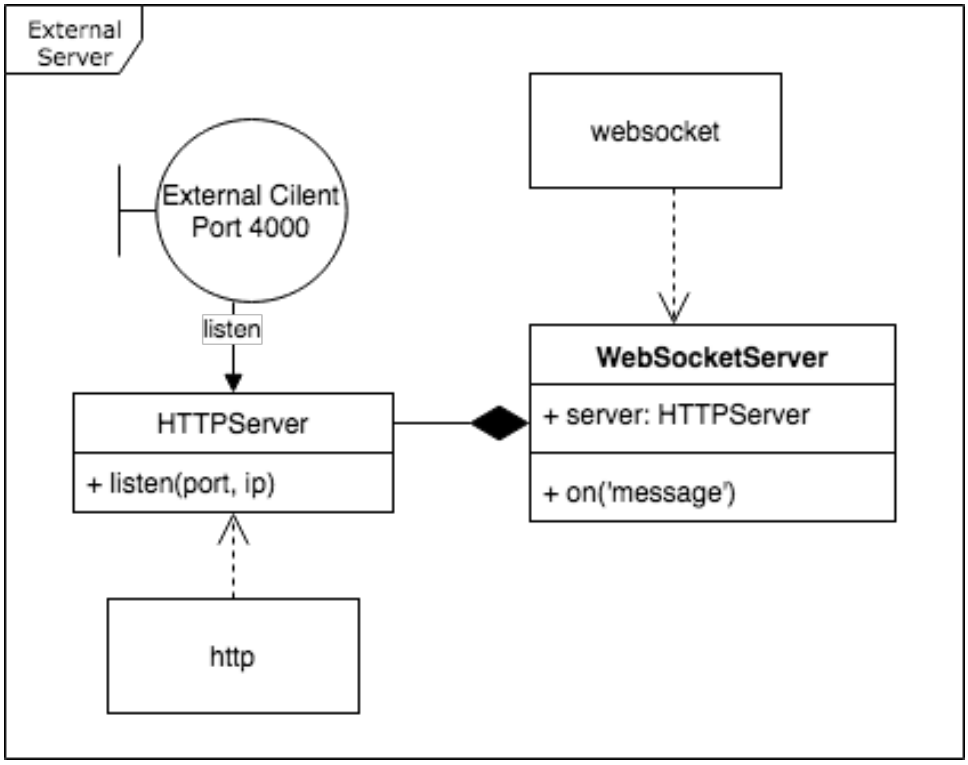
Touch OSC can detect the OSC data sent over the /Note /Velocity tags, and can even map them to any mappable parameter in Ableton. The problem is, this doesn't allow for output to a MIDI channel. In order to enable this, the patch must be modified so that the data is sent directly to the midi out. To do this, each OSC tag's value is parsed and sent to the midiformat patch. This patch will accept the raw integer values and convert them into a MIDI packet.

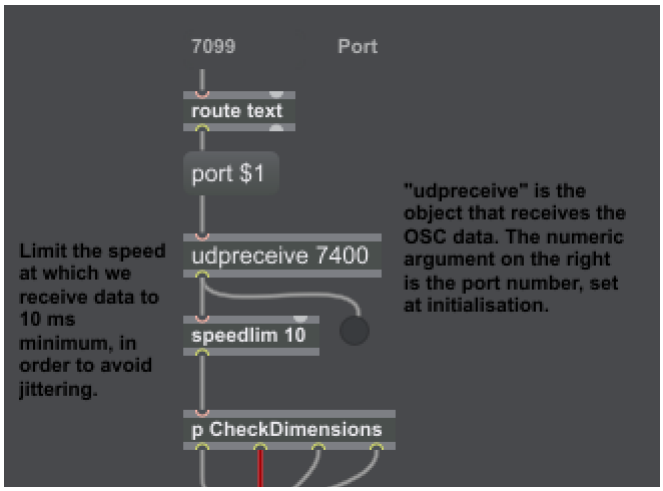
V. Future Consideration

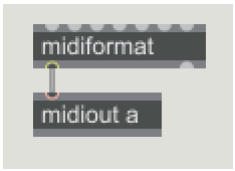
A. Findings

Because the only information passed over the internet in this program is simply two OSC tags, one for note and one for velocity, this significantly reduces the data transfer overhead in comparison to Skype, which has to transfer video data with audio as a compressed mp3. This significantly reduces the potential for lag, especially with low bandwidth connections.

Additionally, to account for inevitable transfer delay between computers separated by large physical distances, Ableton allows the MIDI transport to be nudged by a few milliseconds. [2] With a remote instrument player, the remote player can nudge the data they see from the band ahead by the total relay delay time, so that effectively the remote band member's playing is in sync with the rest of the band.







B. TCP Forwarding Alternatives

Although the TCP connection performs well enough, it is possible to further optimize by making the program use UDP entirely. To solve the issue highlighted earlier with MIDI note on/off signals, a distributed log algorithm such as Wu-Bernstein can be used. [18]

This will guarantee that if there is a record of an event y at a client computer, and event x happened before event y, then there exists a record of event x in the log of A. Thus, by looking at the log, we can be sure that every note off signal has a corresponding note on signal before it.

Although this solves any corruptions that may occur, it still may take awhile for log entries to propagate to every client. Thus, the receipt of the signals may be delayed by more than what is inevitable from the physical separation of the machines.

REFERENCES

[1] *Ableton. Max for Live Connection Kit. https://www.ableton.com/en/packs/connection-kit/*

[2] *Ableton. Live's MIDI Ports Explained. https://help.ableton.com/hc/en-us/articles/209774205-Live-s-MIDI-Ports-explained*

[3] *Amazon Web Services, Inc. Amazon Elastic Compute Cloud. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.h*

[4] *Back, David. Standard MIDI-File Format Spec. 1.1, updated. https://www.cs.cmu.edu/ music/cmsip/readings/Standard-MIDI-file-format-updated.pdf*

[5] *Choney, Suzanne. How Skype helps people with learning disabilities use sign language to gain confidence in the workplace. https://blogs.microsoft.com/firehose/2016/04/15/how-skype-helps-people-with-learning-disabilities/*

[6] *Christopherson, Robin. Siri update makes AI work harder for disabled users. https://www.abilitynet.org.uk/news-blogs/siri-update-makes-ai-work-harder-disabled-users*

[7] *Cronbie, Duncan. System: Accessing Public IP address from behind NAT. http://www.the-art-of-web.com/system/iptables-nat/*

[8] *Curtis, James. UDP vs. TCP. https://wand.net.nz/pubs/19/html/node6.html*

[9] *LeBlanc, Mark. Wenzel, Maria. Using UDP Services. https://docs.microsoft.com/en-us/dotnet/framework/network-programming/using-udp-services*

[10] *McKelvey, Brian. websocket. https://www.npmjs.com/package/websocket*

[11] *Microsoft reporter. Microsoft's Seeing AI app for visually impaired people released in the UK. https://news.microsoft.com/en-gb/2017/11/15/microsofts-seeing-ai-app-for-visually-impaired-people-released-in-the-uk/?utm_source=t.co&utm_medium=referral*

[12] *Schmidtt, Daniel. LoopBe1: A Free Virtual MIDI Driver. http://www.nerds.de/en/loopbe1.html*

[13] *Smock, Robert Dylan. Assistive Music Production and Performance Using the Jamboxx, Ableton Live 9, and Max for Live.*

[14] *Softwind. The Synthophone MIDI Sax. http://www.softwind.com/synthophone.html*

[15] *Rios, Ian. Using Javascript and Max MSP to Create Assistive Music Education and Performance Software.*

[16] *World Health Organization. Disabilities. http://www.who.int/topics/disabilities/en/*

[17] *Wright, Matt. The Open Sound Control 1.0 Specification. http://opensoundcontrol.org/spec-1_0*

[18] *Wu, Gene T.J. . Bertsein, Arthur R. . Efficient solutions to the replicated log and dictionary problems. https://dl.acm.org/citation.cfm?id=806750*