ANGLIA RUSKIN UNIVERSITY

FACULTY OF ARTS, HUMANITIES AND SOCIAL SCIENCES

DEVELOPMENT AND EVALUATION OF INTERNET OF THINGS ENABLED APPLICATIONS FOR MUSIC PRODUCTION AND CREATIVE PRACTICES

Marques J. Hardin

A thesis in partial fulfilment of the requirements of Anglia Ruskin University for the degree of Doctor of Philosophy

Submitted: August 2019

Acknowledgements

I first and foremost would like to thank my original supervisor, Professor Rob Toulson, for allowing me the opportunity to conduct this research under his expertise, and the unwavering support and guidance he provided during this journey. Even after his time at Anglia Ruskin University he offered continued external, supervisory assistance, dedicating hours for testing and evaluations to help see through the overall research.

My sincere thanks goes to my 1st supervisor, Dr. Shreepali Patel, for taking over after Rob's absence and for both offering insight and being available to help navigate the Ph.D process. I would additionally like to thank the funding body at Anglia Ruskin University for awarding the 3 year studentship to commence my studies.

Lastly, I would like to thank my mom and dad who were always there lending supportive words, and the rest of my family and friends for constantly motivating to continue through the journey even when encountering immensely difficult odds. Thank you.

ANGLIA RUSKIN UNIVERSITY

ABSTRACT

FACULTY OF ARTS, HUMANITIES AND SOCIAL SCIENCES

DOCTOR OF PHILOSOPHY

DEVELOPMENT AND EVALUATION OF INTERNET OF THINGS BASED PROCESSES FOR MUSIC PRODUCTION AND CREATIVE COLLABORATION

Marques J. Hardin

August 2019

The Internet of Things (IoT) is an emerging paradigm that introduces the concept of ubiquitous interconnected devices, where everyday objects across the world are connected over computing networks to accept, collect, and exchange data. Although there have been growing implementations of IoT in commercial and enterprise-driven applications, to date it has not seen substantial development and evaluation within creative fields, and less so regarding music and music production.

Mixing and recording techniques in traditional forms of music production largely employed the use of physical and analogue hardware. While each hardware device adds its own distinct sound attributes to processed audio, the growth of digital technology and software plug-ins granted more accessibility to high-quality production practices. IoT, however, presents a unique opportunity to maintain desirable characteristics of past (and perhaps lost or disappearing) hardware processes. With IoT-enabled hardware, for example, it is possible to add remote connectivity to rare, expensive, and bespoke physical audio systems. This can promote the concept of the ‘virtually-extended music studio,’ where music producers may work within personal environments and still retain options to access remotely-available devices.

This research explores IoT-enabled music processing by utilising practice-based methodologies to develop and evaluate a creative work that facilitates virtual engagement with remote audio hardware. The creative work is compounded by mixed-method investigations that asses and verify open source technologies and current network capabilities that can implement IoT music production systems, and additionally incorporates surveyed music producer feedback to give insight into how IoT can better bridge musicians to the music production process. The resulting analyses exposes how IoT-enabled music systems can empower new forms of creative engagement and collaboration, and can help adapt non-traditional techniques for greater options to express music, revolutionise new markets for equipment hire and distribution, and bring about the ‘best of both worlds’ in terms of analogue and digital production benefits.

Key Words: IoT, IoT Music, Internet of Things, Networked Music

Table of Contents

List of Figures

List of Tables

Abbreviations & Glossary of Terms

AES – Audio Engineering Society

commodity network – public, commercially-available Internet networks.

in-the-box (ITB) – music production exclusively mixed within a computer using software techniques.

IoT – Internet of Things.

JackTrip – Mac OS and Linux based system for streaming high quality, uncompressed audio over the Internet.

LAN – Local area network; computing network limited within a building, office, or institution.

latency – measurable delay between the transfer and arrival of data over a computing network.

ms – milliseconds.

NREN – National Research and Educational Network; dedicated, robust computing network that can interconnect research institutions over a high-speed backbone.

URLLC – Ultra-reliable and low-latency communications.

WAN – Wide area network; computing network incorporating broader, widespread computing devices and independent LANs.

WebRTC – an open source platform that allows real-time communication of audio, video, and data over web browsers and mobile devices.

Websockets – communication protocol allowing full-duplex, bi-directional communication between a computing server and client over the Internet.

Statement of Originality

I, Marques Hardin, confirm that the research included within this thesis is my own work, and that any work that has been carried out in collaboration with, or supported by others, is duly acknowledged and my contribution indicated.

Previously published material is also acknowledged below. I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Date: 31 August 2019

1. Introduction

1.1 Introducing the Internet of Things

> *The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services. An Internet of Things makes computing truly ubiquitous – a concept initially put forward by Mark Weiser in the early 1990s* (Mattern and Floerkemeier, 2010, p. 242).

The Internet of Things, or IoT, is a modern trend where standard, everyday objects are embedded with computing technology, creating a virtual interface to the Internet where they can collect and share data through networked connections. It is a paradigm where smart “things” are continuously interlinked through a series of complex communication networks in order to interact freely with humans and other interconnected devices. Miorandi, et al. (2012, p. 1497) argue:

> *It is predictable that, within the next decade, the Internet will exist as a seamless fabric of classic networks and networked objects. Content and services will be all around us, always available, paving the way to new applications, enabling new ways of working; new ways of interacting; new ways of entertainment; new ways of living.*

IoT has become one of the largest growing paradigm shifts in the early 21st century, and it is estimated that the number of connected devices world-wide is expected to reach 20 billion by 2020 (Gartner, 2015). As the number of networkable devices rises, the Internet is moving away from simply being a hub of static web pages and human services. The Internet of Things is giving way to a virtual framework that “allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any path, network and any service” (Vermesan, et al., 2009, p. 8).

1.2 Motivations for the Presented Research

Modern technology trends often inspire new forms of innovation, and many industries are exploring how IoT can play a role in their markets. (Newmarker and Buntz, 2017) acknowledge:

> *Every nascent industry needs a killer app. But in the Internet of Things realm, it can be hard [to] identify the most promising use cases. For one thing, the field is gargantuan, including everything from drones to connected jet engines*.

Recent years have seen large investments in IoT for industrial and economic development. In 2016 the UK established the IoT-focused PETRAS research consortium, a group of nine UK universities funded to investigate "critical issues in *privacy, ethics, trust, reliability, acceptability, and security"* (EPSRC, 2016). The PETRAS group was granted £9.8 million in funding in 2016 by the Engineering and Physical Sciences Research Council, yet while this investment offers huge contributions into new and innovative IoT research, the areas focus predominately around smarter cities, facilities, systems, and security, and exemplify a lack of current research and development for IoT services within creative fields.

Audio and music production is one area where IoT can augment creative processes. Whalley (2015) expresses concern that current musical applications of the Internet of Things have primarily focused on adding remote or virtual dimensions to the traditional composer-leader based musical performance, and provides limited focuses on other methods of expression. An unconventional application for music, for example, can be using the Internet of Things as a tool for designing unique interfaces to control virtual audio production processing, and add greater personal interactions with remote technology instead of simply being an amendment to the traditional music performance. This research addresses a growing motivation to incorporate IoT into the creative landscape by employing creative practice methods to investigate implications of the Internet of Things for music, specifically exploring opportunities using IoT technology to augment the music production process. IoT infrastructures can allow professional and bespoke hardware devices to be remotely extended into personal production environments, and thus conceptualise an idea of the *virtually-extended music studio,* where new opportunities to engage musical experiences and production workflows are not restricted by the need of physical presence.

Through practical design, implementation, and analysis this research evaluates how IoT-enabled music systems can potentially unlock new and impactful creative opportunities for musical artists and professional producers. IoT can offer a wider spectrum of readily-available music tools to the disposal of musicians, and offer such benefits as greater collaboration amongst producers, greater accessibility to devices and acoustic resources such as echo chambers and reverberant spaces that can be utilised in private work environments, and new strategies to share, distribute, and market audio and music assets.

1.3 Research Approach

This research follows a ‘creative practice’ approach to developing original contributions to knowledge. The creative practice itself is the design and creation of a unique IoT demonstrator for use in music production scenarios, inspired by the learnings of a thorough review of background literature and prior art practice in the field. A number of knowledge ‘gaps’ are identified from the prior art and are constructed as research questions to be answered through a justified methodology of creative practice and evaluation.

The research takes an iterative form of creation and testing, evaluating open source and emerging technologies that can be manipulated and combined in order to realise a working IoT system for use in music production. The developed demonstrator is then thoroughly tested both technically and from a user experience perspective, in order to realise justified answers to the posited research questions. Significant conclusions are drawn from the findings and further suggested work in the field is presented.

1.4 Original Contributions to Knowledge

A number of original contributions to knowledge are presented in this thesis, summarised as follows:

- A first original, detailed analysis of open source IoT technologies with respect to creative music applications.

- The creative development of a unique and innovative IoT demonstrator unit for use in music production scenarios, enabling and realising concepts including the ‘Internet connected reverb chamber’ and ‘Internet connected hardware units’ for music production, and hence enabling the first detailed evaluation of the concept of the ‘virtually-extended music studio’.

- The first analysis of Internet-controlled hardware alongside Internet streaming protocols for real-time, two-way audio streaming and real-time processing via the Internet.

- The first and most detailed critical analysis specifically of JackTrip and WebRTC streaming protocols in supporting high quality, real-time audio transfer across a number of modern computing networks.

- The first case of documented feedback from practitioners and experts in music production addressing impressions, principally highlighting perceived opportunities and concerns, of IoT-enabled music production systems.

- Original and unique implementation of enhanced mixed-method methodologies to critically investigate practical uses of the Internet of Things opportunities within a creative industry, focusing specifically on music production.

1.5 Organisation of the Thesis

The organisation of the thesis follows the progressive practice and development of the research, beginning with a review of relevant literature within IoT and music production, and a methodical overview of how the research is conducted to address raised research questions. The following sections detail the creative work, showcasing stages of the practical build of an IoT-enabled music system including links to relevant media and videos, and discuss the mixed-methods analysis of IoT music applications facilitated by the practical work. Lastly the thesis is concluded with a summarised discussion of the research analysis and the key findings of the investigations.

The arrangement of the thesis chapters are as follows:

Chapter 2 provides a review of existing literature exploring modern realisations and applications of the Internet of Things, potential opportunities to democratise the technology with the availability of open-source resources, and practical applications fusing IoT with creativity. The second half of the chapter offers insight into prior creative practice relating to networked music, additionally commenting on the rise of the Internet and its impact on Internet-driven applications. The chapter concludes with implications of IoT in influencing music production and composition.

Chapter 3 offers gaps in knowledge arising from the literature and practice review, and presents the research questions, aims, and research objectives. The chapter also provides an overview of the formal methods involved to investigate, conduct, and assess the goals of the research to answer the research questions.

Chapter 4 highlights the creative approach to the research and gives a detailed account of the practical development of an IoT-enabled music application which serves as a tool to answer the research questions and evaluate the research aims.

Chapter 5 details the first analysis of the research, offering a primarily quantitative approach to evaluate the performance of real-time audio streaming platforms over existing computing networks that can facilitate efficient delivery of audio to remote, IoT-enabled music processors.

Chapter 6 provides a second analysis focusing on qualitative understandings of current production trends, particularly comparing the use of software and hardware among music producers, and gathers insights into possible impacts of IoT-enabled systems on music production.

Chapter 7 offers a critical analysis of the overall research and summary of key findings obtained from the investigations and evaluations of Chapters 4-6.

Chapter 8 concludes the research presenting a summary of the main contributions to knowledge, addressing each research question and alluding to future work that can be derived from the investigations.

1.6 Associated Publications

Sections of Chapter 5 were published as:

> Hardin, M. and Toulson, R., 2019. Quantitative Analysis of Streaming Protocols for Enabling Internet of Things (IoT) Audio Hardware. In: Proceedings of the 146th Audio Engineering Society Convention, *Audio Engineering Society*, Dublin, March 2019.

The author of the thesis wrote and conducted the main research. Secondary author had supervising role, adding editorial feedback and assisted with technical programming code.

Additionally, elements of this research have been presented at the following international conferences, resulting in opportunities for future peer reviewed publications:

Hardin, M. and Toulson, R., 2019. Development and Evaluation of Internet of Things Technologies for Music Production and Creative Collaboration. In: *Art of Record Production Conference*, Boston, May 2019.

Hardin, M. and Toulson, R., 2019. Development and Evaluation of Internet of Things Technologies for Music Production and Creative Collaboration. In: *Innovation in Music Conference*, London, December 2019.

2. Literature and Practice Review

The review of relevant literature presents the primary areas of interest for the Internet of Things while giving additional insight into shifting use cases of the technology for personal and creative works. The chapter also discusses the development of networked music applications and the transition to modern day, Internet driven music processes. The chapter additionally presents a review of relevant creative practice and technology, and reflects on current investments towards intermixing IoT-based music concepts and theory with practice.

2.1 IoT Vision and Implementation

2.1.1 IoT Vision

The Internet of Things envisions a reality of omnipresent devices where the Internet provides a universal infrastructure for embedded machines that constantly exchange information and interaction. Miorandi, et al. (2012, p. 1497) states, “in such a perspective, the conventional concept of the Internet as an infrastructure network reaching out to end-users’ terminals will fade, leaving space to a notion of interconnected ‘‘smart’’ objects forming pervasive computing environments.” In order to realise the Internet of Things, Miorandi, et al. (2012, p. 1497) elaborates that “this innovation will be enabled by the embedding of electronics into everyday physical objects, making them ‘smart’ and letting them seamlessly integrate within the global resulting cyberphysical infrastructure.” Three technological areas can be seen as aiding in the adoption of more smart embedded devices:

1. **Identification**
   In order to be tracked and managed, non-IoT devices need to be addressable and uniquely identified across the network. As discussed by Bandyopadhyay and Sen (2011, p. 6), “in the vision of IoT, things have a digital identity (described by unique identifiers), are identified with a digital name, and the relationships among things can be specified in the digital domain.” One of the earliest technologies enabling the Internet of Things was radio frequency identification (RFID). RFID ‘tags’ can be embedded into an object and use radio technology to create a digital representation of the object on the virtual network. This also allows encoded information about the object (physical properties, historical data, current status of

the device, etc.) to be collected and used as desired by a virtual recipient (Thiesse and Michahelles, 2009).

2. **Sensing and Actuating**

   Sensory nodes are small, technical components that allow embedded objects to interact with their immediate surroundings. These nodes help IoT-based objects become "smart," as they allow the devices to interact with the environment, gather and process information, and seemingly "think" for themselves either autonomously or with the assistance of human feedback. The nodes usually serve as an interface to receive analogue signals (sound, light, pressure, displacement, etc.) and convert them into a digital format for processing and communication (Gubbi, et al., 2013). Actuators are mechanical components (typically small motors contained within an IoT system) that accept control information to perform specific tasks, such as taking active parts in moving, manipulating, or engaging directly with objects in the physical environment. Sensors and actuators serve as bridges linking the digital and physical realms of IoT (Miorandi, et al., 2012).

3. **Communication**

   The Internet itself is a huge network of interconnected communication devices that provide swift and widespread dissemination of information. Ethernet, Wi-Fi, and Bluetooth are currently popular communication mediums for computers and smart phones, however, new protocols are constantly being developed to additionally accommodate the range, speed, and power requirements of newly-adopted embedded devices. Some transport layer protocols include: Bluetooth (more specifically Bluetooth Low Energy (BLE) or Bluetooth Smart), Zigbee, and LoRaWAN, which allow short and mid-range communication between connected devices while reducing power consumption.

With growing mechanisms for adopting a number of widespread use cases, IoT can help contribute to a new world of ubiquitous computing. Ubiquitous computing, or "Ubicomp," explained by Weiser, Gold and Brown (1999, p. 694) "created a new field of computer science, one that speculated on a physical world richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives and connected through a continuous network." Within this vision, devices will always be connected, always sensing the world and interacting with the environment, collecting, processing, and distributing data, and have the option to do so autonomously while running in the background of normal human activities.

In a ubiquitous IoT world, interconnected devices will develop into an ever-present aspect of daily life and have an affective influence on every industry. However for real impact, successful implementation of IoT systems requires the cooperation of various independent and isolated industries, ultimately encouraging new collaborative markets. One example of this is the smart fridge (Miorandi, et al., 2012), where a networked refrigerator uniquely identifies its contents by reading RFID tags placed on super market items. In this scenario, the fridge can provide data regarding the number of items and properties of those associated items it contains. The smart fridge requires additional collaboration between numerous industries (i.e. electrical appliance companies working with the farm and agriculture industry) in order to collectively work together to fulfil the needs and demands of the customer (Miorandi, et al., 2012).

The future of the Internet of Things requires an intricate infrastructure of different, independent technologies woven together to create a qualitative change in life. To create a smart vision of the world, multiple systems (sensing and computing networks, signal processing, data collection, security, etc.) have to continuously work together to create accurate processes that benefit changing environments and the complexity of human nature (Stankovic, 2014). This poses new business models as it increases the need for more cross collaboration and research across a spectrum of trades. As summarised by Miorandi, et al. (2012, p. 1509):

> *Besides enhancing the competitiveness of various vertical markets, IoT technologies can open up new business opportunities by: (i) bridging vertical markets, giving rise to cross-cutting applications and services, based on the use of a common underlying [Information and Computing Technology] platform, (ii) enabling the arising and growth of new market segments and applications, made possible by the ability, provided by IoT technologies, to interact with physical objects via digital means and (iii) optimizing business processes by leveraging on advanced analytics techniques applied to IoT data streams.*

2.1.2 Movements towards Democratised IoT Applications

In order to make IoT truly ubiquitous, there is a constant push to make IoT-enabling technology faster, cheaper, more efficient and accessible. As the technology becomes more available to public consumers, unique opportunities emerge allowing individuals to produce products and implement projects catered to specific groups and personalised audiences. Greater availability to IoT technology creates a growing demand for more resources to aid in personal IoT development, and the response has resulted in a rising trend of do-it-yourself applications and open-sourced knowledge exchange.

The open source movement emerged from a decentralised community of programmers, coders, and developers who create software with unrestricted access to source code in opposition to traditional proprietary software that govern the commercial world (Bonaccorsi and Rossi, 2003). The driving force behind open source software is its "community participation model and licensing model" that "encourages community participation, which means that open source software is truly software by developers, for developers" (Hendrick, 2018). The open source community benefits from free and open knowledge transfer through code sharing, as well as software and programs that can be used, manipulated, and improved upon by other professionals. Additionally, hobbyists and developing programmers benefit from open code that serves as learning aides and tutorials that assist in the development of personal projects.

Although IoT is still in its infancy, open source tools have provided a significant factor in driving IoT development and are reported to be used in 39% of cases by enterprises as compared to 36% for proprietary tools in the 2017 Worldwide IoT Innovation Survey (Hendrick, 2018). The heterogeneity of IoT devices and applications combined with the lack of cross platform standards adds some complications to the development of IoT projects, and a reliance on dedicated tools from proprietary resources could further compound issues. As a result, Bonaccorsi and Rossi (2003, p. 1245) mention that

> *Many Open Source projects take shape because the people promoting them have looked in vain for a programme to perform a particular function. They arise, that is, to satisfy a demand for which there is no corresponding supply, in short to 'fill an unfilled market.'*

Whether used for the benefit of single individuals or entire enterprises, the open source community has become a key tool for knowledge exchange used to promote efficient implementation of IoT projects that would be difficult to realise and otherwise costly if solely reliant on proprietary support.

2.1.3 Extensions of IoT into Creative Fields and Applications

While IoT technology is effectively becoming more available and accessible, Gubbi (2013) lists that the four domains largely impacted by the Internet of Things are personal and home, enterprise, utilities, and mobile. Within these industries, a bulk of the applications are geared towards improving services for home automation, healthcare, information technology, power and public services, transportation, and security. This list arguably shows that the Internet of Things is still heavily invested in the business and commercial sectors, but cheaper components and readily available prototyping interfaces are making the Internet of Things more accessible to innovators, hobbyists, and allowing the creation of a niche space for artists and practitioners within creative markets. The following examples explore some IoT use cases within creative industries:

Long Distance Art

Figure 2.1 Alex Kiessling's Long Distance Art exhibit (Zolfagharifard, 2013).

The *Long Distance Art* exhibit was created by Austrian artist Alex Kiessling, and sought to use robotics and wireless networks to create and replicate a painting simultaneously in three different locations across Europe. The project involved Kiessling sketching a drawing in Vienna, Austria, which consisted of a full image of a human face in the centre of a canvas with two half-faces drawn on the sides of the canvas. While Kiessling produced the drawing in Vienna, two robotic arms located in Trafalgar Square, London and Breitscheudplatz, Berlin recreated his drawing at exactly the same time in their respective locations (Zolfagharifard, 2013). The technology involved infrared sensors on a touch frame canvas that was captured movement by a *Microsoft Kinect* device. The Kinect tracked the artist's pen movements on the canvas from Vienna and transmitted the coordinates to servers that directly controlled the robots' movements in both London and Berlin over a dedicated satellite network (Visnjic, 2013). Once the three drawings were completed they were combined to make one larger, full image that was displayed in exhibits in Vienna and London.

Underwater

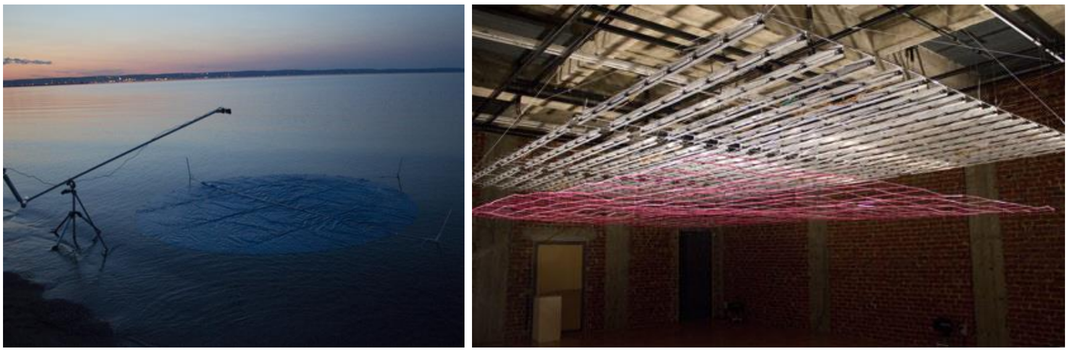


Figure 2.2 David Bowen's Underwater art exhibit (Chalcraft, 2012).

Underwater is an exhibit created by American artist David Bowen that articulated the motion of water into mechanical movements of a grid structure suspended in air. In order to capture the movement of the water, a circular membrane was placed on the surface of the water, and a Microsoft Kinect was mounted above the membrane to capture a 3-dimensional model of the rippling movement of water under the membrane (Chalcraft, 2012). The data from the water movement captured by the Kinect was then sent to 486 servo motors connected to specific points along the frame of the grid (Chalcraft, 2012). The servo motors were able to accurately move the grid in a way that replicates the subtle movements of the membrane in the water and thus gave a unique perspective of how the real-time waves look in an alternative space.

Silophone

Figure 2.3 [The User]'s <u>Silophone</u> sonic art exhibit ([The User], 2000).

Silophone is a sonic art exhibit created by Thomas McIntosh and Emmanuel Madan, collectively known as *[The User]*, that turns an abandoned grain silo in Montreal, Silo #5, into a musical instrument. Microphones and speakers are placed throughout the silo and audio can be transmitted inside using the speakers, where the audio is then "transformed, reverberated, and coloured by the remarkable acoustics of the structure, yielding a stunningly beautiful echo" (Reddel, 2003, p. 19). The transformed audio is then recaptured by the microphones and delivered to the listeners outside of the silo. Originally, audio files were able to be transmitted inside the silo by uploading and streaming the files via the Internet on the www.silophone.net webpage. Additionally a telephone number could be used to place a call directly into the silo, and a permanent, publically-accessible sound platform located outside of the silo is currently available for users to sing or have other acoustic sounds delivered into the silo returned back to the platform. The Silophone exhibit inherently added new significance to a devalued, under-utilised sonic structure and turned the silo into a natural, remotely-accessible reverb chamber.

While a major drive in the development of IoT is optimising commercial and business applications, these examples show diversity in how IoT can stimulate creativity and promote development in creative industries. Networked and interconnected devices can assist generating unique productions for arts and humanities, and contribute to novel techniques to engage different aspects of multimedia, examples being art, film, and music.

2.2 Network and Internet-Driven Music

2.2.1 A Brief History of Networked Music

A broad concept of "networked" music can extend as far back as the earliest civilisations, as collaborations between human performers have been used to create consolidated pieces of sound and music compositions. However, it was not until the 20th century that *network music* began to describe musical compositions aided by an assembly of complex technology connections. Modern advancements in electronics (i.e. transistor radios, personal computers) enabled a new domain in which digital technology added to the creative process of music creation.

One of the earliest contributors to the idea of networked music was John Cage. Cage was a music composer who was interested in experimental music, and in the 50s began to focus on unstructured music governed by chance operations (Cage, 1990).

In 1951 Cage performed *Imaginary Landscapes No. 4,* which was an experimental performance using various sound outputs from 12 transistor radios. The performance consisted of 24 performers; a pair for each radio that involved one person tuning the stations for the audio streams and the other adjusting the volume levels. Although there was a score sheet dictating the tuning frequencies and sound levels, Cage nor the other performers knew what type of sounds would be produced during the experience or if the dictated stations would produce sound at all (Worby, 2009). While not an explicit case for networked music, this was notably a first case of electroacoustic music relying heavily on the interconnectedness of digital devices.

The 20th century continued to see an evolution in digital electronics, and most notably the creation of personal computers gave people greater access to computing resources and digital networkability that had previously been unavailable. The personal computer enabled new tools for independent music making, and in the 1970s *The League of Automatic Music Composers* emerged who, taking influence from musicians like John Cage, wanted to investigate new techniques in electronic music. *The League* as they were sometimes referred to, "arose within a tradition of cooperation and self-directed, self-designed electronics uniquely configured for the expression of individual pieces" (Gresham-Lancaster, 1998, p. 40). Their approach to music making involved individual members composing their own pieces of music via computer generated instruments, and later these separate pieces would be performed and adapted together into a final composition guided by their unique interactions (Gresham-Lancaster, 1998). The individual pieces had to conform to specified configurations and procedures as "scores consisted of sets of specific instructions that related the technical requirements to obtain the results the composer wanted to realize” (Gresham-Lancaster, 1998, p. 41), but like John Cage’s works the effort put into creating a final composition was based on one-of-a-kind interactions between the musicians and the music that was difficult to replicate and reconstruct.

In the 1980s The League benefitted from the implementation of MIDI (Musical Instrument Digital Interface), allowing more precision and control generating musical sounds as computers became capable of exchanging control messages with audio producing devices as opposed to just transmitting audio streams. Also in the 1980s The League of Automatic Music Composers changed their name to *The Hub*, and in 1985 performed their first network-based concert from two remote locations within New York City (Gresham-Lancaster, 1998). There were 3 musicians at each site (6 total), and they played simultaneously over phone lines connected via a modem (Gresham-Lancaster, 1998). The performers individually performed their own computer-generated pieces, but their computers were networked to a common, shared memory space within their hub. This allowed the musicians to exchange information with each other during the performance and additionally allow them to mutually influence of the overall musical performance in real-time (Gresham-Lancaster, 1998). The League of Automatic Music Composers and later The Hub helped usher in the genuine concept of musical composition and collaboration over virtual networks.

One example of a popular form of musical expression using networked technology is the music conducting system, where an active listener engages with a piece of music and uses a control system interacting with a musical producing device to add, subtract, or enhance features of the sound. In 1975, Max Matthews created his *Conductor Program*, a system using networked batons that allows a musician to control or “conduct” the playback of a MIDI file (Fabiani, et al., 2013, p. 59). The original design used a mechanical baton to strike a mechanical plate, and the electrical pulse generated from this action sent control information that determined the tempo of the MIDI file playback (Fabiani, et al., 2013). Later, Matthews upgraded his program to incorporate the *Radio Baton*, which “involved two sticks whose 3D position in the space above a plate is measured by an antenna array, contained in the plate itself” (Fabian, et al., 2013, p. 59). The batons could be customised to interact with the music in a variety of ways, but typically one baton was used to keep the tempo of the piece and the other was used to control volume levels or dynamics.

This Conductor Program was one of the first programmable, networked systems that "allowed real-time control of a music performance through the orchestral conductor paradigm" (Fabian, et al., 2013, p. 59).

Figure 2.4 Max Matthews and the Conductor Program (Fabian, et al., 2013).

In the early 2000s, music conducting-like systems were implemented into mainstream gaming platforms with huge success. In 2005, *Guitar Hero* was released where the player controls a guitar-shaped game controller and needed to press the correct button sequence at the right time based on the tempo of the song to score points and hear the full quality of the music. While the gamer was not creating music or controlling the playback of the song itself, Guitar Hero did allow the user to interact and creatively manipulate characteristics of the songs. Its popularity led to spinoffs which included more instruments, such as *Rock Band*, and implementations of networked conducting systems were also seen on other control systems using different platforms, such as with Nintendo's *Wii Music*.

Figure 1.5 Guitar Hero controller influencing music playback (ekawa, 2016).

2.2.2 Modern Applications of Networked Music

Arguably the biggest enabler of networked music was the emergence of the Internet. The Internet made it possible for people to share data, whether it was a music file or audio stream, to large groups of remote clients with speed and ease. In 1999 the peer-to-peer music streaming service, *Napster*, was created and established a huge marketplace where users can share and download music files for free (Evren, 2015). The service was largely opposed by the music industry because accessibility to free music took a financial toll on musicians and record labels. However, it did revolutionise a new open market for music distribution. As a response, some of the larger music corporations began to work together to create online music stores (one example being *MusicNet*) that allowed users to either submit membership payments for their service or pay a fee to download a song. These steep prices were not well-received by consumers, however, and musicians felt they were not receiving adequate royalty payments for their music (Evren, 2015). In 2001, *Apple* was able to capitalise on this market by releasing the iPod and iTunes. Users were able to upload CDs and other legally purchased digital music files to iTunes and make copies of their music on CDs to distribute to peers or simply store the music on an iPod for mobile listening (Borenstein, 2008). With the creation of the iTunes music store in 2003, consumers were able to legally purchase digital downloads of songs (at only $0.99 per mp3) or full CDs to iTunes directly, and it successfully created a centralised hub where people can purchase and manage their music libraries altogether (Borenstein, 2008). Following Apple's achievement, other large companies, such as *Google* and *Amazon*, began to follow suit with similar models of online media distribution hubs.

Apart from online marketplaces, the Internet created opportunities for distributed digital music streams and the growth of online radio stations. In 2004, *Pandora Radio* was able to revolutionise the online radio station by creating a service that generates an automatic playlist adapted from user-specific listening tastes (Evren, 2015). When a listener searches for an artist, song, or genre of music, Pandora uses complex algorithms to analyse the song choice and make a playlist that includes songs by the same or similar artists as well as songs with similar sonic attributes. The user can "like" a song they felt was a good addition to the playlist to help tailor the listening preferences and "dislike" a song that they did not desire or want to hear again. Subsequent companies like *Spotify* followed this trend, allowing users to stream songs and albums on their platform, and using money from adds (on their free services) or membership fees to pay royalties to companies and music artists (Hayes, 2019). This paved the way for more advanced applications like *Tidal*, which includes an option for high-fidelity, lossless music streaming for their users. While these types of online services provided a means for the record industry to distribute and monetise commercial music, modern research has recently been shifting to uncover ways for consumers to creatively share and express their own, unique music with peers.

The evolution of computing systems and the Internet's growing ability to handle greater processing resources over time has opened doors for experimentation with newer networked music applications. Some of the current investigations into networked music applications seek to understand achievable forms of virtual music performances and improve areas of remote collaboration. Schober (2006) examined a project by the New School of Social Research that investigated the requirements needed for musicians to perceive a virtual space as a proper, compositional environment. Notable issues among collaborating musicians when performing in remote environments are being unable to receive subtle, non-musical cues typically present while located in the same space in addition to timing coordination during performances (Schober, 2006). The advent of video conferencing helped facilitate these issues by providing both audio and visual feedback to the performers. However, a limitation to video conferencing is broadband capacity, where speed and video degradation are affected by bandwidth.

Other areas of interest in networked music evolved to using technology to express musical objects. This is done by the embodiment of intelligent machines. As part of the CHI 2015 workshop with emphasis on human-computer interaction, Grote, Anderson, and Knees (2015) provided some areas of focus including embodying machines. One area of focus involved understanding the physical structure of instruments, such as what traits make them perform as they should and how external sources (such as the human musician) affect functionality. Another area included understanding how a machine interprets ways in which physical musical equipment operates. This means exploring what it takes for the machine to comprehend the interaction between the human and musical equipment, or simply how the machine itself experiences music. By enabling intelligent machines to embody instruments, "musicians are holding two parallel understandings of music simultaneously: On one hand there is the full embodied experience of playing and simultaneously experiencing music and on the other hand there is the understanding of how the machine or instrument understands the sounds in use" (Grote, Anderson, and Knees, 2015, p. 2346).

2.3 IoT Implications for Audio and Music

2.3.1 Moving Towards IoT-Enabled Music Applications

The IoT framework provides major opportunities for new music applications by turning music technology (including instruments, recording devices, and music processing equipment) into physical nodes that have network capabilities for communication and control. Some recent efforts have gone into developing practical, user-friendly objects that create unique and expressive interpretations of musical data, or finding methods to sonify complex data streams. Companies like *Lucie Labs* in France and *Rescued Ideas* (developers of the *Basslet*) in Germany have created music influenced wristbands, the former used to monitor the pulse of music and user movement to gauge crowd engagement at concerts and the latter which vibrates to allow the wearer to feel the pulse of the music throughout their entire skeleton (O'Brien, 2015).

Additionally, artificial intelligence (AI) has been paired with the Internet of Things, allowing networked devices to make complex computational decisions that create unique, data-driven experiences influenced by audio and sound. Popular AI enriched devices like Google's *Assistant*, Amazon's *Alexa*, and Apple's *Siri*, utilise IoT-based voice control to create 'smart speakers,' which serve as personal assistants that address many consumer needs like scheduling appointments and services, answering queries, and facilitating other daily tasks. As of 2018, smart speakers are reportedly owned by nearly 20% of adults in the United States (Kinsella, 2018), and more smart IoT devices have been developed that specifically impact music engagement. One such product is the *Prizm,* a small, pyramid shaped music player developed in France that can connect and play music from almost any modern streaming service (O'Brien, 2015). Prizm is instilled with machine learning allowing playlists to be configured based on preferred music types as well as updating chosen genre types around different times of day and the user's listening patterns. The playlist can also be customised upon the shared preferences of multiple individuals in the room and determine the atmosphere of the crowd based upon the ambience of the room.

Determining new and unique ways to implement IoT technology in practical and creative ways offers more challenges outside of simply creating new IoT objects. Whalley (2015) feels what currently is described as "telematic music" focuses too much on old, traditional forms of composer-leader based performance architectures; and that IoT technology can go deeper allowing users to directly engage each other and devices for added collaboration and creativity. With so many pervasive devices finding their home on the Internet, he remarks:

> *Our human-focused view of time/space/location in relationship to networks is then likely to become secondary to information-centric networks that interface with human needs and networks inhabited by ubiquitous intelligent machines that can interact with people and other machines* (Whalley, 2015, p. 93).

He feels not enough attention is given to the role of intelligent machines, who themselves can add another layer of complexity to music composition and become complementary tools to aid the needs of musicians.

Furthermore, for IoT music to break new grounds there needs to not only to be a focus on the enabling technology, but collaborative input of what musicians want to achieve and what smart and embedded machines can deliver. In an *Ask Audio* article, Liam Lacey (2015) offers 4 areas within music technology and production can be influenced by IoT:

1. **Remote Performances**
   IoT is allowing musicians to replicate performances remotely that could only traditionally occur in the same environment. There are a number devices and software that now enable the distribution of high resolution audio in almost real time. Lacey (2015), however, alludes to a scenario where performers desire to control physical instruments remotely instead of just sending audio streams. This would require sending actual performance and control data to a control system that interacts with an instrument to create natural, desired sounds (Lacey, 2015).

2. **Remote Recording**
   Remote recording often occurs with two or more musicians physically performing in different locations and having their individual performances recorded and then recombined at a later stage into a final mix. Lacey (2015) offers an idea of being able to remotely record an instrument in its natural setting (i.e. embedding a church organ with technology so that it can be played and recorded remotely to capture the cathedral sound) and being able to deliver the audio back to a studio for editing and mixing.

3. **Remote Live Mixing**
   In the area of live mixing, there are a number of networkable mixing consoles that support virtual inputs and outputs to transmit data as well as be controlled using a graphical interface operating over the same Local Area Network as the consoles. However, these mixing boards do not currently allow for control over the wider Internet. Lacey (2015) proposes a solution of having a set of networkable mixers

that can be accessed and controlled through browsers or other interfaces over Internet, thus allowing engineers to easily connect and disconnect to desired mixers when needing to work different engagements.

4. **Generative Music and Algorithmic Composition**
   Lastly, Lacey (2015) explains that the Internet of Things can play a larger role in generative music and algorithmic composition due to the massive amounts of data acquired from sensing devices. Data recovered and recorded from the environment through embedded devices and sensors can be mapped to musical tones, scales, and frequencies, providing interesting and unique forms of musical expression.

2.3.2 Modern Applications of IoT in Sound and Music Production

An evolving high-speed Internet continues to provide more robust means of virtual data transmission. Networked music based research has already aided in sharing higher resolution, bi-directional audio, video, and data control for performance and compositions with low latency speeds. The SoundWire group at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University created *JackTrip,* a music software that uses high-speed education and research networks to distribute multitrack, high-quality, uncompressed audio across the Internet with low latency (Cáceres and Chafe, 2010). JackTrip supports *online jamming,* where remote musicians can play together in-real using the Internet and stay in sync without noticeable delays. The *LoLa* (abbreviated for LOw LAtency) audio visual streaming system developed by the Conservatorio di Musica Giuseppe Tartini in Italy (Drioli, Allocchio and Buso, 2013) added a video component to real-time performance applications, boasting 20-50 ms of latency and allowed a seamless concert to occur between a clarinettist in Edinburgh and a pianist in London in 2012 (Ferguson, 2013). This led to subsequent music performances across Europe as well as between the UK and US to occur with visual and auditory cues available across the sites (Moir, Ferguson, and Smith, 2019). *Open Sound Control* (OSC), developed at the UC Berkeley Center for New Music and Audio Technology, is a modern communication protocol that supports the exchange of low-latency data between computers, sound synthesisers, and other media devices over computing networks (Wright, 2005). Primarily use cases for OSC have been interactive computer music applications that permit remote control of musical devices and networked music performances. Furthermore, protocols for networked audio streaming and open control architectures have been developed as part of the official AES (Audio Engineering Society) standards. AES67 specifies recommendations for interoperability between media networks and streaming protocols (AES Standards Committee, 2018a). AES70 establishes a scalable framework for controlling and monitoring network connected devices, and specifies attributes such as how these devices are connected, defined media stream paths between the connected devices, and control and monitoring configuration parameters (AES Standards Committee, 2018b). Open control architectures paired with network streaming offer unique solutions for engagement with networked and interconnected music systems. With the evolution of high-speed computing networks and rise of applications for real-time transmission of media and control data, Moir, Ferguson, and Smith (2019) admitted “we are confident that the traditional barriers to network performance no longer stand in the wary of real-time, remote, interactive performance and recording.”

In addition to the aforementioned developments in distributed music performance and control, some organisations have begun utilising the Internet through the sharing of resources between individuals, individuals and smart programs, or embedded devices to create practical music solutions to facilitate music production and composition. Companies like *LANDR Audio Mastering* (http://www.landr.com) and *Sage Audio Mastering* (http://www.sageaudio.com) have begun using the Internet to optimise mixing and mastering. LANDR’s mastering algorithm was composed by a combination of musicians, signal processors, and an astrophysicist to use artificial intelligence and adaptive listening mechanisms to learn and make subtle, frame-by-frame adjustments to audio files uploaded to the site in order to digitally master the audio. Sage Audio allows users to create an online profile and upload a recorded piece of music directly to the site where a live audio engineer then master the file before returning it back to the user. The site offers a free trial where users can submit their audio and hear a 1-2 minute comparison of the original vs mastered file.

In an IoT realm regarding interconnected, controllable hardware, one example is the *Patchwerk,* developed at the Massachusetts Institute of Technology (MIT). Patchwerk consists of a large, networkable synthesiser that can be controlled physically in-person inside the MIT Media Lab or through a web interface online. The original, stand-alone analogue synthesiser was built by Dr. Joseph A. Paradiso in the 1970s and 1980s and, due to its size, became largely underutilised as it was complex to operate and confined to the Media Lab. In order to create Patchwerk, a group of postgraduate researchers added custom upgrades to the synthesiser, using the digital and analogue outputs of an embedded computer to electronically manipulate its functionality and designing a web-based interface to send Websocket commands allowing virtual engagement from remote users (Mayton, Dublon, Joliat, and Paradiso, 2012). The synthesiser could additionally be switched into a read-only mode so that patching can occur on-site while virtual users were still provided visual and auditory feedback. Virtual control of music devices were eventually implemented into commercial hardware, such as Tegeler Audio Manufaktur's *Schwerkraftmaschine*, a vacuum tube-based analogue vari-mu compressor, which has knobs fitted with servo motors allowing control via the Internet or a plug-in in addition to in-person engagement (Tegeler Audio Manufaktur, n.d.). Additionally artistic applications emerged using IoT-enabled music hardware, such as the Google *Universal Orchestra*. The Universal Orchestra, part of the Chrome Web Lab, was a series of five museum exhibitions that sought to combine virtual and physical spaces. The Universal Orchestra allowed users to play 8 different percussive instruments in real-time, and the instruments could be controlled in either the museum or online through a web browser.

Other companies found unique ways to market music hardware engaged through the Internet. *The Audio Hunt* (http://www.theaudiohunt.com) allows a community of users to share their analogue audio hardware equipment and/or their audio skills for signal processing. Members who do not own a particular piece of hardware equipment can pay other members to borrow their devices for a specified time. Remote users can pass along audio files and hire other device owners to process audio files on desired equipment. After processing, the edited audio can be digitally sent back to the customer. *mix:analog* (http://www.mixanalog.com) is one of the latest instalments in IoT-enabled music systems and allows users to interact with bespoke, analogue audio devices that have custom-made modifications granting them remote control through a virtual, online environment. Their web-based control system mimics the appearance of the desired analogue music device chosen by the user, and additionally provides real-time VU and audio monitoring of the processed sounds. Personal music tracks simply need to be uploaded to the mix:analog application, and their servers deliver the audio to the analogue devices for processing. Afterwards, the new audio can be bounced and downloaded when the mix is completed.

These examples demonstrate the capability of the Internet and IoT in granting musicians and music producers opportunities to mix, record, or process music remotely using the Internet or engage with physical devices through virtual connections. IoT can shape future practices for developing music and lead to new, networked-based workflows in music production.

2.3.3 New Paradigms Inspired by IoT-enabled Music Composition

The popularity of personal computers, digital audio workstations (DAWs), and in-the-box music processing software in the 21st century has "brought to the territory of home practices that were considered the norm in studio production… technologies and techniques once reserved almost exclusively to professional studios" (De Carvalho, 2012). Some of the older methods of professional music production relied on analogue hardware typically found in commercial music studios, but the advent of the digital age allowed many popular elements of the music studio to become "reduced to a laptop" (Dixon, 2016). De Carvalho (2012) mentions that as a result, many of these traditional music applications, "from pre-production to mastering and mixing, could then be performed from one's house."

Consequently, the standard view of the 'music studio' continues to evolve as more individuals are opting to work in the security and isolation of their own private spaces, promoting the concept of the *bedroom musician*. Jonze (2010) describes the bedroom as being more than just a studio for some, but "essential to their whole aesthetic." The bedroom not only provides a new escape where these artists can feel freedom, but an environment where they are more in control of their external identities. There are limitations to this practice, however, "whilst there are many musicians making music in their bedrooms, there are still bands who want their music recorded and do not simply want a whole set of loops and samples editing" (siteadmin, 2013). In some cases artists will spend part of their time recording in a studio using professional quality hardware and other parts mixing at home (siteadmin, 2013).

While software tools have become deeply embedded into modern audio processing, physical devices continue to hold importance in music production and sound engineering. Contradictory views of quality and performance between music producing hardware and software are constantly held in public discussions and forums, and with specific regards to analogue hardware, Strickland (2008) presents the case of audiophiles who believe that digital sampling processes cannot truly replicate the subtleties of a continuous, analogue audio signal since sound is naturally analogue. Evens (2002, p. 172) expands on this topic describing how the vibration of sound at a point in space "adds up to a single, continuous variation in pressure, a wave," and given the complexity of these waves "it is not surprising that engineers are challenged to record and reproduce sound satisfactorily."

Additional appeals to the use of physical hardware in modern production emerge around the aesthetic of analogue composition. In an interview in *The Guardian*, Mack Wilson, tech editor of DJ Mag states that "guys are sticking to analogue because the sound is softer and it brings a bit of the past into the future, which is something that can't be achieved with new apps and soft synths" (Reidy, 2014). Similarly, DJ and producer Alexander Green states:

> *You can't beat a machine that's 20 years old where the circuits have degraded and it has its own character. It may sound a bit romantic but you can't perfectly emulate analogue equipment with software…. Maybe it's the history behind the instruments, maybe it's the authentic sound of the machines. Or perhaps it's just me* (Reidy, 2014).

IoT processes have opportunities to create novel shifts in modern music by extending production workflows to virtually accessible audio hardware. Having greater options to access physical music devices utilising IoT may present an attractive option to individuals who desire analogue processing in the current, digital-centric atmosphere, and contributes to the seemingly "analogue revival" shown through the growing popularity of vinyl records (Morris, 2016) and other tactile devices, both musical and non-musical. Furthermore, IoT-embedded musical hardware incites the idea of virtually-extended music studios, where producers can have remote access and digitally engage rare, bespoke, or professional hardware, retaining the ability to obtain analogue quality processing from distributed music systems without a need to leave their personal production environments.

3. The Method of Research

The research methodology details the unanswered questions emergent from the literature review and the methodological approach to answering them. In this chapter gaps in knowledge are presented, followed by developing research questions and aims, and lastly a detailed discussion and justification of the practical analysis conducted to address the previously unanswered research questions.

3.1 Research Goals and Objectives

3.1.1 Summary of Gaps in Knowledge

The literature review established a vision of the Internet of Things where an infrastructure of ubiquitous computing devices is interwoven into modern industries, however, IoT opportunities impacting creative industries represent an area where improvements can be explored. The chapter additionally illustrated that creative use cases for IoT often emphasise unconventional methods to generate artwork and artistic displays as opposed to offering tools for the practical development of projects and workflows by creatives. For musicians, this can reflect having greater options and techniques to produce music in comparison to what is physically available in the current market. As such, one implicit gap of knowledge is the need for original and critical investigations directly exploring IoT architectures applied to music production where opportunities, benefits, and challenges are evaluated and assessed within an academic context.

Understanding IoT's ability to impact music production first requires developing a better comprehension of current IoT architectures that can help bridge musicians to remote production devices. Specifically, this involves exploring and evaluating technical components that can be adapted into analogue and hardware devices in order to facilitate meaningful interaction with these remote devices. Once an IoT-enabled music system can be realised and achieved through modern techniques, additional gaps arise regarding the impacts of these systems on existing methods of music production. The concluding sections of the literature review indicated that there is still an appeal for analogue and physical hardware in modern music production, so it is important to understand whether a virtual music system allowing remote access to physical devices can effect current production workflows and have further influence on the creative processes in which music is generated. This requires open and in-depth conversations with both amateur and experienced music producers to determine any emerging values from these systems, in addition to uncovering notable opportunities and measurable outcomes they may provide.

3.1.2 Research Questions

The aforementioned gaps in knowledge help draw a number of unanswered research questions:

RQ1: What are the current capabilities of IoT infrastructures to support distributed audio system networks, and what improvements can be identified and evaluated?

Answering this question involves examining current Internet architectures that can support robust networked audio systems, which includes control infrastructures that allow real-time manipulation of physical systems over extended networks and computing protocols that allow for low-latency transmission of control and media (audio and visual) data over the Internet with high reliability.

RQ2: How can IoT-enabled music systems facilitate new music production engagement, workflows, and collaboration methods?

This question seeks to understand how IoT systems can help musicians better engage and collaborate with musical resources and the impacts they can have on current production processes. New opportunities for networked audio and music processing chains can allow the incorporation of numerous physical, analogue audio processing devices independent of brand, type, or location into a virtually-accessible work flow accessible by remote users. Adding IoT components to rare and high-demand devices can allow them to operate in a hybrid manner, bringing the best qualities of both digital control and natural analogue sound aesthetics.

RQ3: What cultural, enterprise, and creative benefits do IoT- based music systems present?

IoT-enabled music systems may offer opportunities for enhanced accessibility and productivity for music producers from various production backgrounds and expertise, and provide them new tools to help generate and express their musical works. Networkable audio systems can also add increased value to rare and under-utilised hardware, and in unique cases be extended to natural environments where the acoustic sound qualities of physical, material spaces can be added as real-time echo and reverb chambers, in addition to other natural effects. Lastly, networkable devices may open new business markets where virtually-accessible audio hardware can be rented and hired from retailers and collectors.

3.1.3 Aims and Objectives

Key aims and objectives of the research are developed to address the presented research questions and provide a practical roadmap for answering them. These involve researching IoT technologies and infrastructures that can effectively support a distributed IoT audio system, and conducting further testing, observation, and data gathering to understand how an IoT audio system can facilitate novel practices of networked audio and music processing. These evaluations additionally help reveal benefits IoT-enabled music practices can deliver to the audio and music industry.

The specific aims and objectives of this research are as follows:

1. Investigate the emergence of IoT and evaluate cultural and creative aspects with respect to music technologies and audio processing systems

2. Design and prototype an IoT-enabled audio and music processing system with audio streaming and remote control capability

3. Test and evaluate capability for lossless audio data transfer across various broadband Internet networks (i.e. Local Area Network, public Wide Area Network, High-speed Research and Educational Networks)

4. Evaluate opportunities and benefits of IoT-enabled music production systems through public demonstrations and focus group discussions

5. Evaluate deeper creative, cultural, and enterprise opportunities of IoT-enabled music production systems through interviews with audio innovators and music technology experts

3.2 Research Design

3.2.1 Experimental Approach

This research adopts a practical development approach specified by Shakhovskoy and Toulson (2015) illustrated in Figure 3.1:

RESEARCH ⇒ DESIGN ⇒ BUILD ⇒ TEST ⇒ EVALUATE

Figure 3.1 "Research-Design-Build-Test-Evaluate" research method approach.

Shakhovskoy and Toulson (2015) used this methodology for developing and evaluating mobile applications (apps) for the music industry, funded by the NESTA Digital R&D Fund for the Arts programme. Applying the methodology to this research, the research stage consists heavily of information gathering and conducting the literature review. From the literature review, past and present concepts regarding IoT and music production are comprehensively examined and documented, and key challenges, gaps, and opportunities are identified allowing research questions to be posed. The research stage lays a foundation for conceptualising a practical design of a creative work (the IoT-enabled music system) which is then built, tested, and evaluated publically in order to gather professional user data that is analysed for the purpose of answering the research questions and filling in any relevant gaps of knowledge. The results are then critically assessed and discussed, prompting new insights into music production practices augmented by IoT and additional future work that can emerge from the research, and finally widely disseminated as part of the complete thesis. A summary of the full methodology is presented by Patterson et. al. (2015), who also adopted and enhanced the approach of Shakhovskoy and Toulson (2015) for further music app development and evaluation funded by the UK Arts and Humanities Research Council in 2014 and 2017. The methodology is illustrated in Figure 3.2.

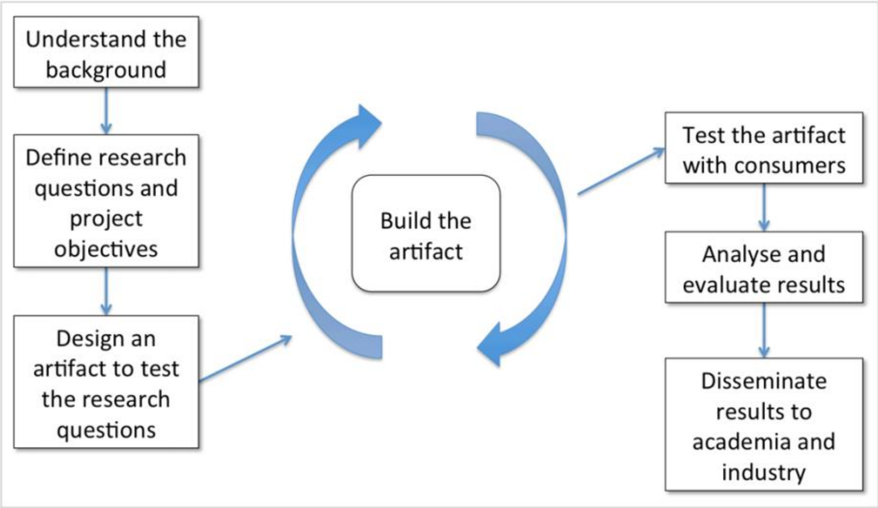


Figure 3.2 Summarised, practice-based research methodology (Patterson et. al., 2015, p. 198).

Patterson et. al. (2015) introduces more detail to each stage of the methodology and emphasises an iterative process in the artefact build phase of the research. At the central stage of the creative practice project, the artefact is rapidly developed and tested regularly with incremental improvements until the final artefact is realised. Having been verified by Research Council funding for music-technology investigations, this methodology is justified as the overarching creative practice presented in this research.

3.2.2 Proof of Concept Design

The overall design of the research is based upon a practical, creative work showcasing a proof-of-concept IoT-enabled music application that demonstrates the abilities of virtual control, real-time audio streaming, and remote engagement with physical and analogue audio systems. The demonstrated IoT music concept is used to collect feedback regarding effective transmission of audio data across the Internet and explore opportunities for unique and novel production practices that such a system can empower. The following sections give insight into the pragmatic design of the research and the processes for data collection and evaluation.

The practical work revolves around a concept representative of an extended, modular virtual channel strip, shown high-level in Figure 3.3. Similar to a channel strip, the IoT-enabled music application aims to provide virtualised tools allowing a user to incorporate any desired audio equipment or other musical resource (i.e. acoustic spaces) into an audio processing chain and enabling remote production work flows independent of each devices' locations.

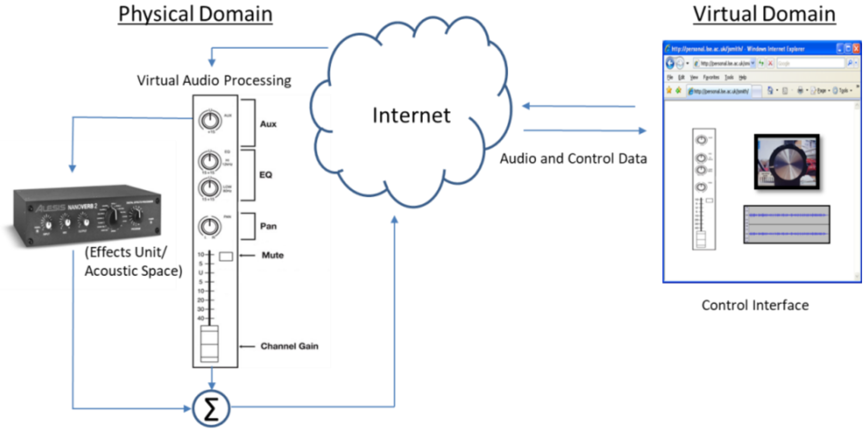


Figure 3.3 IoT-based channel strip.

The application includes a web-based user experience that utilises IoT communication protocols to transmit control and media data to access and manipulate hardware music devices in remote locations, therefore allowing the processing of audio to be centralised, be it through a workspace, laptop, or mobile device, and ultimately making music mixing and composition more global and portable.

To complete the IoT-enabled music application, the adapted interface experience needs to be fitted to network-controlled audio processing devices. One of the audio processors involves an audio mixing board embedded with networked actuators allowing a user to remotely adjust the Hi and Low EQ (equalisation) frequencies of the mixer. This demonstrates the capabilities of augmenting an existing analogue audio processor with IoT functionality, creating a hybrid analogue-digital audio system. An additional focus is the inclusion of a reverberant space, and utilising virtual networks to incorporate unique acoustics of material locations into the audio processing chain. This overall system combining the discreet technical components of both networked control and networked audio is used to analyse and evaluate the benefit of IoT architectures applied to audio and music practices.

3.2.3 Prototype Build and Test

Facilitating the practical nature of this research, the main approach for the design, build, and test stages follow a waterfall-style methodology. The waterfall method detailed by Keith (2010) is a commonly used methodology in product development and follows a list of discreet steps or phases that begin with an initial concept and lead to the testing and evaluation of a finished product. Furthermore, each subsequent phase builds upon the results of the previous stage to accomplish a larger and more complex overall goal.

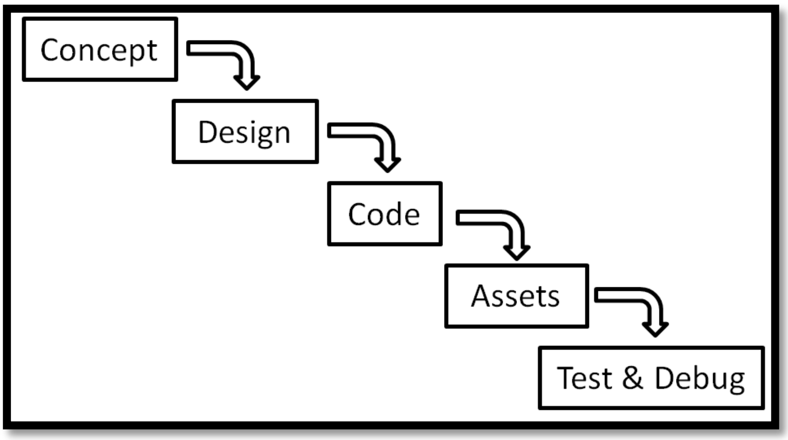


Figure 3.4 Example of a Waterfall game development method.

The overall design and build occurs in stages; once the initial concept is specified, small targets and testing milestones are set, eventually culminating into a final design. Adding an agile design component, the design and build stage incorporates an "inspect and adapt" cycle that analyses the strengths and weaknesses of each developed stage and informs decisions to adapt to new goals when significant changes need to be made (Keith, 2010, p. 30).

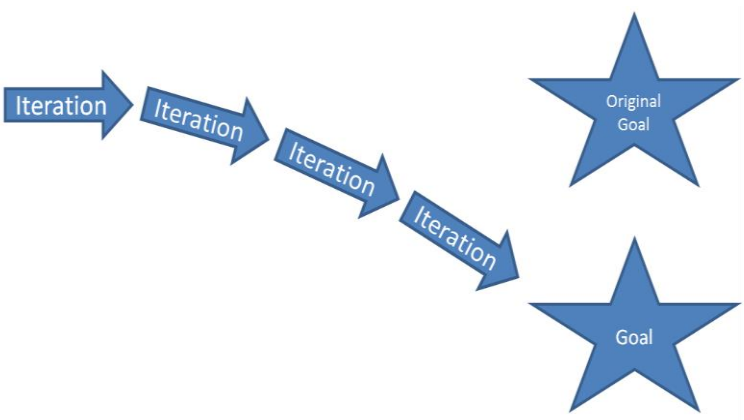


Figure 3.5 Agile "Inspect and Adopt" cycle.

The major stages in the design and build process is laid out below:

Stage 1: Embedded systems and fundamental IoT processes research

The initial research stage (see Figure 3.1) involves exploring emerging developments in the Internet of Things and examining the leading technology to determine which hardware and/or software solutions can sufficiently support remote interconnectivity and control in audio applications. These solutions are based upon low cost computing resources and publically available open source code that are readily available to the general population and do not require commitments to proprietary commercial products.

The key technologies utilised in this research are:

Microcontrollers

The primary microcontroller used for this research is the Freescale FRDM-K64F. It uses a 32-bit core ARM® Cortex™ M4 processor with 1024 KB of flash memory and 256 KB of RAM (ARMmbed, n.d.a). Added benefits to this particular microcontroller is that it comes equipped with an Ethernet jack for physical network connections and provides extension sockets allowing easy connections for external sensor modules and shields that plug directly into the board. This allows prototyping technical designs and concepts to be simpler and more efficient.

HTML5 Websockets

The Websockets protocol allows full-duplex, bi-directional communication between a computing server and client over the Internet. HTML5 Websockets is an application programming interface (API) that is implemented into a web browser enabling the browser to set up direct communication with a server in real-time, and this communication can also extend to other clients (i.e. a networked microcontroller) connected to the server (Kaazing, n.d.). Consequently, HTML5 Websockets set the frame work for establishing the graphical interface to interact with the connected IoT devices and allows the design to be more portable across mobile devices.

Webservers

A webserver takes online requests from a client (ex: a computer user trying to access a webpage), processes the requests, and serves the client with the desired information. Specifically for this research, the webserver mediates the communication between the embedded IoT devices and other networked functions. The webserver used in this research is the *Tornado* webserver and is programmed in the Python programming language.

JackTrip

JackTrip is a Mac OS and Linux based system for streaming high quality, uncompressed audio over the Internet. It can support a max amount of audio channels governed by the computers capacity, and can manage bi-directional audio streams (Caceres, 2007a).

WebRTC

WebRTC is an open source platform that allows real-time communication of audio, video, and data over web browsers and mobile devices. WebRTC incorporates APIs such as getusermedia allowing access to a computer's internal and external media devices to transfer media data over the web. While WebRTC contains its own APIs for capturing media, it relies on other resources such as signalling and NAT and firewall traversal to share data between peer computers.

Motors/Actuators

The use of motors serves as the bridge between the digital and physical components. Motorised potentiometers in particular allow voltages, commands, and input values from digital sources to be interpreted into physical movements that can control the rotation of a knob or the sliding of a fader while still allowing physical, hands-on interactions to achieve the same effect. Motors connected to microcontrollers allow virtual interactivity with physical components.

Stage 2: Design and prototyping IoT control systems

Two microcontrollers were chosen for this research: the mbed LPC1768 and the FRDM K64F. The mbed LPC1768 was one of the original development boards designed by ARM and offers a great deal of developer support for application prototyping. The FRDM K64F is designed by Freescale, but runs ARM's M4 Cortex processor and offers 1024 KB of flash memory (twice that of the LPC1768) and 256 KB of RAM (8 times that of the LPC1768), making it powerful and capable of managing a range of immediate computing requests (ARMmbed, n.d.a). For transmission of control information, HTML5 Websockets stood out as a robust communication platform to support a virtual control infrastructure between a user interface and microcontroller processing unit. Websockets allow for bi-directional communication over the Internet between a server and a client, and can support a HTML5 web-based graphical user interface.

The current webserver is programmed in the Python programming language and incorporates Websockets headers to allow transmission of commands between a server and client. During this stage the Websockets server and client HTML5 webpage were created and configured to establish an interface for Internet-based control of the microcontrollers and the physical hardware systems. Upon successful configuration of the microcontrollers with the client interface, actuators were incorporated to demonstrate the capabilities of fully developed, IoT control system that can manipulate controls on a physical audio device.

Stage 3: Investigating, testing, and incorporating audio streaming functionality

Modern audio streaming platforms were evaluated for their effectiveness in audio transmission to support an IoT-enabled music application. The JackTrip application was eventually selected as the primary audio streaming application in this research after conducting local and wide area streaming tests to evaluate the quality of network-based audio transmission and robustness in managing network data loads. Application testing showed that JackTrip accurately maintains the qualities and properties of source audio files when streaming over the Internet, making it the optimal for a networked music application. Initial tests delivered several instances of audio dropouts and errors in the streams, however, further evaluation discussed later in this research determined that JackTrip performs better over specific types of networks (i.e. local area and high speed educational networks) as compared to traditional, public networks. JackTrip was compared to WebRTC, another real-time audio streaming platform used primarily for Skype-like voice and video applications within web browsers, and ultimately outperformed WebRTC in evaluations conducted in this research. Details of the audio streaming evaluations are presented in Chapter 5.

Stage 4: Design and prototyping hybrid analogue-digital IoT audio systems

The last design stage combines elements of the IoT control systems and audio streaming phases to create a final IoT enabled audio system. Main focuses involve centralising control, audio, and an additional visual element into an integrated user experience. The prototyped IoT audio system focused on two audio processors: a physical, analogue music device that would be embedded with computing hardware creating a hybrid, digitally-controlled analogue device and the second being a unique space whose acoustic qualities could be used for a real-time reverb application. This is discussed in further detail in Chapter 4.

3.3 Research Evaluations

The final analysis of the research is evaluated using mixed-method data collection which account for technological capabilities supporting an IoT music system and public feedback regarding the implementation of such a system. The first set of research evaluations involves analysing functional attributes of real-time, high-quality audio distribution. The second set of research evaluations focus on user insight data obtained by feedback from qualitative professional interviews and open and closed-ended participant survey questionnaires. The two evaluations validate the research on two tiers: first by providing data showing the effectiveness of emerging technology to support high-quality, real-time audio transfer between IoT enabled audio systems with detailed replicable observations, and second by offering feedback regarding user impressions of broader, overall IoT audio systems and insights into whether distributed physical audio devices give greater value to analogue music production and enable innovative practices into how physical audio processes are approached. The data aims to validate novel practices emerging from the fusion of IoT and music, and give insight into future implications that should be considered through continued research, such as ideas into enhanced designs and accounts of interests and issues that can be adapted and rectified in the future. A deeper context of the research evaluations are described in the next two sections.

3.3.1 Audio Streaming Evaluations

A successful IoT audio system is one that supports professional quality audio, so the capability to deliver high-resolution audio streams with low-latency across the Internet presents an important characteristic of such systems. The audio streaming evaluations firstly compares the effectiveness of two platforms, JackTrip and WebRTC, for enabling real-time audio streaming for remote production applications and secondly examines the performance of audio transmission across different types of broadband networks.

An audio testing package containing 3 files: a 1000 Hz sine wave, a 0-22.5 kHz sine sweep, and a music sample are used for replicable observations of the audio quality transmitted over the network. Two computers conduct 5 streaming trials comparing the following scenarios: WebRTC and JackTrip, JackTrip over public vs high-speed educational networks, and JackTrip over Wi-Fi networks vs JackTrip over Ethernet. The final analysis follows a primary quantitative approach with embedded qualitative research process discussed by Creswell and Plano Clark (2007), as the recorded audio streams are compared to the original audio source file using spectral analysis with additional mathematical comparisons of performance, including the number of dropouts, total harmonic distortion and noise, and roundtrip latency, being conducted and discussed for each trial set. A qualitative listening test has also been implemented to give subjective, unbiased public feedback regarding the audio quality perceived from recordings of music samples captured from both JackTrip and WebRTC platforms. The full audio streaming evaluation is provided in detail in Chapter 5 of the dissertation.

3.3.2 User Insight Evaluations

Marshall (1996, p. 552) explains that while “the aim of the quantitative approach is to test pre-determined hypotheses and produce generalizable results,” qualitative studies primarily aim to “provide illumination and understanding of complex psychosocial issues and are most useful for answering humanistic 'why?' and 'how?’ questions.” The user insight evaluations are primarily qualitative in nature, and seek to obtain rich and meaningful data information regarding individuals’ experiences and perceptions of IoT-enabled music applications. The feedback was initially collected following a brief talk and demonstration of the proof-of-concept IoT music application implemented in the design-build phase of the research. This consisted of a 10 minute seminar or presentation discussing the aims and approach of the research and included a 5 minute, real-time demonstration showing remote control of both physical processors (mixing board EQ and reverb effects unit) and live manipulation of an audio stream as it is transmitted back and forth between the processors and demonstration venue. Immediately following the demonstration, surveys were presented to collect direct feedback from the attendees. As the research evolved over time an online webpage providing a summarised context of the research, live video demonstrations, and an embedded questionnaire was created at http://mjhardin.com/iotsurvey to replace the live demonstrations and provide a virtual method for collecting a remainder of user insights from remote respondents.

Curtis (2016) describes two types of surveys:

1. Questionnaire based – Involving sets of either open or closed-ended questions answered independently by respondents, and

2. Interview based – Involving asking a set of prepared questions to an interviewee and noting the responses.

Surveys are widely accepted standards in social sciences as they allow information gathering "from a smaller group for the purpose of gaining an understanding of the nature of the group as a whole" (Curtis, 2016, p. 322). With regards to questionnaires, the attendees are provided a survey questionnaire of 10 questions (with an 11th optional question for further comments) that seek to understand their background in music production and how the fusion of IoT and music could impact their approach music production if these types of technical applications became widely available to the public. The respondents are broken down into two core groups, experienced music producers and casual music makers, however, the individual identities of the respondents are kept anonymous.

In addition to the questionnaires, a select group of 7 individuals with professional expertise in music production and sound engineering underwent interviews to provide greater qualitative feedback regarding additional insights and research applications of an IoT-enabled music system. The interview questions are modelled after the questionnaire questions but with the explicit focus of seeking open-ended responses and encourage more conversational feedback amongst the interviewees.

Three advantages to survey research presented by Curtis (2016) are that they:

1. Offer better generalisations of a whole population compared to other research methods as a result of large data sets collected from "real-world situations,"
2. Allow for comparisons to be drawn from people who can share opinions and experiences anonymously, and
3. Are flexible in being able to derive both quantitative and qualitative data as a result of question design.

The survey questions are evaluative in nature, focusing on an implemented change in which music, particularly involving remote musical production systems, can be engaged and explored, and are clear and neutral in their presentation (Curtis, 2016). The analysis of the data is split into two parts: a statistical breakdown of the closed-ended questionnaire responses reflecting the musical identity of the respondents and thematic coding for the open-ended questionnaire questions and interview responses. As the questionnaires produce as large volume of data, the closed-ended responses are stripped down into cross-tables to clearly and concisely show the key findings (Statistical Service Centre, 2001). The tabulated data is represented as percentages comparing the responses of each target group as representative of the whole and is accompanied by a brief discussion of the results for the given question.

The thematic analysis of the open-ended feedback involves searching across each individual response and observing repeated patterns or particular topics of interest in the data sets, subsequently finding meanings in these patterns. Theme selection tends "to be driven by the researcher's theoretical or analytic interest in the area, and is thus more explicitly analyst-driven" (Braun and Clark, 2006, p. 84). In the research, theme categories are driven directly by the second and third research questions:

RQ2: How can IoT-enabled music systems facilitate new music production engagement, workflows, and collaboration methods?

RQ3: What cultural, enterprise, and creative benefits do IoT- based music systems present?

A methodical outline for conducting thematic analysis presented by Braun and Clark (2006) with additional feedback derived from (Vaismoradi, et. al, 2016) is provided below:

Familiarising yourself with your data: This involves transcribing the interview data if possible, but also the researcher immersing themselves in the transcripts so to be aware of meaningful and reoccurring concepts that are presented.

Generating initial codes: This involves identifying and labelling texts, passages, and data concepts that produce interest insight and may address relevant research topics. The process of coding "reduces the amount of raw data to that which is relevant to the research question, breaks the data down to manageable sections, and takes researchers through the transformation of raw data to higher-level insights or abstractions as the development of theme" (Vaismoradi, et. al, 2016, p. 104).

The coding process will particularly revolve around responding to the subsequent research questions.

RQ2: IoT Workflows: Understanding current workflows of musicians and understand how IoT could influence their practices in the future.

RQ3: User Benefits: Examining in what ways respondents feel IoT can impact the field of music production.

Searching for themes: This process consists of collating similar codes and identifying relevant themes for these codes. After initial codes are observed, Vaismoradi, et. al

(2016, p. 105) states that “if a group of codes are repeated in a patterned way in multiple situations, they have potential to become a theme.” Ultimately the more times a similar code is found across the data set, the more likely the code is considered to be a theme.

Reviewing themes: Selected themes should accurately relate to the collation of codes while additionally making sure the themes reflect important topics in the research.

Defining and naming themes: Clear names of the themes should be presented with relation to how they fit into the research. The defined themes are detailed in a final report.

Producing the report: This is the process of analysing each theme and extracting useful information and points that present valid arguments toward the research questions. The final analysis sums into a scholarly report providing clear names and definitions of the themes as they are related to the research.

The use of surveys in the data collection process for this research is most effective due to the subjective and personal experiences associated with music. However, it is important to note that “no survey research, however, is perfect; instead it is a careful balance, maximizing the advantages while minimizing the flaws” (Rea and Parker, 2012 cited in Curtis, 2016, p. 325).

4. Design & Build

The design and build chapter highlights the efforts of a practice-based creative work, shadowing the development of a proof-of-concept, IoT-enabled music system that offers new opportunities and unconventional methods for music production. The chapter gives detailed insights into the practical build of the overall system; underlining key technologies and the testing and evaluation of each development stage, and culminates into a full implementation of the IoT music system.

4.1 IoT Control Systems

One of the design and build objectives of the IoT-enabled music processing system is establishing remote interaction with physical audio equipment, additionally accounting for engagement from very far geographical distances. Within this research, two key elements aim at facilitating this type of interaction:

1. Embedding control and networking hardware into music devices so that they can be controlled virtually, and
2. Creating an intuitive user experience that uses network-delivered commands to interact with the interconnected music device.

The overall control system incorporates motors and actuators that can create analogue movements of physical knobs on a device and a web-based user experience that contains virtual buttons and other interactive inputs to deliver digital commands to the actuators.

4.1.1 Microcontroller Selection

At the early stages of the research, microcontrollers were considered in the implementation of the IoT music system to help bridge the digital and analogue components of the control system. Numerous types of microcontrollers currently exist on the market, many of which are tailored to developing and enhancing IoT-based applications. Ultimately, *ARM*-based microcontrollers were chosen due to familiarity, ease of use, and convenience. Specifically, the two microcontrollers explored in this research are the *mbed LPC1768* and the *FRDM K64F*. The mbed LPC1768 was one of the flagship development boards designed by ARM and offers a range of developer support for application-based prototyping. The FRDM K64F is designed by *Freescale*, but runs ARM's M4 Cortex processor and offers 1024 KB of flash memory (twice that of the LPC1768) and 256 KB of RAM (8 times that of the LPC1768), making it powerful and capable of managing a variety of immediate computing requests (ARMmbed, n.d.a). The FRDM development board also has the added bonus of coming equipped with an on-board Ethernet socket for networking capability. ARM additionally supports an online integrated development environment (IDE) which, although requires Internet connectivity to access, allows prototyping to be more portable and flexible across various operating systems.

Interconnectivity

An original research aim was establishing Internet connectivity for the microcontrollers using wireless networks. The *Roving Networks Wifly RN-131c* module (https://www.sparkfun.com/products/10050) was initially selected to connect the mbed to a standard Wi-Fi network. However, early tests produced challenges, as connectivity success varied with different types of Wi-Fi networks. The Wifly module was well-suited for connecting to older model network routers secured by an SSID and Password (see Appendix A). However, further investigations quickly uncovered that at the time of development, networks implementing a captive portal (a network login that requires submission of personal information and/or user passwords into a webpage) provided difficulty for microcontrollers to access, mainly due to lack of a display monitor or input device to view and log into the webpages. Even in the early research stages, these types of security measures had become widely common in many publicly-accessible networks. As a result, wired Ethernet connections were determined to be the best solution to bypass the connectivity issues as most did not require a captive portal or extra security to gain access to the Internet. After incorporating the mbed Ethernet Interface library (mbed official, 2012) into the microcontroller source code (see Appendix B), there were no issues connecting the microcontrollers to the Internet, thus making Ethernet the standard connection protocol used.

4.1.2 Establishing the Websockets Server

The next stage involved exploring platforms that would aid in the development of an interface to send virtual commands to physical devices over the Internet. One of the first communication protocols observed was *MQTT*. MQ Telemetry Transport, or MQTT for short, is a lightweight publish/subscribe messaging protocol designed for low-bandwidth communication, and is optimised to accurately deliver information over constrained, high-latency networks (MQTT, 2009). The publish/subscribe model creates a system where a server, or in many cases another client, publishes messages or other information that can be subscribed to by other clients. A broker sits in between the publisher and the subscribers and filters messages to each subscriber based on the type of message or content the subscriber desires to receive (HiveMQ, n.d.). Aspects of MQTT have been used in popular messaging applications such as Facebook messenger, and even novel scenarios like a "twittering house" (http://mqtt.org/projects/andy_house), which allows IoT devices inside the house to send tweets about their status or states. While MQTT presents an effective tool allowing engagement with many client devices at one time, it was ultimately bypassed as simpler protocols were found that allow direct interaction with individual devices.

Further research led to the discovery of *Websockets*, which became the main communication protocol used to bridge an interface to interconnected devices in this research. Websockets allow full duplex, bidirectional communication between a server and a client and is commonly paired with HTML5 code to provide webpages opportunities for real-time interaction with webservers (Kaazing, n.d.). Websockets is presented as an upgrade to the standard HTTP (Hypertext Transfer Protocol) application. With traditional HTTP standards, a client computing device needs to establish a new connection with a server every time there is a server request for information. This involves sending additional header data that contains information regarding the client device and other relevant data to help facilitate the connection. With Websockets, an initial connection is established between the client and server, however, that connection is held open over a period of time and each subsequent request for information by a client requires fewer resources to be transmitted (Lubbers and Greco, n.d.). Since new connections are not required for each information request, the server can provide swifter transmission of data with lower latency. This proved to be a more effective case for the real-time interactivity requirements of this research.

A Websockets server can be implemented multiple ways, but a combination of prior experience with the Python programming language and sufficient documentation on the ARM mbed website made the *Tornado* Websockets server an ideal server to develop. Being based on Python, Tornado requires the installation of Python 2 (version 2.7 or higher) or Python 3 (version 3.3 or higher) to properly operate. During this period of development Python 3 was not widely implemented and many useful libraries were unsupported in this version. As a result, the developed server required scaling back to

Python version 2 for desired functionality. ARM provided program code for a simple Tornado Websockets Server and an HTML5 webpage that echoes back user text input, demonstrating opportunities for real-time communication between a client and server (ARMmbed, 2012). This ultimately served as skeleton for the interface development going forward.

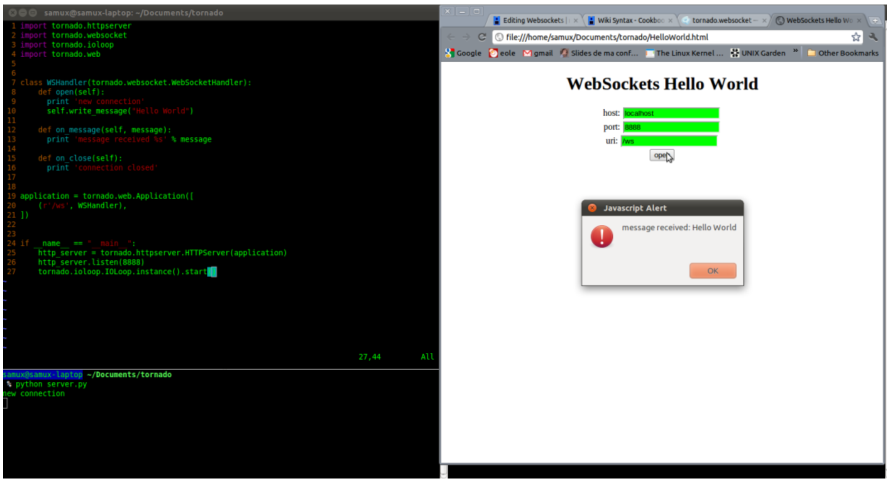


Figure 4.1 Tornado echo back server from ARM (ARMmbed, 2012).

4.1.3 Remote Procedure Calls and Serial Communication

After development of a Websockets server and HTML5 web client, the next aim was to establish communication between a microcontroller and the server, where communication would then be extended from a Websockets-enabled webpage to the microcontroller through the server. Interaction between the Websockets server and the microcontroller was initially established using a method called *Remote Procedure Calls*. A Remote Procedure Call (RPC) is “a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details” (Rouse 2016a). After incorporation into the Tornado server, RPC was used to map Python classes to mbed Libraries that influenced actions and control processes on the microcontroller (ARMmbed, n.d.c). Specifically, when the webpage-based interface sent Websockets commands to the Python server, Remote Procedure Calls were able to translate these commands into actions carried out directly by the microcontroller.

A first test involved using the web interface to turn LEDs attached to the microcontroller on and off. The client webpage interface code was modified to incorporate 3 virtual buttons that sent 3 different string messages to the Python server (see Appendix C). The server code was then modified to use Remote Procedure Calls (see Appendix D) to convert these messages into serial commands that were forwarded to the microcontroller via serial ports of a USB cable. Within the microcontroller source code, the messages were finally interpreted as instructions to turn on 3 different coloured LEDs (red, blue, and green).

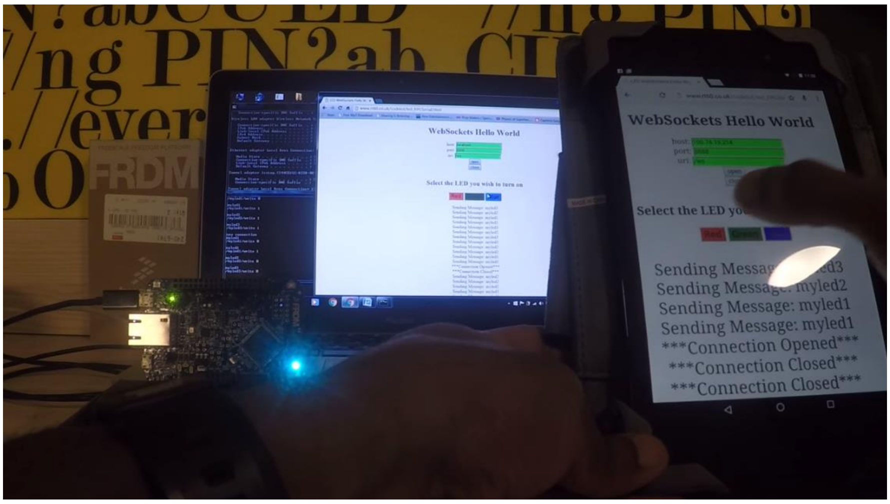


Figure 4.2 Websockets and Remote Procedure Calls illuminating LEDs on a microcontroller (https://youtu.be/k-eqnbvftul).

When one of the coloured buttons was clicked on the webpage, a string of text (either “myled1,” “myled2,” or “myled3”) was sent to the Python server via Websockets. The server would then associate the specific string with the correct RPC variable, and the microcontroller code (Appendix E) would translate the designated procedure call into a specific action performed by the microcontroller (e.g. the variable “myled1” would send a digital output command turning on the red LED1 on the microcontroller).

After successfully transmitting commands to turn an LED on or off, the next step focused on variable communication, such as sending a range of numerical values instead of a binary “on” or “off” command. The HTML5 code was modified to include a slider that could send stepped values between 0 and 1 to the server, and two extra buttons were added to allow the user to choose between two LEDs that the numerical slider values were sent to (see Appendix F). The server was also updated to accommodate the two LEDs and specified the RPC commands that triggered the specific LEDs (see Appendix G). The mbed code was finally modified to include 2 pulse width modulation (PWM) outputs (see Appendix H) that could transmit the slider values to the LEDs to create a dimming effect.

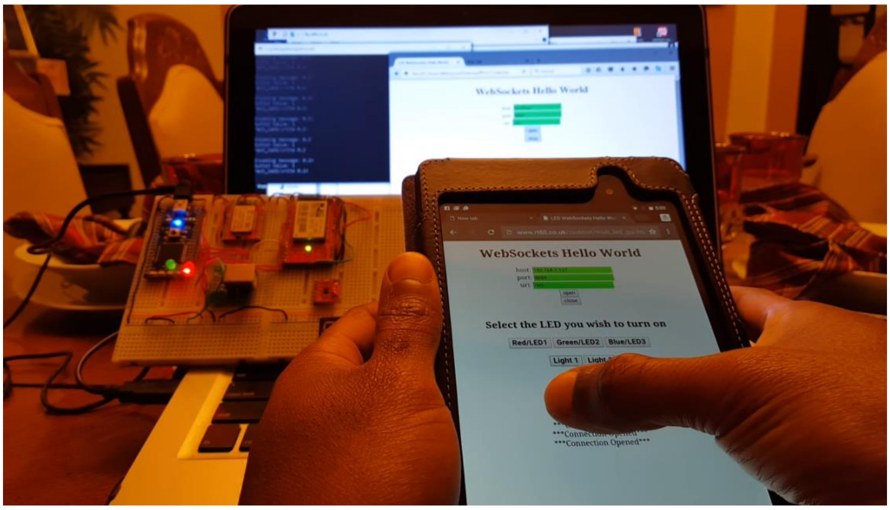


Figure 4.3 Using Websockets and Remote Procedure Calls to dim LEDs with a virtual slider (https://youtu.be/LtTWSq93MTo).

Although serial RPC communication required the microcontroller to be tethered to the server computer by a USB cable, this demonstrated the possibility of transmitting commands between a client web interface and the host server over a virtual network. Furthermore, delivering control messages from the interface to the microcontroller's PWM output was the first step in extending control to actuators and other electrical components that use pulse-width modulation for movement.

4.1.4 Dedicated Websockets Communication

After successfully testing and evaluating network communication with Websockets and Remote Procedure Calls, the next phase focused on bypassing the serial connection between the microcontroller and server and employing a full communication system solely using Websockets. The Tornado server code was simplified, removing elements of the Remote Procedure Calls and only including functions that delivered the client web interface commands to other clients connected to the server (see Appendix I). At this stage, any further development relied on the FRDM K64F microcontroller exclusively due to its on-board Ethernet connection and higher processing power. The microcontroller source code was modified to incorporate the Websockets Library (Mokrani, 2012b), allowing it to connect directly to the Tornado server when the correct I.P. Address and port number of the server computer were provided (see Appendix J). When both the client web interface and microcontroller connected to the server, a user could interact with the slider on the interface to send values via Websockets to the server, where the server would in turn deliver those values to any other connected clients, although only the microcontroller would interpret these values into useful commands to perform desired actions. Building on previous tests, the web interface slider was configured to adjust the brightness values of an LED on the microcontroller.

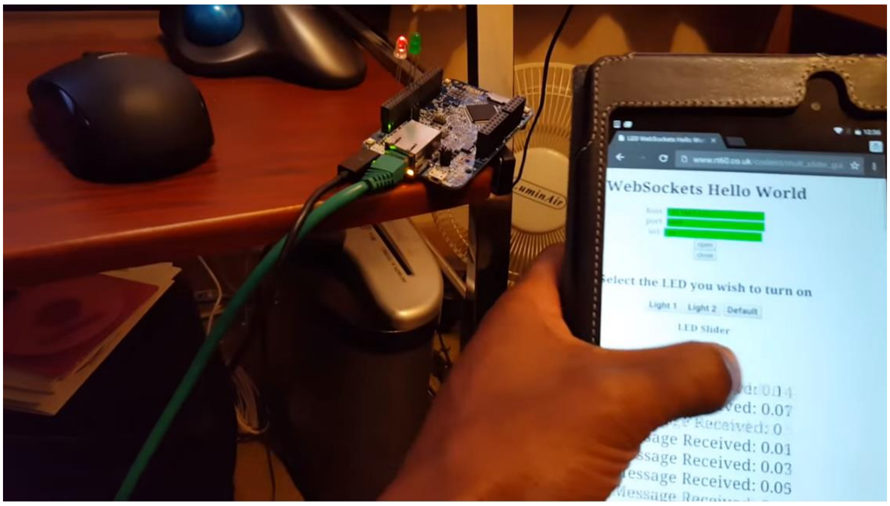


Figure 4.4 Using Websockets to control an LED (https://youtu.be/6TyzcbpnU2Y).

The removal of Remote Procedure Calls effectively demonstrated complete IoT-based communication between a computing server and clients, and the use of the mbed's PWM output to control the LEDs provided the next steps for incorporating motors and actuators.

4.1.5 IoT Actuator Control

Since small electric motors and LEDs can both operate using PWM signals from a microcontroller, the next stage focused on incorporating actuators into the design and build stage. An *Alps RK27* 50 kΩ motorised potentiometer was obtained and evaluated with the expectation of augmenting a movable knob on a physical music device with a rotating motor.

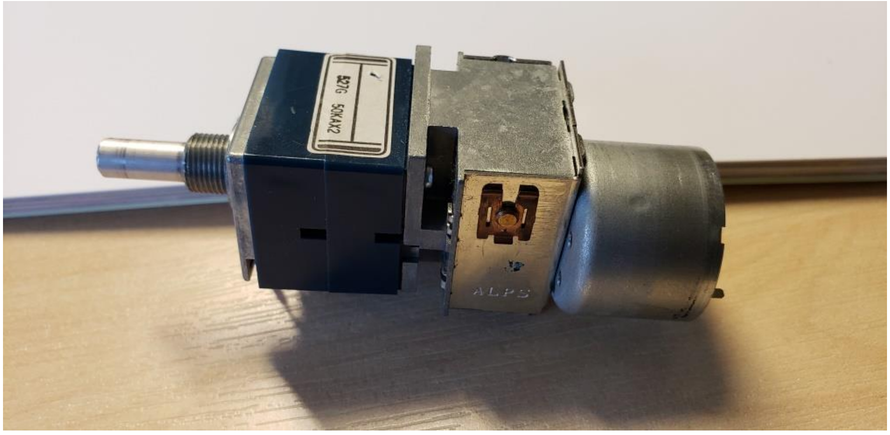


Figure 4.5 Alps RK27 50 kΩ motorised potentiometer.

The motorised potentiometer's movement is driven by a DC motor, so an H-bridge chip was needed to control the direction of the current to spin the motor. A *L298* dual full-bridge driver (https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf) was acquired and a simple motor driver code was provided by mbed for both the motor and the H-bridge (Ford, 2010), with an additional motordriver library to control speed and dynamic braking (Hasler, 2010). Initial tests involved rotating the motor using input from a computer keyboard. The motor driver code was modified to take serial inputs from a computer terminal and translate them into stepped movements to turn the motor in either the clockwise or counter-clockwise direction (see Appendix K). Pressing down the 'u' button on the keyboard would rotate the motor in the clockwise direction, while pressing 'd' would rotate the motor counter-clockwise. The stop() function in the microcontroller code caused the motor to brake just after either button was released. The max rotation of the motor was roughly 300 degrees.

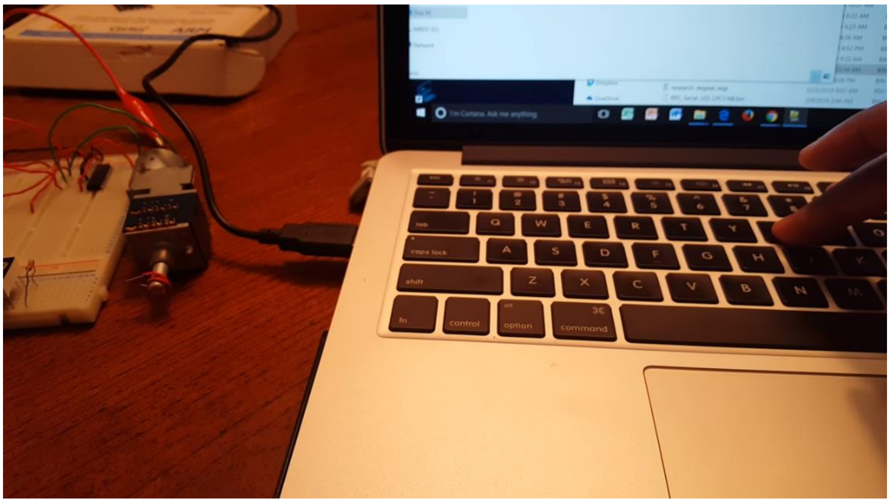


Figure 4.6 Rotating a DC motorised potentiometer using keyboard inputs from a computer (https://youtu.be/2CccFY-ziMI).

The following step involved tying Websockets into the functionality of the motor control. To implement this, the keyboard input values needed to be replaced by values delivered from the web interface. A simple JQuery knob was found online (Terrien, 2015) and adopted into the interface, allowing stepped rotations of a virtual turn dial in addition to providing the programmer freedom to specify the dial's angle, offset, and min and max values (see Appendix L). The current numerical value of the dial was sent from the web interface to the server using Websockets, and the mbed source code was adapted to receive the Websockets' dial values and rotate the motor clockwise if the numerical values increased or rotate the motor counter-clockwise if the values decreased (see Appendix M). Since the rotational speed of the motor was not proportional to the rotation of the virtual dial, a user would inevitably rotate the dial faster or slower than the actual rotational speed of the motor. To account for this, the virtual dial was allowed to rotate freely in 360 degrees, thus any rotation clockwise, no matter how fast or slow, would rotate the motor clockwise at a constant speed, and similarly any rotation counter-clockwise would rotate the motor counter-clockwise at a set speed. When the user stopped rotating the virtual dial the motor would stop, although minor drift could occur if the dial was rotated very quickly.

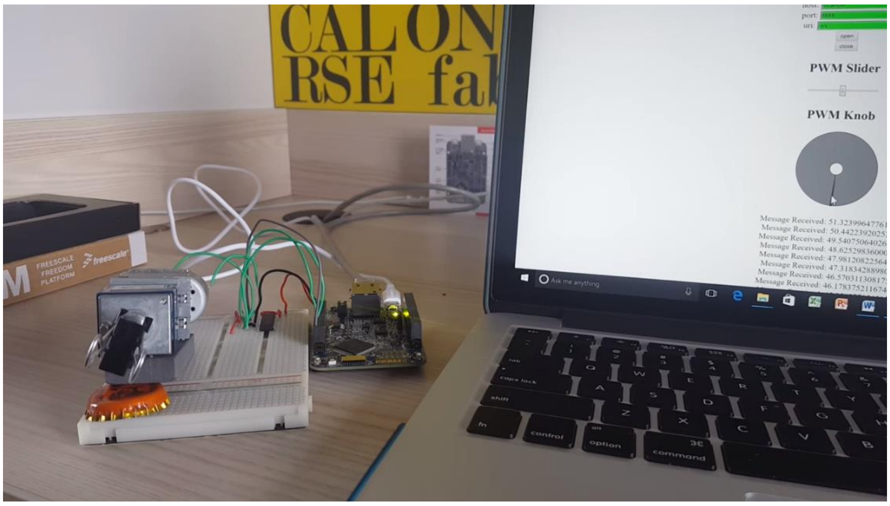


Figure 4.7 Using a virtual dial to rotate a motorised potentiometer (https://youtu.be/OADWqp5Pj3Y).

Note: These tests were effective using the interface dial primarily when working on a fixed resolution screen, such as on a PC or laptop. Use of a smart phone or tablet added difficultly making tactile contact with the virtual dial due to the image being smaller to accommodate the smaller screen sizes.

Most smart phones and tablets have zooming features, or the HTML "viewport" attribute could be set to scale the dial to a more suitable size, but changing the natural resolution of the image caused the dial to become less responsive than desired. As a result, the dial would either need to be made really large so that it is suitable on smaller devices, or it would have to be used at a very small size on a mobile device to look a standard size on a desktop.

After successful trials were conducted using the web interface to control an actuator over the Internet, the next step involved adapting this functionality into a musical device. Initially a *Moog Minitaur* analogue bass synthesiser was purchased as it contained several knobs and controls to easily manipulate sound.

Figure 4.8 Moog Minitaur analogue bass synthesiser.

A brief interaction with the Minitaur showed that the device's low-pass filter produced the most noticeable and immediate effect, so bands were used to secure a motorised potentiometer to the synthesiser. Minor slippage occurred at times as a result of the bands, however, the control system worked well enough to adjust the knobs on the physical synthesiser.

Figure 4.9 Web interface manipulating knobs on a synthesiser (https://youtu.be/2t8-VVcCPTo).

Next, the synthesiser was then connected to a Mac laptop which acted as a MIDI controller, and music was transmitted into the synthesiser's audio input. The web interface was then able to allow remote user interaction with the synthesiser to sweep the low-pass filter and manipulate the audio in real time. An example of this demonstration is referenced in the figure caption below:

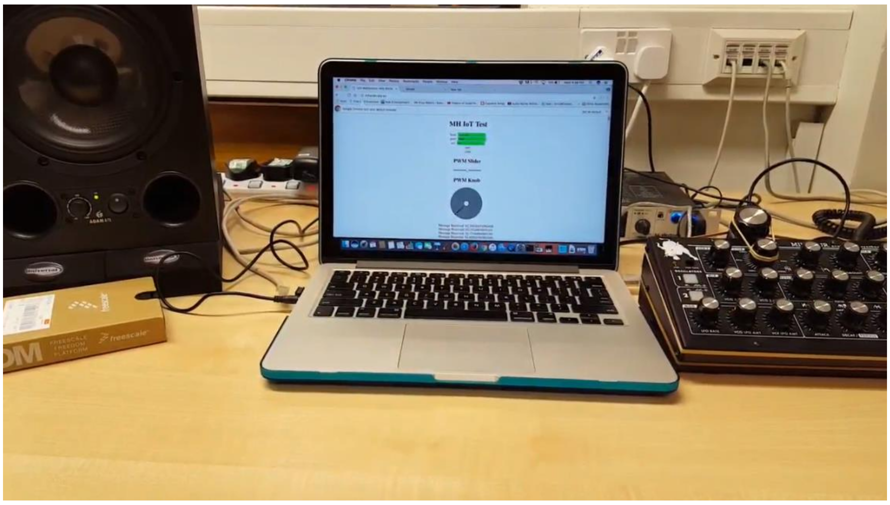


Figure 4.10 Synthesizer w/ IoT Motor Control Test (https://youtu.be/6sTJib8oAko).

Virtual interaction with a physical music processor proved that IoT communication protocols, such as Websockets, in conjunction with networked computing nodes provided by microcontrollers and online interfaces, have the opportunity to present unique tools to engage remote musical hardware. The previous developments can be scaled and modified to incorporate different audio devices, and addresses the capabilities of accessing and interacting with musical systems without the limitation of physical presence.

4.2 Exploring Audio Streaming Functionality

4.2.1 Initial JackTrip Audio Streaming Implementation

After successful implementation of a web-based control interface using Websockets, the next development stage involved exploring audio streaming applications that can deliver real-time music to the remote audio systems. Current investigations have been undertaken into developing low-latency, high-quality audio distribution over computing networks, and one platform in particular showed promise for use in an IoT music application: JackTrip. JackTrip is a Mac OS and Linux based system that enables real-time, online music performances between multiple computer clients over the Internet (Caceres, 2007a). JackTrip operates on top of the Jack Audio Connection Kit, which is a low-latency audio server that provides an application programming interface (API) for connecting, routing, and media distribution between sound and audio applications (Davis, 2001). JackTrip can support a maximum number of audio channels governed by the computers capacity, and manages bi-directional audio streams (Caceres, 2007a).

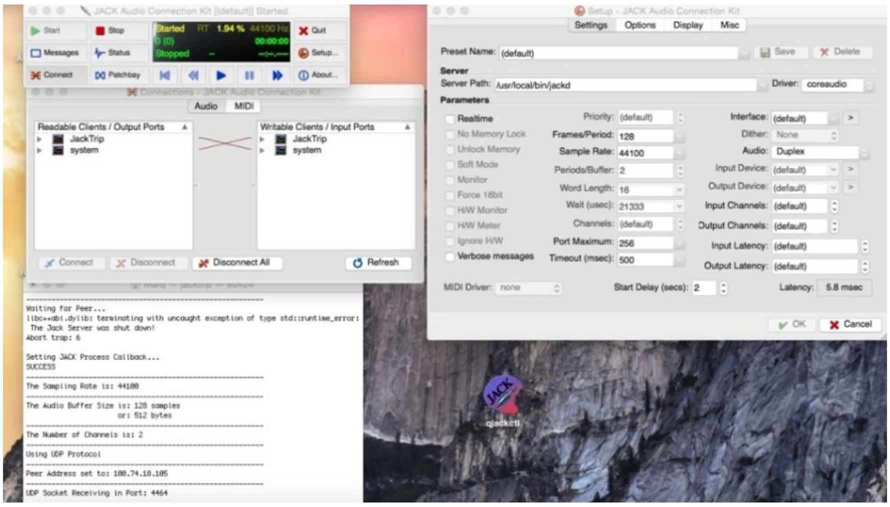


Figure 4.11 Jacktrip screen shot.

First explorations of JackTrip involved connecting two computers on a local area, Digital Subscriber Line (DSL) network using Wi-Fi, and sharing their internal microphone inputs to transmit audio. Although the latency between the streams was minimal, occasional artefacts and audio drops outs occurred over the DSL connection.

A second observatory test involved using JackTrip to stream audio between the United States and the United Kingdom. This first international test occurred between a host computer using a DSL-based home network in Los Angeles and a client computer on Anglia Ruskin University's enterprise network in Cambridge, UK that utilises the JANET high-speed education network. Using laptop microphones again as audio inputs to communicate between the two computers, the stream was less reliable, providing more latency as well as significant amounts of artefacts and audio drops. A traceroute taken of the connection between the two computers showed inconsistencies in the network signal paths between the two computers as well as high latency in the early stages of the connection (see Appendix N). After conducting an Internet speed test on the DSL network, one consideration made was that the low connection speed could adversely affect quality and not support demands for real-time audio streaming.

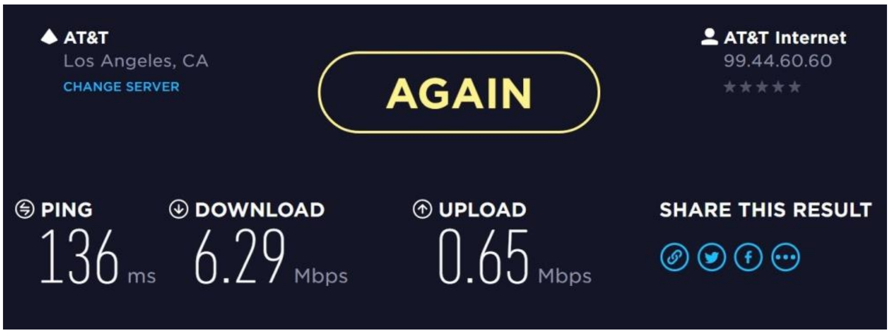


Figure 4.12 Ookla speed test of Los Angeles DSL network.

To compensate, the US base of testing was moved to the University of Southern California where high-speed network transfer could be facilitated by the Internet2 national research and educational network (NREN). The incorporation of an Internet2-based network in conjunction with JANET provided more reliable results. Traceroutes showed that the data packets from both sites followed more consistent signal paths, and early rates of transmission speeds were improved (see Appendix N). Some slight audio dropouts occurred at times during transmission, but a clear verbal conversation was able to be held between the two sites. These initial tests set the basis of a more thorough exploration of audio streaming detailed in Chapter 5.

4.3 Merging IoT Control and Audio Streaming

After initial testing of Websockets for virtual interactivity and JackTrip for real-time audio transfer, the next step was to combine the two applications into a first proof-of-concept demonstration of an IoT music system. To begin, an *Apple Mac Pro* was obtained and setup at Anglia Ruskin University to be used as a server computer to host both the Websockets and JackTrip connections. Additionally, an *A T.C. Electronic Studio Konnekt 48* audio interface was added to retrieve audio from the Mac Pro's sound card and deliver it to the Moog Minitaur, where audio processing would occur before being routed back into the Mac Pro for re-transmission to a client over JackTrip.

During this stage, further JackTrip research uncovered the limitation of Wi-Fi networks supporting real-time audio transfer over the Internet and the need of a hardwired Ethernet connection for stability (see Section 5.5: Wireless Network Testing). The Mac Pro by default requires an Ethernet connection for Internet connectivity; however, any connecting client laptops were also required to use a wired connection and the overall audio streaming quality drastically improved as a result.

The first full IoT music tests occurred with computers on Anglia Ruskin's local area network. The web interface was used to control a motorised potentiometer affixed to the Minitaur's low-pass filter knob, with JackTrip streaming a music file between the client and server computers. A *Skype* video call was also arranged between the two computers to show the manipulation of the synthesiser to the client. This test proved successful and audio was transmitted, processed, and returned to the client computer with negligible latency and no artefacts or audio packet drops.

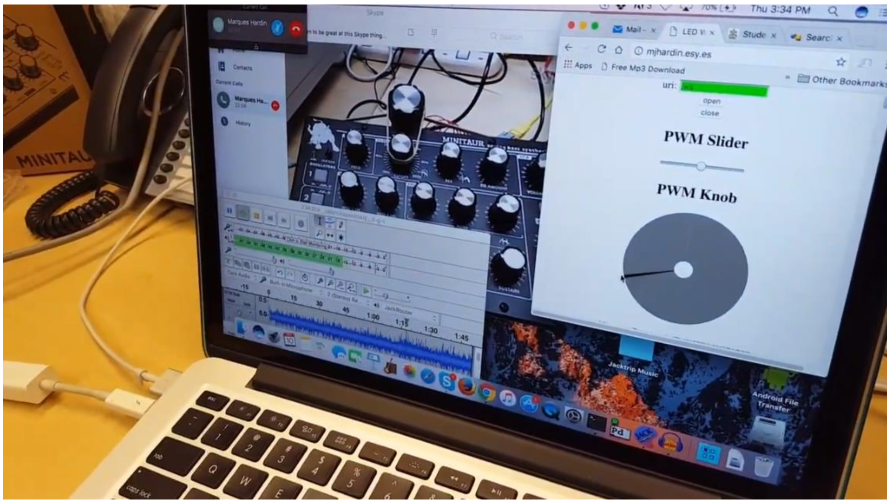


Figure 4.13 First stage full IoT music test incorporating virtual control and audio streaming (https://youtu.be/dwC713Hpdmc).

This ability to stream networked music, virtually control audio processing hardware to manipulate sound, and recapture the processed audio all from a centralised space showed promise of implementing a real-time IoT-enabled music system, and served as a functional concept of the final design.

4.4 Optimising the User Experience

4.4.1 Far-end Video Component

A useful application to optimise the user experience was the inclusion of a video element allowing users to monitor the control of any remote audio processing hardware. This ultimately would replace the need for organising a concurrent Skype connection between the server and a client computer.

YouTube Live-Video

An immediate focus was to utilise one of the popular social media live streaming platforms that support real-time video. *Facebook* and *YouTube* are two of the largest social media providers that allow their users to produce one-to-many live video feeds for remote audiences. During this stage of development, Facebook's infrastructure to support live videos was primarily based on the use of mobile devices. YouTube, however, showed potential to be a viable video option, especially with its offering of HTML code that can allow videos from their platform to be embedded into an external webpage. In order to live stream a video feed from a desktop computer or non-mobile device, additional encoding software was required to capture the camera feed and send it to YouTube. To achieve this, the open source software *Open Broadcaster Software (OBS) Studio* was used. The encoder software needed to be provided with a stream key from an active YouTube account, where afterwards the YouTube streaming server encodes the PCs audio and video feed for playback. Once the desired encoding properties for a video stream are set and the media is encoded, OBS provides a URL where the streamed content can be observed in addition to being watched from YouTube's livestream dashboard.

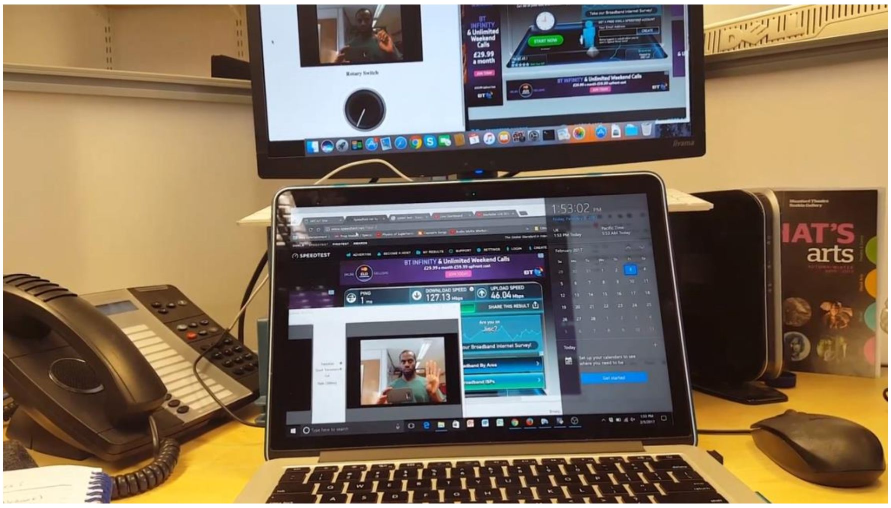


Figure 4.14 YouTube live media stream test (https://youtu.be/qt1UXeNCdtE).

The video quality was sufficient for hardware monitoring, however, the latency of the stream provided less than desirable results, as roughly 10 seconds of delay was observed and was deemed unsuitable for real-time control applications in this research.

WebRTC

Another option for media streaming was a fairly new concept called *WebRTC* (Web Real Time Communication), which enables real-time communication protocols for audio, video, and data transfer through web browsers and the Internet (WebRTC, 2011a). One of WebRTC's noteworthy qualities is that it enables Skype-like video conferencing with very low latency self-contained within a web browser. WebRTC utilises APIs that allow a user to capture media streams from input devices, built around the getUserMedia() command, and shares these streams with remote devices using RTCPeerConnection (Dutton, 2012). While WebRTC enables the functionality to access media inputs on a computer and share the media information with a peer, the actual connection between peer devices is facilitated by a process called *signalling*, which is not specified within the WebRTC standards so that developers have the flexibility of implement their desired mechanisms for connecting peers.

Signalling sets parameters for peer computing devices to find and identify each other behind secured networks and establish connections to exchange data. This process "involves network discovery and NAT [Network Access Translation] traversal, session creation and management, communication security, media-capability metadata and coordination, and error handling" (Castrounis, 2015). Due to security protocols, most computers connected to large enterprise networks (such as school or business networks) sit behind a firewall which regulates Internet traffic between the internal machine and external computers to help protect the internal network and computing devices from outside threats. To facilitate communication with the external network, a network access translation (NAT) device sets parameters allowing internal devices to be identified securely on the external network. Any device connected within a network is provided a private I.P. address for identification purposes. NAT devices, however, help establish a separate public I.P. address for computing devices that is displayed outside of network, allowing these devices to be identified to the outside world beyond the firewall (Castrounis, 2015). External devices can then use the public I.P. address to establish communication and send or request information to internal machines, where these requests are again managed by the firewall and NAT device and, if allowed, delivered to the internal machine using the private I.P. address.

For WebRTC communication to occur, devices rely on STUN (Session Traversal Utilities for NAT) and TURN (Traversal Using Relays around NAT) servers that request identifying information about the machine or device and subsequently present this information to external devices. After identification, the signalling process occurs, creating the method to "negotiate and establish the network session connection with [the] peer" (Castrounis, 2015). As WebRTC does not provide APIs or support to handle network traversal and signalling, the *PubNub* API was used as it unified all the components of WebRTC and signalling into one package. Pubnub offered tutorials on establishing a one-to-many WebRTC video stream with options for embedding the video into a live webpage (Gleason, 2015). The code was modified to develop a video feed for this research and is found in Appendix O and P.

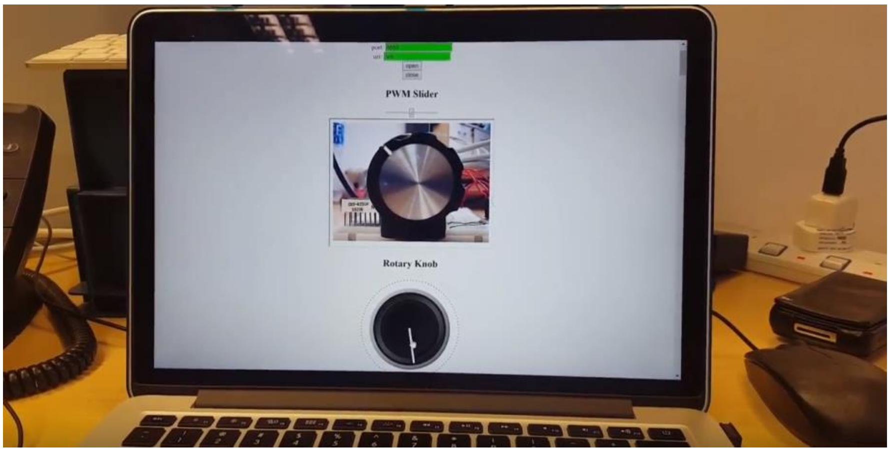


Figure 4.15 Pubnub-enabled WebRTC live stream test (https://youtu.be/FB9caB-OtAk).

WebRTC proved to be a suitable option for real-time video feedback, showing negligible latency speeds of less than a second when controlling a remote audio processor. Additionally, since the code was written using HTML and JavaScript, the video feed was able to be embedded into the web interface alongside the Websockets controls. A separate WebRTC video broadcasting page was used capture the video of a desired audio processor and that was broadcast to the web interface for viewing by the client.

4.4.2 Follow-up Evaluations to Real-time Audio Streaming

WebRTC offered an effective way to share media data over a webpage and had potential to consolidate control, video, as well as audio streaming into a unified interface, a desirable trait not currently applicable with JackTrip as it is a self-contained, external application. Observing this, new focuses concentrated on evaluating and comparing JackTrip and WebRTC to determine which would be a better solution for network-based music transfer.

A full assessment of JackTrip and WebRTC used for real-time audio streaming is detailed in Chapter 5. However, a comparison of both systems concluded that WebRTC provided processes ideal for voice data and video conferencing applications, whereas JackTrip was optimal in supporting a wider spectrum of sound frequencies better suited for musical instruments and audio files. Although WebRTC's native development in HTML5 and JavaScript was convenient for developing a centralised audio, video, and control user experience, JackTrip provided greater reliability in audio streaming and better accuracy preserving musical sounds across the Internet.

4.4.3 Updating the Control Interface

A reoccurring issue within the design and build stage was the responsiveness of the control system; data was effectively transmitted from the web interface to remote audio devices, but the response of the virtual control system lacked precision. An original focus of the web interface was to model the controls after recognisable elements on a standard audio device, such as a rotary knob or dial. HTML5 accurately handles input data from many virtual objects, such as graphical buttons or sliders, but there were no simple solutions for developing a knob and this task ultimately required using external resources and amendable, open-source JavaScript code.

With regard to solutions, Baskar (2017) states, "typically, an IoT solution needs to handle multiple data types from multiple devices on a user interface (UI) that flows seamlessly across interfaces… With such diversity at many levels, UX design becomes incredibly complex for IoT solutions." One of the main issues with the web interface was that the virtual dial implemented in the earlier design produced linear data to manipulate logarithmic tapered potentiometers with undetermined log ratios. Additionally, the dial needed to be controlled by screens of different resolution sizes, resulting in issues with the tactile experience. While the dial could often be rotated as desired, the response did not feel natural (ex: when the knob was rotated to 50% of its max output, 50% of the processing affect should occur, however, this was not the case with the logarithmic potentiometer). In addition, changes to the screen resolution, including zooming in or out of the web interface page, prevented the dial from responding accurately to user input.

Baskar (2017) argues that choosing complex approaches to simple IoT solutions is often a problem, stating "once these multiple data types from multiple devices are together, the end user needs to access a simple yet informative visualization on any interface they want". The initial tests controlling a motorised potentiometer using serial keyboard inputs worked well at rotating a motorised potentiometer, so the rotary dial was scaled back to virtual push buttons that determined the direction of rotation for the motors. The push buttons provided better feedback than the rotary dial and more accurate movement of the motor.

4.5 Consolidated IoT Music Application

4.5.1 Networked Mixer and Remote EQ Processing

The final phase of the design and build stage was incorporating networked audio processors to complete the IoT music processing chain. The first processor involved creating a hybrid analogue-digital audio system by augmenting a stand-alone analogue music device with digital control capabilities. This required embedding actuators and networking resources into the existing hardware of the device, allowing it to be physically controlled and manipulated from a remote location through the Internet. A decision was made to adapt motorised potentiometers into the equalisation (EQ) processing of a mixing board so that remote users could adjust the Hi and Low-cut frequencies of the transmitted sound. A *Peavey PV8* 8-channel mixer was acquired for this purpose.

Figure 4.16 Peavey PV8 8 channel mixing board.

A majority of the knobs on the mixing board are composed of linear potentiometers typically ranging from 10-20 kΩs. To adapt the motorised potentiometers to the mixing board, the three legs of the motorised potentiometer were soldered in parallel, creating a voltage divider, to the three legs of both the Hi-cut and Low-cut frequency knobs on the first channel of the board. Testing showed that even though the motorised potentiometers had a logarithmic taper, placing them in parallel with the linear potentiometers was able to produce a linear-like taper output.

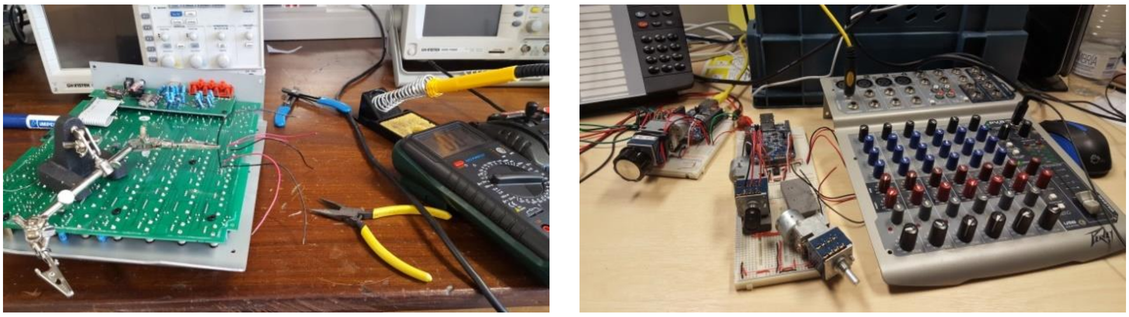


Figure 4.17 Soldering motorised potentiometers to the mixing board EQ knobs.

To achieve a balanced effect, the EQ knobs of the mixing board needed to be set to their unity position (direct centre so that there is no attenuation or boost of their respective frequencies), allowing the EQ effects to be solely generated by the changing position of the motorised potentiometers. New sets of streaming audio tests were conducted from a client to the host computer, where the host used an audio interface to route audio into the mixing board and back to the computer. Once the EQ effect was added to the audio, the processed audio was transmitted back to the client computer through JackTrip for monitoring and recording.

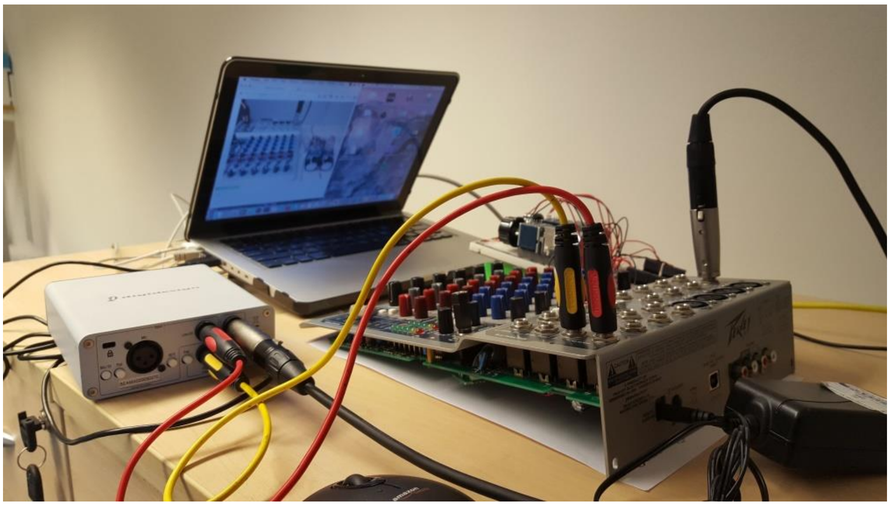


Figure 4.18 IoT EQ processor with host computer and audio interface.

The tests provided positive results, and finalised one-half of the complete, proof-of-concept IoT-enabled music system. A link to a reference video is provided in the figure caption of Figure 4.19

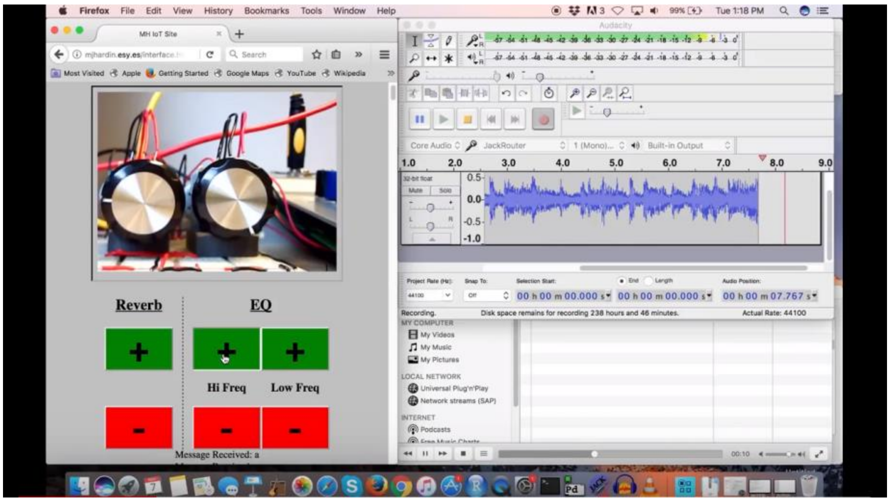


Figure 4.19 IoT-based EQ processor (https://youtu.be/3wwX_5OTVaE).

4.5.2 Distributed Natural Reverb Processor

The second music processor sought to exploit the unique sonic attributes of physical spaces and adapt them into a natural, real-time reverb application. The concept involves sending audio from a client computer to a host in a remote acoustic environment, broadcasting the audio into the space using loud speakers connected to the host computer, and finally using a microphone to capture the broadcast audio as it echoes through the space, where it is retransmitted as reverberant sound back to the client. The audio captured in the microphone needed to be added to the source signal within an audio interface, and the product would then be a mixed sum of the transmitted source audio plus the captured reverb. To prevent audio feedback, the sound captured by the microphone was not routed to the loudspeaker through the audio interface. Finally, the client computer would have the ability to control the amount of signal collected by the microphone through the web interface and determine how much reverb is applied to the overall mix.

To manage how the reverberant sound is acquired, the T.C. Electronic Studio Konnekt 48 audio interface was connected to the host computer and routed the computer's system audio from one of the interface's outputs to a loudspeaker. The female end of a balanced XLR cable was connected to the left channel output of the interface and the positive (+) signal pin of the male end was connected to the first leg of a motorised potentiometer.

The positive signal pin of another XLR was then connected to the wiper of the potentiometer and the opposite end of the cable was plugged into the input of the loudspeaker. The third leg of the potentiometer was simultaneously connected to the ground pins of both XLR cables while the inverted (-) signal of the XLRs were left unconnected so only an unbalanced signal was transmitted. The dc motor of the rotating motorised potentiometer was controlled through the web interface by a microcontroller using a wired Ethernet Internet connection to received control information. When the transmitted audio signal passed through the host computer to the loudspeaker, the client could determine how much signal reached the loudspeaker by manipulating the potentiometer's wiper from the web interface.

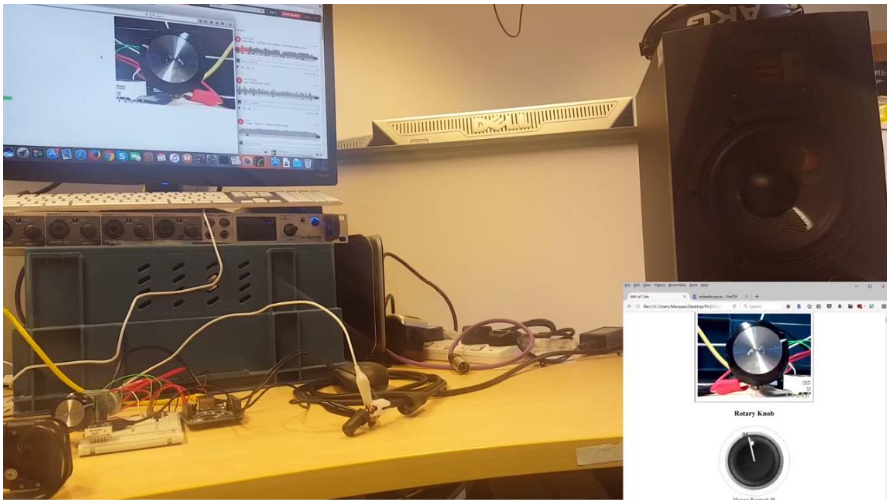


Figure 4.20 Motorised potentiometer controlling the amount of signal sent to host computer (https://youtu.be/iPayeOPipec).

One consideration that needed to be made was the placement of the motorised potentiometer in the chain between the microphone and the audio interface. A dynamic *SM58* microphone was acquired for the natural reverb application, but these microphones require a balanced signal input, causing the initial unbalanced configuration not to work. Fortunately, Alps RK27 potentiometers have a dual-gang design, where a single unit contains two internal, independent potentiometers controlled equally by the same rotating shaft for stereo audio applications. This meant that the signal and the inverted signal of an XLR could be sent to the first leg of each potentiometer leg on the RK27, and the wiper outputs could be routed to the respective signal and inverted signal pins of the XLR connecting to the input of the audio interface. The third legs of the potentiometers would then form a common ground with each other, the XLR coming from the microphone, and the XLR going to the interface.

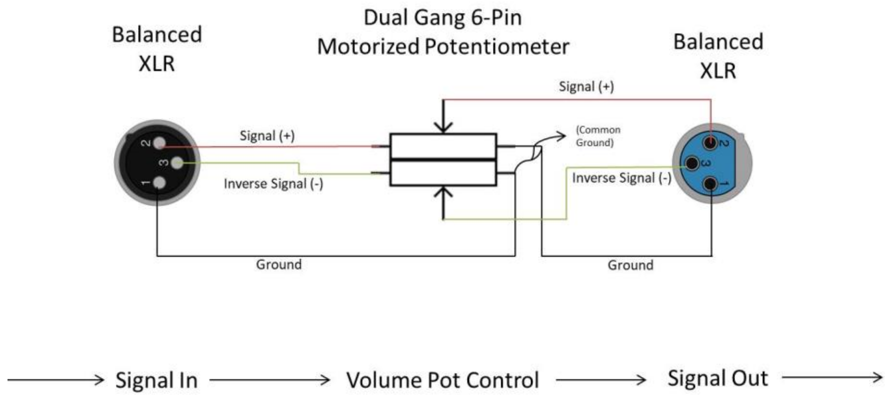


Figure 4.21 Balanced XLR signal routing through the Alps RK27.

To test, the motorised potentiometer was placed in between the microphone and the mic input of the audio interface. This caused problems because mic level signals are very low, roughly a thousandth of a line level audio signal. Since the microphone input of the audio interface contains a preamp to boost mic level signals to line level, this also amplified any noise associated with potentiometer hardware and the movement of the motor. To resolve this, the microphone was connected directly to the mic input of the audio interface, where the audio was routed to an XLR output of the interface. An XLR was then run from the output of interface to the potentiometer in order to control the now line level signal, and afterwards the audio from the potentiometer was routed back into a second input of the interface. At the interface the amount of reverberant audio added from the potentiometer was mixed with the original audio before being sent back to the client computer.

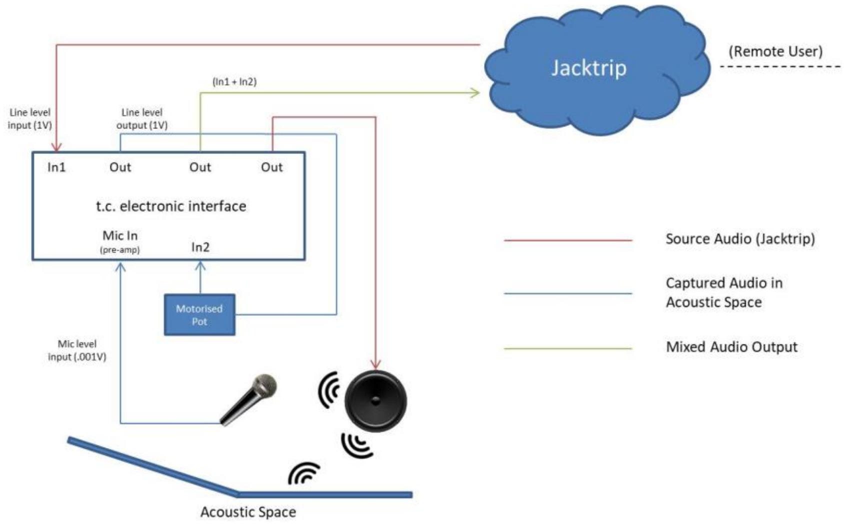


Figure 4.22 Signal flow for distributed natural reverb processor.

In a first demonstration, a small reverb chamber was made placing a speaker and a microphone into a metal filling cabinet.

Figure 4.23 File cabinet used as a natural reverb chamber.

The filing cabinet successfully produced a hollow-sounding echo chamber where the amount of reverb captured could be adjusted from the web interface. A view of the interface as well as a demonstration of the application is show in the image and image caption below.

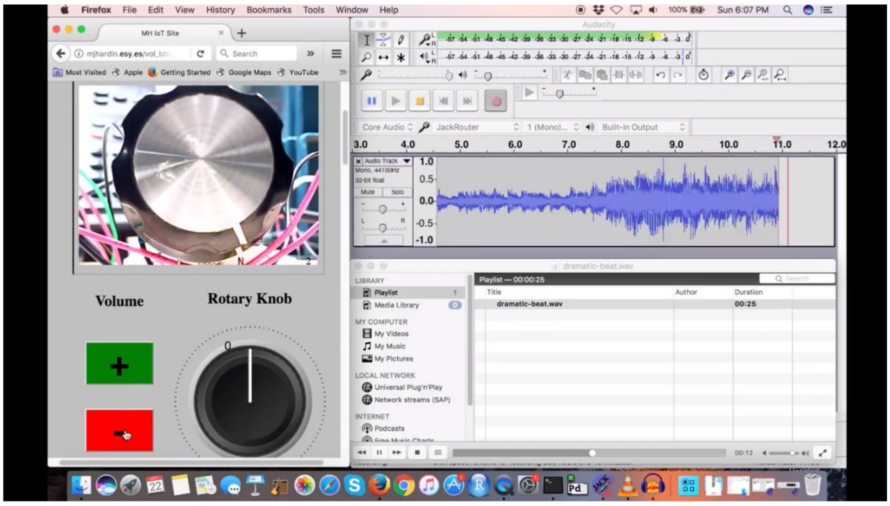


Figure 4.24 File cabinet reverb demonstration (https://youtu.be/WVmnwvPhenM).

In a second demonstration, a venue with pronounced echoes was sought to be used as a natural reverb chamber. The Ruskin building at Anglia Ruskin University is one of the oldest buildings on the Cambridge campus and houses the Cambridge School of Art, which contains many facilities where art and media students present academic works and artistic projects. The basement of the Ruskin building contains several narrow corridors and barren rooms with hard surfaces that are ideal for art exhibitions, and also have the added benefit of containing interesting acoustics with engulfing echoes.

Figure 4.25 Art exhibition space in the Cambridge School of Art, Anglia Ruskin University.

One exhibition space was equipped with Ethernet ports that were configured to support a host computer, and a microphone and speaker were placed in the room to produce the IoT musical reverb effect. The SM58 was also replaced with an *Audio-technica AT2020* condenser microphone as it had better sound pickup. The results were more subtle than the file cabinet demonstration, but offered a more true-to-life reverberant experience.

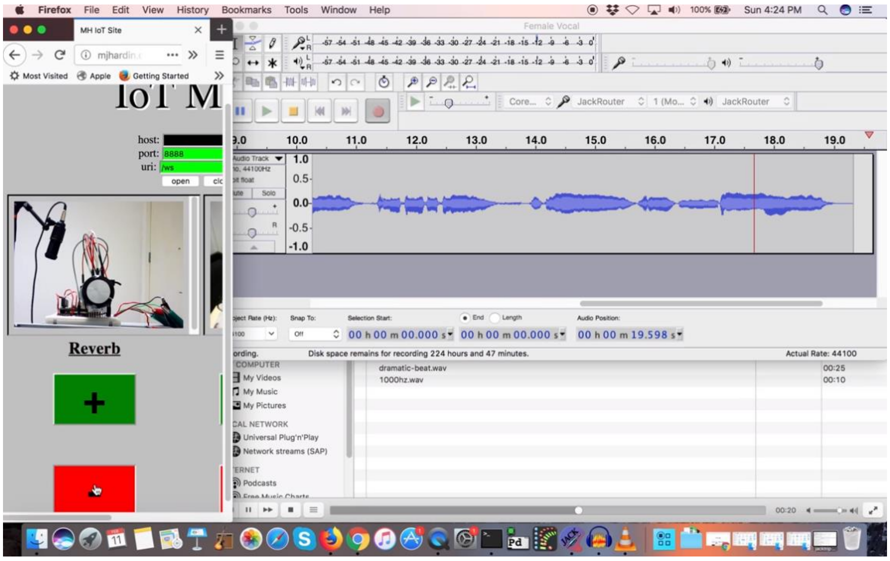


Figure 4.26 Cambridge School of Art real-time reverb demonstration (https://youtu.be/yc0hJcxY9Vg).

4.5.3 Combined IoT Music Processing Application

After completing initial prototypes and testing of both the IoT-based EQ and reverb processors, the final task was combining the two into a consolidated, IoT-enabled music application. The full application consists of a client computer transmitting audio to one processor, and after some processing the audio would then be delivered to the second processor, where it is processed again and completes its round trip journey back to the client. In total this required 5 wired network connections (3 for the client computer and host processors, and 2 for the microcontrollers controlling the two processors), and two JackTrip sessions hosted on the server computer. The signal flow of the system is shown in Figure 4.27.

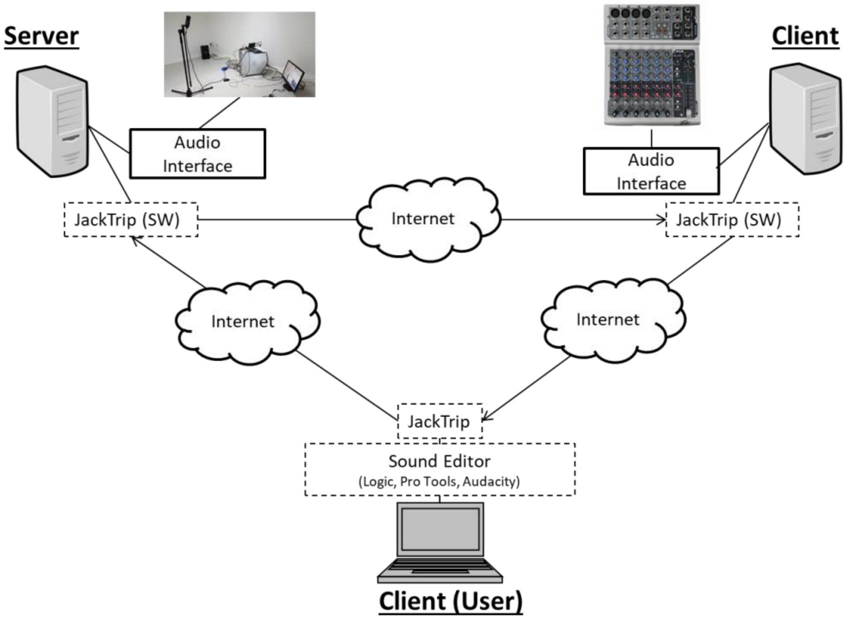


Figure 4.27 Audio signal flow of full IoT music application.

In order to allow remote computers to utilise the IoT music application, the server computer had to be moved to the virtual DMZ ('de-militarised zone' that sits between the internal and external firewall) of the university network and given a static I.P. address. This allowed the server to host internal and external JackTrip and Websockets connections, bridging outside audio traffic to the internal music processors as well as control data to manipulate of the microcontrollers and actuators.

To demonstrate the full IoT music scenario, the Mac Pro server computer (referred to as Host 1) was connected to the T.C. Electronic audio interface and was responsible for reverb-related audio processing. A second internal client computer (Host 2) was connected to the Peavey mixer via another audio interface and was responsible for processing and transmitting the EQ audio. Host 1 established a publically-accessible Websockets server that allows client computers from either inside or outside Anglia Ruskin's network to connect and send control information to the networked microcontrollers controlling the motorised potentiometers attached to each processor. Additionally, as a single instance of JackTrip can only connect two computers over a one opened network port, two JackTrip instances were run on Host 1, allowing Host 2 and a connecting client computer to stream audio between the server over two unique and dedicated open ports. With the two JackTrip instances setup on Host 1, the desired audio path can be arranged to route music between the user-based client computer and the two processors.

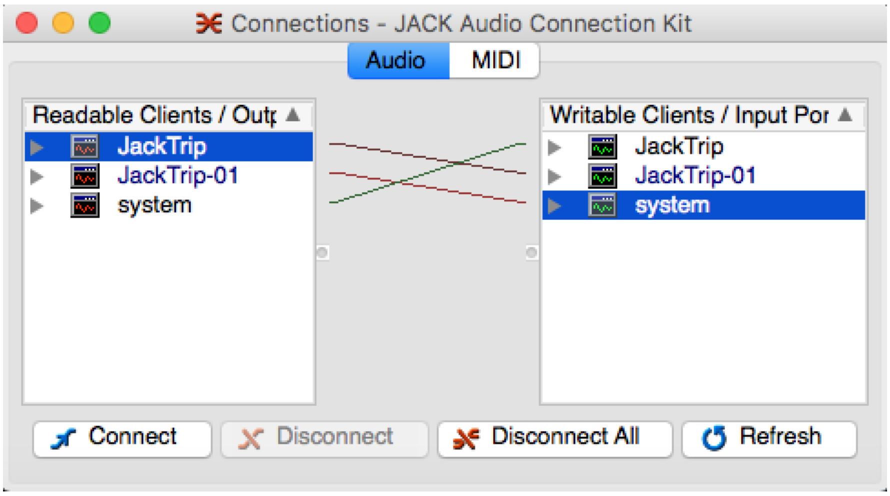


Figure 4.28 JackTrip routed connections.

With this arrangement, music was successfully transmitted in real-time to the two remote processors and returned to the client computer for monitoring and capturing. Figure 4.29 shows the full user experience of a system with a demonstration provided in the image caption.

Figure 4.29 Complete, IoT music application with natural reverb (https://youtu.be/ulKuf920Y20).

The completed, proof-of-concept IoT music system demonstrates a possible scenario where the Internet of Things can increase engagement with physical and analogue musical system and augment music production. The overall system was used to evaluate the opportunities of virtual production processes for musicians and successfully demonstrated to producers from distances up to 50 miles away (max: London to Cambridge). To better facilitate the evaluation phase of the research, a completely hardware processing system was implemented, replacing the natural acoustic space with an *Alesis Nanoverb2*.

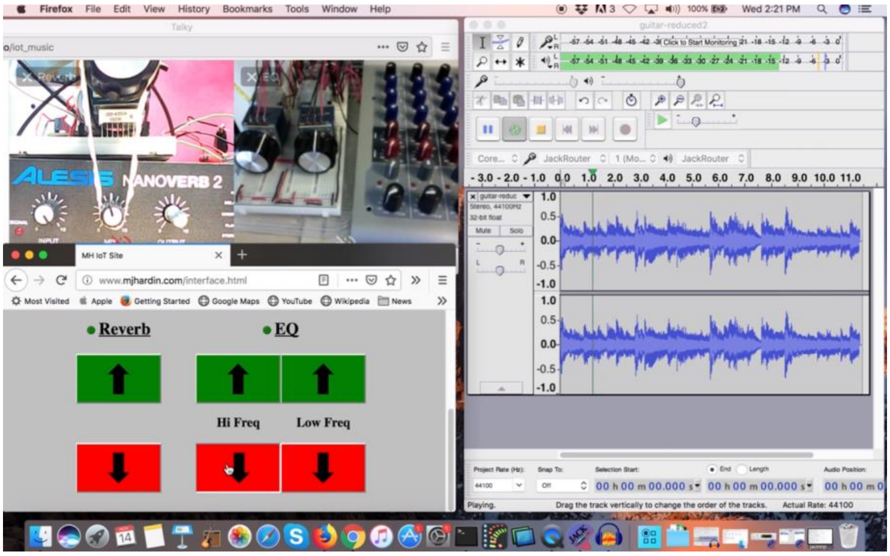


Figure 4.30 Complete, hardware-only IoT music system for user insight evaluations (https://youtu.be/U3X4ekrlmQU).

The Nanoverb2 aided in the user insight evaluations by allowing live demonstrations of the full IoT music system to be more flexible, as they could be set up at any time without the need for an available natural space. The results of the music producer evaluations are presented and discussed in Chapter 6 of the thesis.

5. Performance Analysis of Audio Streaming Platforms

This chapter details evaluative methods seeking to verify the performance of audio streaming platforms and their capabilities to support lossless, networked audio transfer for IoT-enabled professional audio systems. The chapter first gives a comparison of two platforms (JackTrip and WebRTC), providing an analysis of audio dropouts, distortion artefacts, and latency measurements produced by the platforms with accompanying spectral analysis and listener perception tests. The second part of the chapter tests the higher performing platform to a finer level, observing its functionality over various computing networks. This chapter supplements information found in Hardin and Toulson (2019).

5.1 Audio Streaming Test and Analysis Methods

5.1.1 Test Procedures and Source Audio

Three specific audio streaming experiments are conducted. These are:

1. Comparing the performance of JackTrip and WebRTC streaming platforms.
2. Investigating the performance of lossless streaming on local area networks with wired and wireless connections.
3. Evaluating streaming performance under differing wide area network conditions.

The primary aim of these streaming tests is to observe and compare any discrepancies between the source and the transmitted audio files, as well as documenting errors that arise as a result of the streaming process. Typical streaming errors include clicks, pops, buzzing sounds, or gaps of silence in the output audio file.

Three pulse-code-modulation (i.e. uncompressed/lossless) Microsoft Wave audio (.wav) files are utilised for the streaming trials (Fleischman, 1998). These are:

a. 10 second 1 kHz sine wave
b. 30 second frequency sine sweep from 0-22.5 kHz
c. 10 seconds of recorded music (acoustic guitar medley)

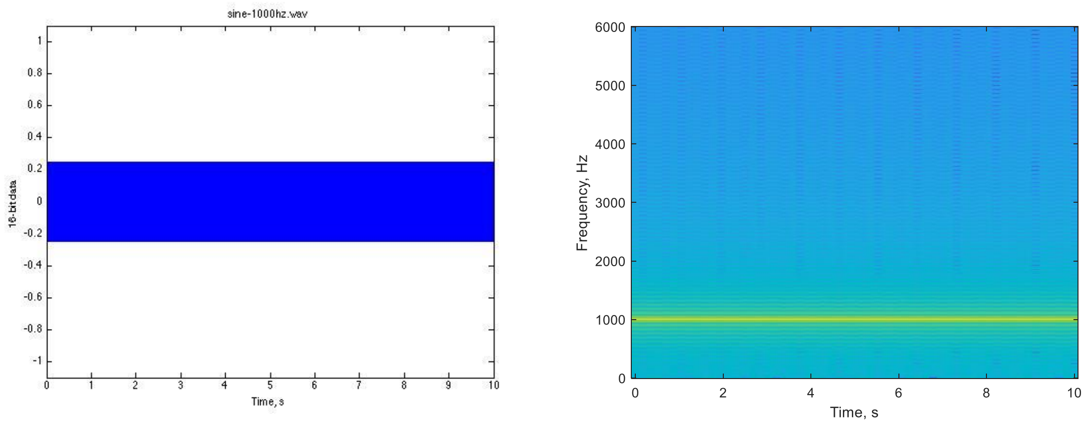


Figure 5.1 Sine wave source audio waveform and spectrogram.

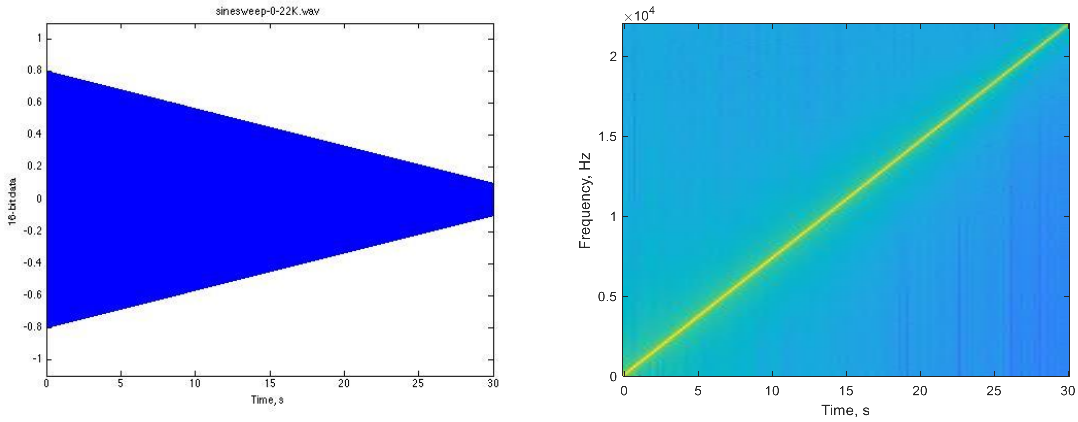


Figure 5.2 Sine sweep source audio waveform and spectrogram.

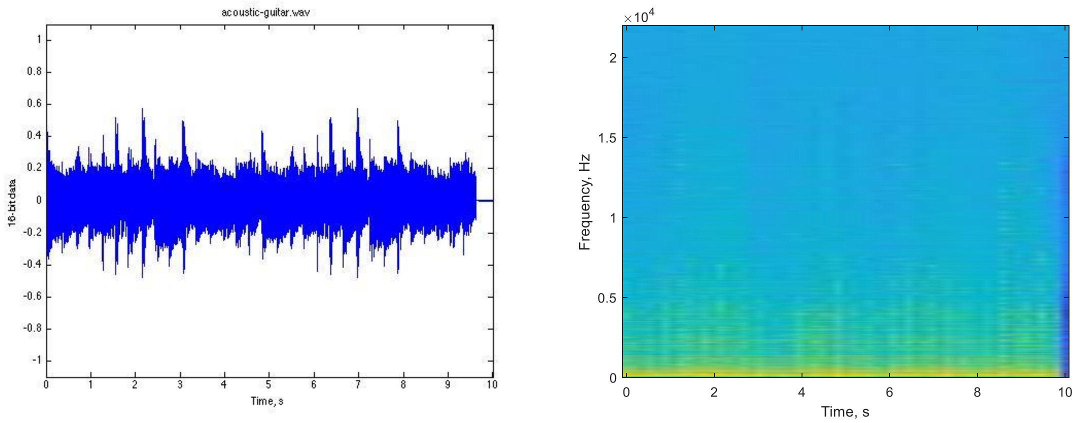


Figure 5.3 Acoustic guitar source audio waveform and spectrogram.

The 10 second 1 kHz sine wave provides a consistent stream of audio at a single frequency, allowing easy observations of data drop outs or distortion to the signal that may occur as a result of streaming. Since the 1 kHz sine wave was imperative for the computational analysis, a pure tone was generated using MATLAB (see Appendix Q). The 30 second 0-22.5 kHz frequency sine sweep determines if the streaming platforms accurately preserve or alter any specific range of frequencies within the audible human hearing range. The recorded music file provides a complex, dynamic audio sample to help determine if the audio streaming platforms manipulate any perceivable characteristics of 'real-world' music. The sine sweep and dynamic audio samples are solely used to determine qualitative differences between the two streaming platforms while the sine wave is used for more rigorous quantitative evaluations. All three test audio files are single channel (mono) and presented at a 44.1 kHz sampling rate.

Each audio trial is conducted 5 times allowing them to be evaluated for performance consistency and repeatability. Audio is streamed from one networked computer to a secondary computer, where the transmitted audio is then recorded as a new Wave audio file, matching the settings of the source file. Audio is analysed using MATLAB scripts to visualise waveform and spectrogram data (Appendix R) as well as measure specific performance characteristics of the transmitted audio in comparison to the source. The focuses of the streaming analysis are described in detail in each case below.

5.1.2 Measuring Dropouts

In the context of this research, audio dropouts account for any sudden loss or fluctuation in the audio data that causes instantaneous step changes in the transmitted signal, altering its characteristics from the source sound. Dropouts can produce undesired glitches including clicks, pops, and intermittent loss of sound in the audio playback resultant from interruptions to the data packet stream (Robjohns, 2008). Dropouts become more pronounced in real-time applications because the low-latency requirements "inhibit retransmission of lost packets," and issues such as network link failures, routers discarding packets, packets being received out of order or delayed in delivery (jitter), and packets being disregarding by the receiver after being received too late for playback all contribute to these interruptions (Voldhaug, Hellerud, and Svensson, 2006). Some examples of audio dropouts in a 1000 Hz sinewave are shown in Figure 5.4.

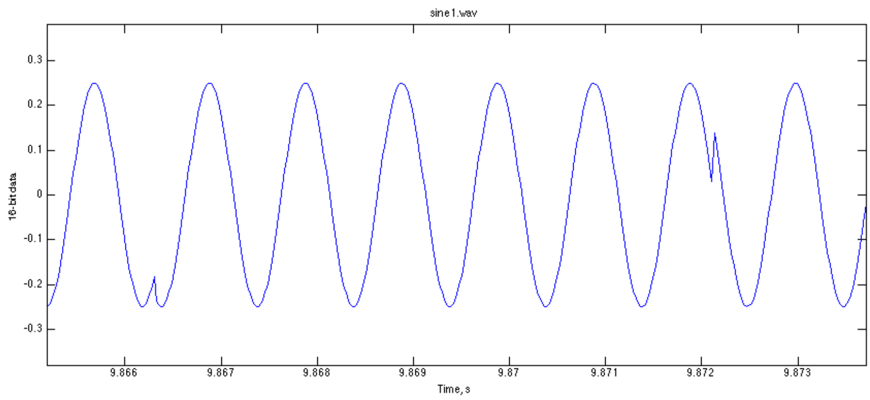


Figure 5.4 Example audio dropouts identified at 9.8663 seconds and 9.8721 seconds.

It is possible to count audio dropouts when using a test sinewave, by evaluating the sample-to-sample difference in the received audio data. The greatest possible inter-sample difference for a 1 kHz sinewave, normalised to unity amplitude and sampled at 44.1 kHz is approximately 0.15, which is observed at the sine wave's maximum gradient at the point of zero crossing, as shown diagrammatically in Figure 5.5.

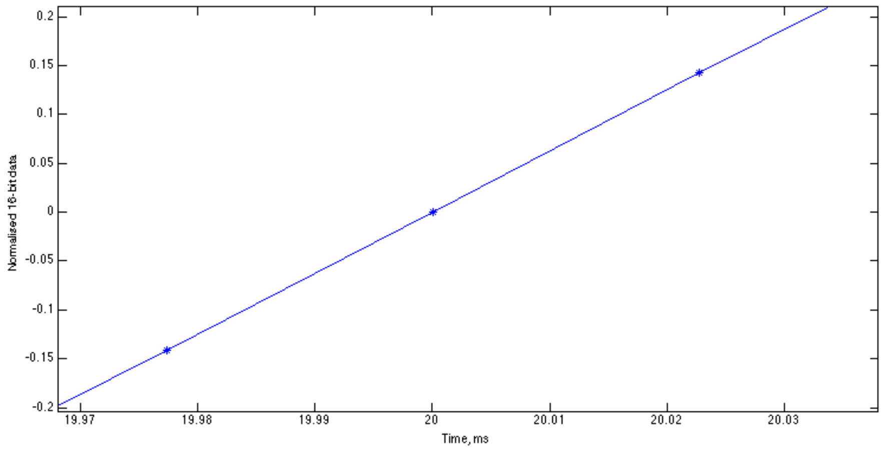


Figure 5.5 A 1 kHz sine wave's maximum gradient and inter-sample difference (=0.1425) when sampled at 44.1 kHz.

The exact value for the maximum sample-to-sample difference is calculated as follows:

The gradient of a sine wave is calculated by

$$\frac{d}{dt}\sin\omega t = \omega\cos\omega t$$

where ω is the angular frequency (rad/s) and t is time (s). The maximum gradient is hence when $\cos \omega t = 1$, so the maximum gradient of a sine wave is simply $\omega = 2\pi f$, where f is frequency (Hz). The sample period, P, for a signal sampled at 44.1 kHz is 1/44100 seconds, so the maximum amplitude increment per sample of a 1 kHz sinewave, is calculated as

$$2\pi fP = 2\pi * 1000 * \frac{1}{44100} = 0.1425$$

It is therefore possible to count dropouts by identifying any sample value step changes in the received sine wave that exceeds 0.1425 multiplied by the amplitude of the sinewave. It is of course possible for a dropout to leave samples perfectly aligned as a matter of coincidence, and in such rare cases may be missed by the proposed dropout counting method. While more elaborate algorithms for identifying dropouts might be possible, the method proposed here is sufficiently accurate for evaluating the relative performance of network audio platforms. The MATLAB script used for counting dropouts in an audio file is included in Appendix S.

5.1.3 Measuring Distortion Artefacts

As discussed by Moore et al. (2004) and Toulson et al. (2014) *nonlinear distortion* refers to the introduction of harmonic and inharmonic frequency components that were not present in the original signal. The amount of unwanted harmonic distortion can be calculated as *total harmonic distortion* (THD), where harmonic frequencies are measured at integer multiples of the fundamental test frequency. THD is usually calculated as a percentage based on the ratio of the power sum (root-mean-square) of all the harmonic components to the power sum of all the harmonics plus the fundamental (Temme, 1992).

When evaluating a single sinusoid test signal, spectral powers which are not identified as fundamental or harmonic are classified as noise. The noise can also be quantified as a percentage of the fundamental frequency power (N), so allowing the value of THD+N to be calculated. THD+N is a much simpler quantity to measure collectively (rather than separately for THD and separately for N) for a single sinusoid test, since it essentially refers to the power of spectral components that are evident in the processed signal when the raw test signal component is removed, as discussed by Prism Sound (2018), who are leading manufacturers of audio test and measurement equipment. In line with the published recommendations by Prism Sound, THD+N is measured in this research by applying brick wall filters in the frequency domain after the signal spectra has been calculated. The filtering includes a notch filter around the 1000 Hz test frequency, with a low-cut filter implemented at 22 Hz, and a high-cut filter implemented at 22 kHz. In order to confidently remove any side bands in the signal spectrum, the notch filter is set relatively wide to cut all frequencies between 900 Hz and 1100 Hz. The filtering profile applied for calculating THD+N is shown in Figure 5.6, which displays the frequency spectra of a distorted 1 kHz sine wave as an example.

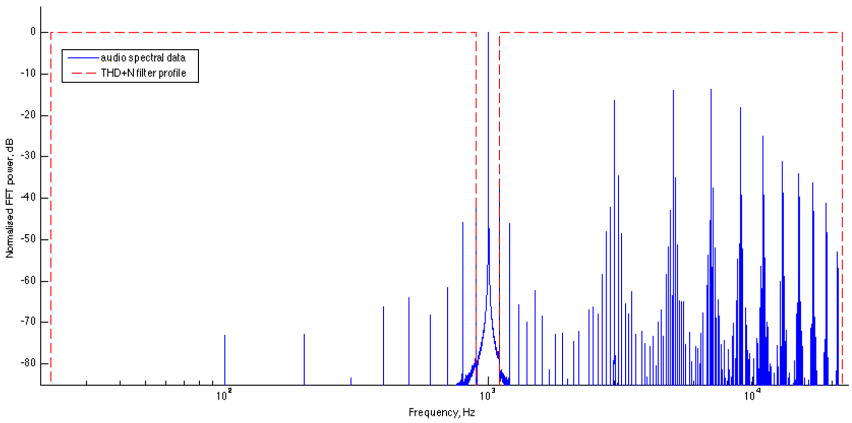


Figure 5.6 Example distorted 1 kHz sinewave spectrum with THD+N filter profile.

Processed audio can exhibit additional distortion and noise depending on the tools and mechanism used for transmission or recording, and THD+N measurements may also vary depending on the resolution of the notch filter used to conduct the FFT.

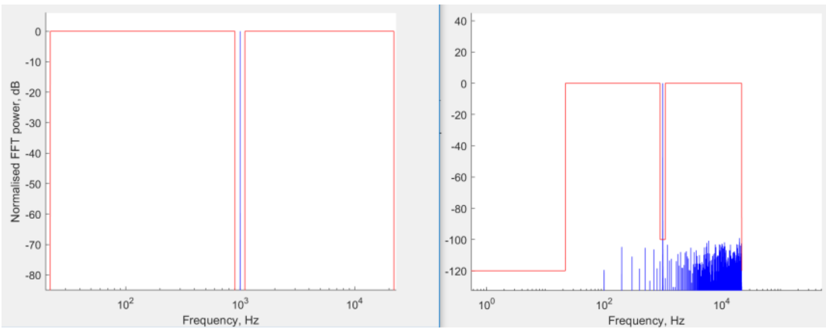


Figure 5.7 THD+N filter used on the 1 kHz sine wave to provide a value of .0048.

While there are no widely agreed values for acceptable THD+N ranges, it is desirable to obtain the smallest ratio possible, and in the case of quality measurements for amplifiers, Prosuk (2017) recommends a distortion value of less than 1%, or .01. The THD+N value calculated for the source 1 kHz sine wave in section 5.1.1 is **0.0048**, and for the purpose of this research, relative comparisons between test results are of most value. The MATLAB script used for measuring THD+N in an audio file is included in Appendix T.

Note: THD+N values captured in this research have been modified since Hardin and Toulson (2019), as the audio editor used for the initial recordings was shown to add minor noise when exporting signed, 16 bit PCM .wav files. Information regarding this can be found in Appendix CC.

5.1.4 Measuring Latency

In audio engineering, latency is defined as “the time delay experienced between a sound or control signal being generated and it being auditioned or taking effect,” and is often measured in seconds or milliseconds (Robjohns and White, 2019). While latency measurements up to 150 ms is deemed acceptable in traditional telephony cases, the average person begins to perceive an individual sound as two distinct sounds after 30 ms of latency (Rouse, 2016b) and some musicians can perceive the effects of latency at much lower thresholds, sometimes lower than 25 ms dependent on the style of music (Bouillot and Cooperstock, 2009). Particularly for live-performance and real-time audio scenarios audio transfer relies on small buffers with no compression (Bouillot and Cooperstock, 2009), and due to these strict parameters, a “sudden, unexpected, increase in latency can cause a drop out in the signal at the destination” (Bouillot et. al, 2009, p. 732).

Source-to-destination latency measurements are useful for networked musical performances and online jamming sessions, but an IoT music application where audio needs to transmitted to a remote node and returned to a central location benefits from the observation of roundtrip latency times. Measuring the roundtrip time of audio transmission over the network can become complicated when incorporating heterogeneous A-D (analogue to digital) and D-A (digital to analogue) processors that account for additional delays in their hardware or software. Bouillot and Cooperstock (2009) propose a manual mechanism for measuring latency using a multi-channel audio editor to compare the time difference between the playback of the source file and a captured recording of the audio as it delivered to a remote node over the network and returned back to the source. Building upon this concept, latency measurements in this research are obtained by configuring the audio interface of the server computer to loopback any audio streams received from the client. As the client computer transmits audio it simultaneously records the audio returned from the loopback server and the timing delay between the two streams determines the round-trip latency. Halving the round-trip delay time determines latency from source to destination.

Figure 5.8 Observing delay between the source and returned audio streams to determine network latency.

By setting the linear timecode (LTC) in a desired audio editing programme to display in milliseconds, the round trip latency can be determined by observing the offset of the start time of the recorded audio as compared to the initial source audio file as shown in Figure 5.8.

5.2 Comparing the Performance of JackTrip and WebRTC

JackTrip and WebRTC are both viable platforms for Internet-based audio streaming applications due to their offers of high-quality media distribution with low-latency. JackTrip is presented as an effective tool for online jamming, allowing musicians in various remote locations to play instruments together and engage in real-time musical performances over the Internet (Cáceres and Chafe, 2010). These performances are perceived as synchronous with minimal, if any, noticeable timing differences despite large physical distances. In comparison, WebRTC is widely used for online video chat applications that offer similar benefits to Skype, allowing video and voice conversations to occur naturally and in real-time through a web browser (WebRTC, 2011b). The transfer of high quality audio with low latency is the driving appeal for both platforms; however, they differ in the fact that JackTrip caters more towards music applications, which includes retaining the accurate frequencies of musical instrument sounds. WebRTC conversely employs mechanisms to optimise voice conversations, including codecs such as the iSAC and iLBC audio codecs by Global IP Solutions that are incorporated into many voice over I.P (VoIP) applications (WebRTC, 2011b).

In a first test, the three source audio files were transmitted between two networked computers utilising both Jacktrip and WebRTC, and a recording of each audio stream was captured at the destination computer allowing the results to be compared. The tests were conducted within a controlled environment using computers connected to the high-speed Local Area Network (LAN) by physical Ethernet connections at Anglia Ruskin University, Cambridge, UK. Download and upload speeds of the host and client computers were measured using the Ookla Speed Test (http://www.speedtest.net) and are shown in Figure 5.9.

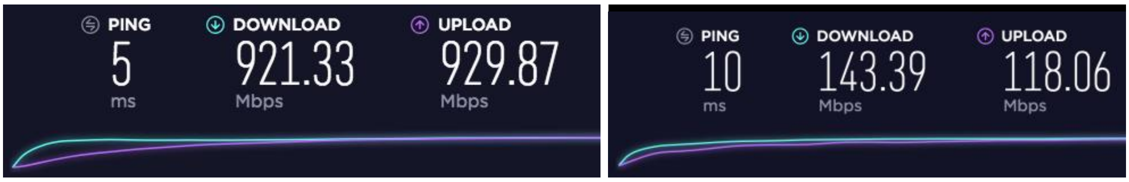


Figure 5.9 Host computer (left) and client computer (right) LAN speed tests.

5.2.1 JackTrip Results

In order to conduct the streaming tests, a Mac Pro desktop computer housed at Anglia Ruskin University was configured as a server computer with a fixed I.P. address to allow streaming connections from computers both internal and external to the network. A secondary Macbook Pro laptop placed on the same LAN using an Ethernet connection was configured as a JackTrip client and connected to the server. The tests produced 5 recordings of each audio sample, resulting in 15 recordings total (See Appendix U).

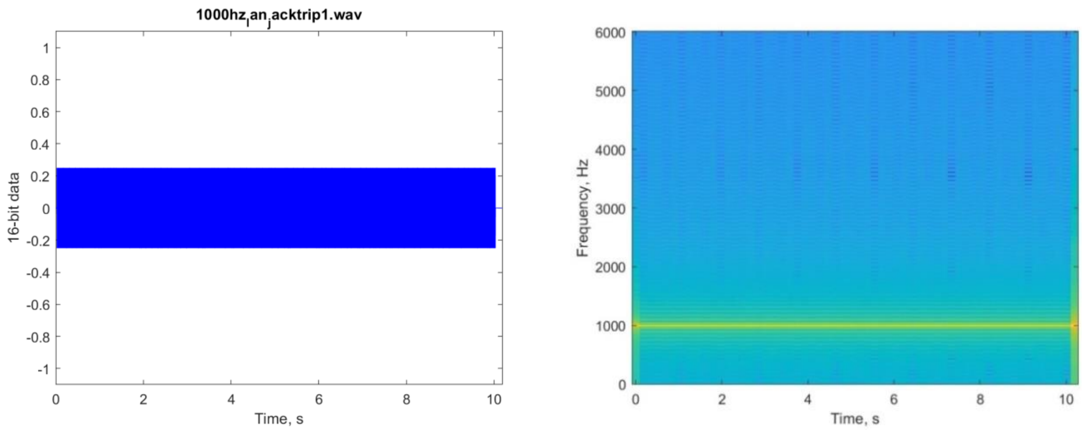


Figure 5.10 Example 1 kHz sine wave LAN capture waveform and spectrogram with JackTrip.

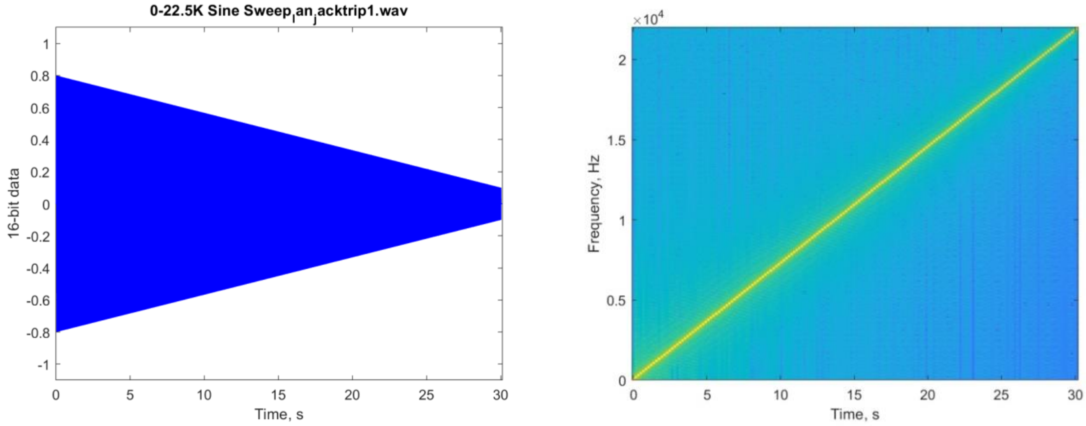


Figure 5.11 Example 0-22.5 kHz sine sweep LAN capture waveform and spectrogram with JackTrip.

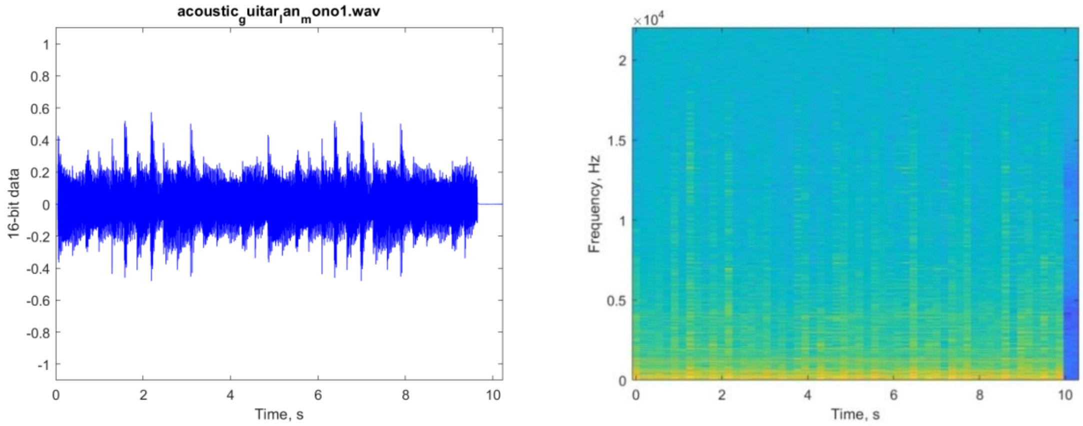


Figure 5.12 Example acoustic guitar LAN capture waveform and spectrogram with JackTrip.

5.2.2 WebRTC Results

Using a WebRTC media broadcast webpage configured at http://mjhardin.com, the server and client computers were allowed to transmit audio streams between each other through the web browser.

The tests were repeated 5 times each per sound file with full results documented in Appendix V.

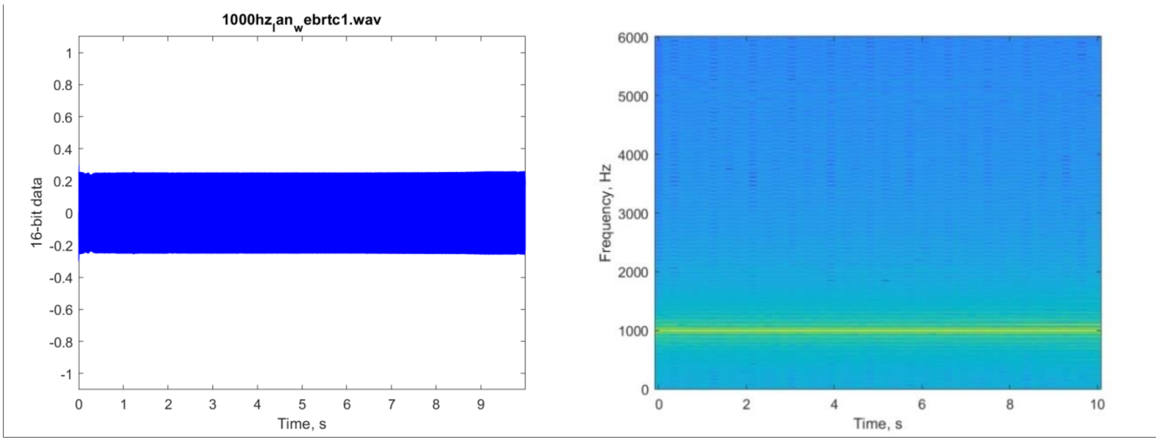


Figure 5.13 Example 1 kHz sine wave LAN capture waveform and spectrogram with WebRTC.

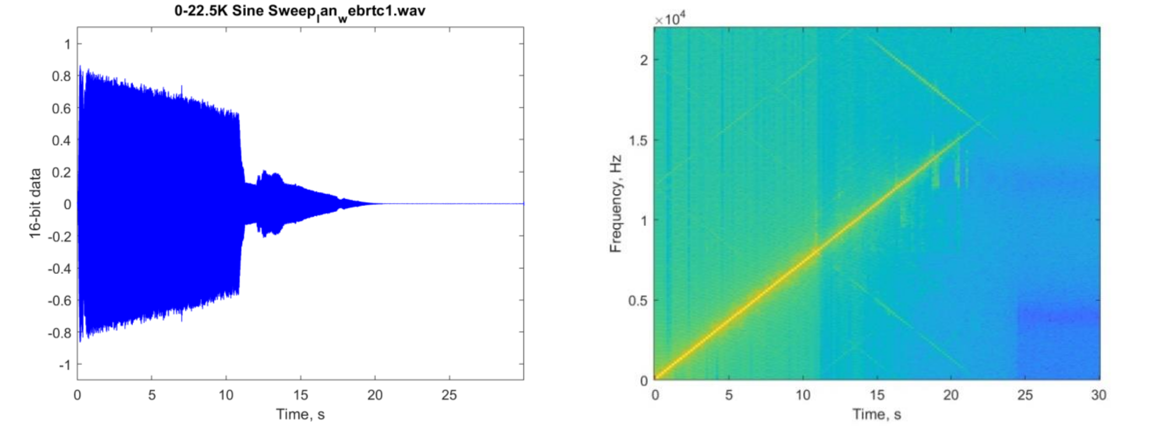


Figure 5.14 Example 0-22.5 kHz sine sweep LAN capture waveform and spectrogram with WebRTC.

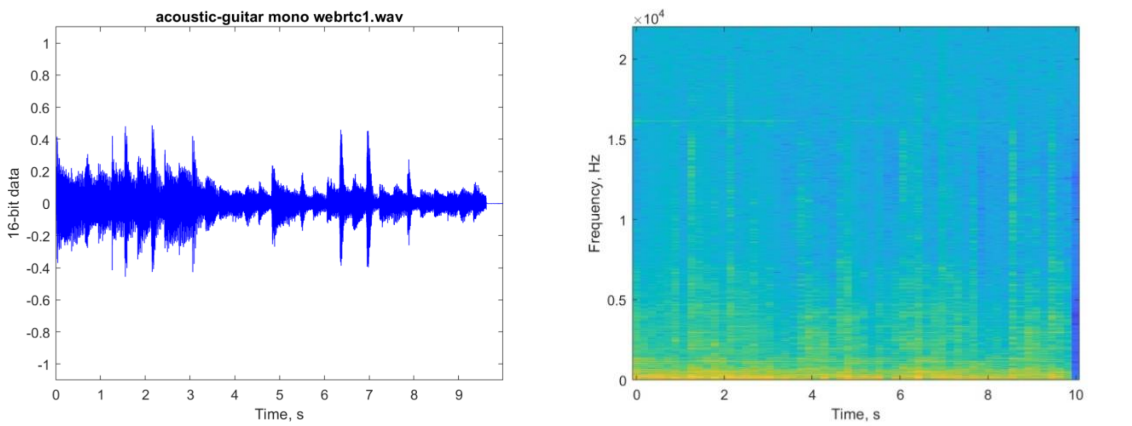


Figure 5.15 Example acoustic guitar LAN capture waveform and spectrogram with WebRTC.

Since Hardin and Toulson (2019), the WebRTC test page became inactive due to deprecated RTC functions. Original WebRTC dropout, THD+N, and latency results are provided below, but follow up measurements using the 0.0048 THD+N sine wave was conducted using *Talky.io* and provided in Appendix DD.

5.2.3 JackTrip and WebRTC Dropout Comparison

The number of dropouts was evaluated for the sinewave source file transmitted between local computers using both JackTrip and WebRTC. The average number of dropouts is calculated from five repeats of each test.

Table 5.1 Dropouts for sinewave tests on LAN.

	Number of Dropouts					
Platform	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
JackTrip	0	0	0	0	0	0.0
WebRTC	0	2	0	1	0	0.6

The table above shows that the 1 kHz sinewave test presented no audio dropouts, showing effective, real-time audio streaming within the local area network. This demonstrated that the local area network was robust enough to appropriately deliver audio from source to destination without interruptions or detectable errors.

Similar to the JackTrip test, WebRTC provided very minimal audio dropouts with the 5, 1 kHz sinewave tests. Observing the waveform of test 4, there is also a possible false dropout reading due an initial, unforeseen amplification of the signal at the start of the transmission. While small errors were detected, a majority of the tests were conducted without incident.

5.2.4 JackTrip and WebRTC Distortion Measurements

Distortion was calculated for the sinewave source file transmitted between local computers using JackTrip and WebRTC. The average distortion is calculated from five repeats of each test.

Table 5.2 Distortion measurements for sinewave tests on LAN.

	THD+N (%)					
Platform	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
JackTrip	0.0048	0.0048	0.0048	0.0048	0.0048	0.0048
WebRTC	0.3870	1.2633	0.5180	0.6178	0.4668	0.6506

The distortion measurements of JackTrip matched that of the source sinewave, showing no additional distortion. WebRTC produced much higher distortion measurements, demonstrating the platform does alter the original signal.

5.2.5 Listening Tests

The results of the quantitative analysis detail unbiased observations of the capacities of JackTrip and WebRTC for real-time audio and music transfer over the Internet. However, audio and music listening experiences are very subjective, and impressions of these experiences will differ, sometimes significantly, from person to person. With this in mind, it was important to explore how individuals *perceive* the differences in the audio captures, and determine if human perception provides different feedback regarding the quality of JackTrip and WebRTC. Thus the creation of a listening test served to provide subjective data regarding JackTrip and WebRTC comparisons to the source audio.

Toole (1982) describes 3 scenarios for the ideal listening tests:

1. The tests need to be “reproducible at different times and places and with different listeners”,
2. They need to “reflect only the audible characteristics of the product or system under examination”,
3. And the tests need to “reveal the magnitude of audible differences or a measure of absolute values on appropriate subjective scales.”

Keeping in line with these objectives, a listening test page was created at: http://mjhardin.com/listeningtest

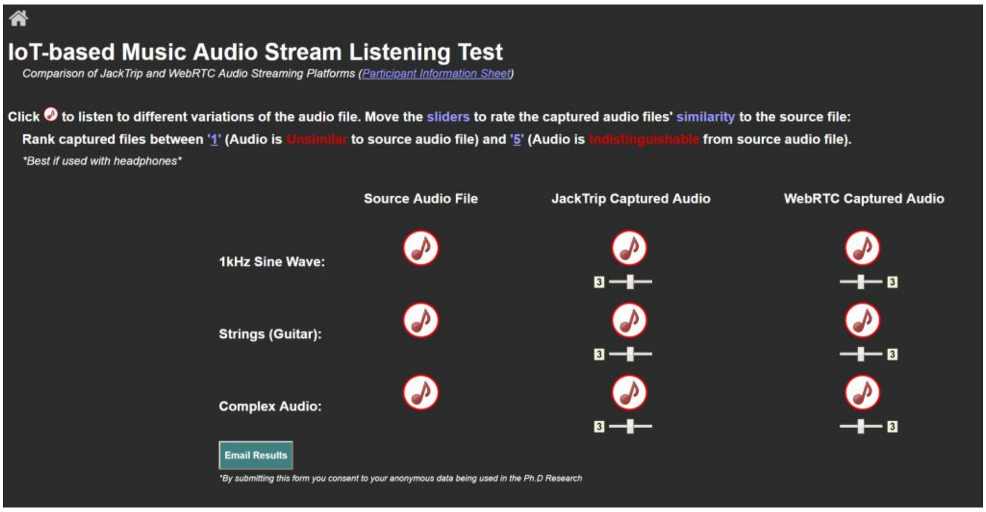
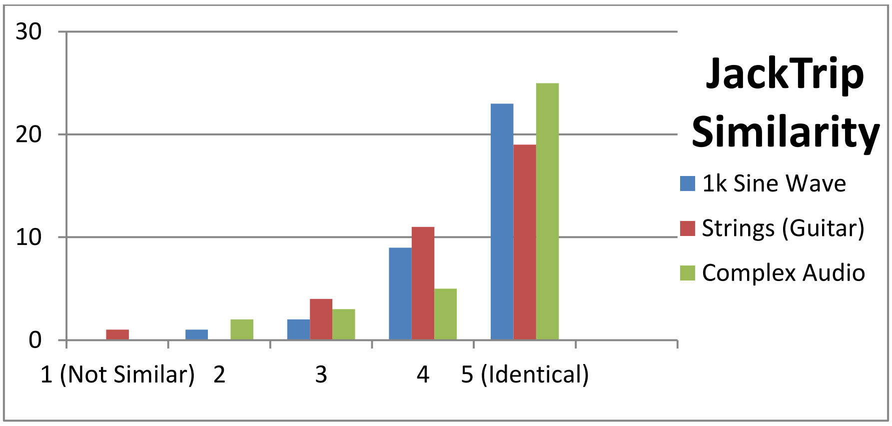


Figure 5.16 JackTrip vs WebRTC listening test webpage.

The International Telecommunication Union (ITU) recommends that the best and most versatile audio sources for listening tests are taken from computer-controlled digital storage systems, and the online audio distribution platform, *SoundCloud,* provided a means to stream audio files over the Internet that are uploaded to their servers (Rec, I.T.U.T., 1996). SoundCloud additionally offers an API to allow their streams to be built into independent applications and music players, thus allowing the design of a tailor-made webpage interface for the sole purpose of comparing streams.

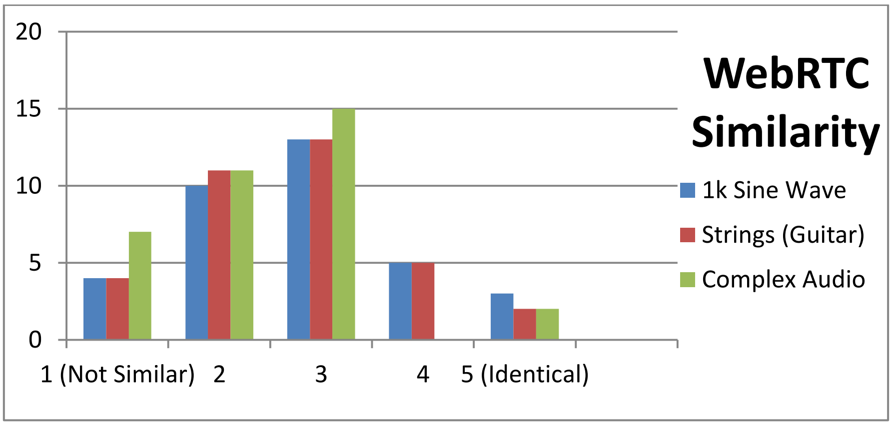
Within the listening test webpage users can listen to 3 variations of 3 audio files (the source, JackTrip capture, and WebRTC capture of the 1 kHz sine wave, acoustic guitar, and an additional complex audio file) and compare the JackTrip and WebRTC audio streams against the original source file. The users are provided with no additional information regarding the streaming platforms within the confines of the test to prevent impaired biases, and a continuous quality scale ranking of numerical values 1 to 5 are used to determine how dissimilar to similar the audio recordings are to the source audio (Series, B., 2014). When the test is concluded, the results can be emailed anonymously for collection and evaluation. The listening test is conducted with the general public in mind, and is not limited to a specific audience with the intention of collecting results from a wide spectrum of backgrounds with diverse experiences with music. Responses were obtained from 35 respondents with results presented in Figures 5.17 and 5.18.

Number of Responses = 35



	Similarity Ranking Responses				
Audio File	**1**	**2**	**3**	**4**	**5**
1 kHz Sine	0	2	3	5	25
Acoustic Guitar	1	0	4	11	19
Complex Audio	0	1	2	9	23

Figure 5.17 Public perceptions of JackTrip captures' similarity to source audio file.



	Similarity Ranking Responses				
Audio File	**1**	**2**	**3**	**4**	**5**
1 kHz Sine	4	10	13	5	3
Acoustic Guitar	4	11	13	5	2
Complex Audio	7	11	15	0	2

Figure 5.18 Public perceptions of WebRTC captures' similarity to source audio file.

In regards to the JackTrip listening tests, 91.4% of participants gave the sinewave a ‘4’ or ‘5’, identifying that the recorded sine was very similar or identical to the source audio. Both of the music samples received high rankings of a ‘4’ or ‘5’ from 85.7% of respondents equally. In only one case of the acoustic guitar recording did a respondent perceive the sound was completely dissimilar to the source audio. Due to subjective nature of these tests there was never an expectation of a perfect result, however, the high percentages of listeners ranking similar audible attributes of JackTrip recording to original source audio files showed promise that the platform upheld the quality of its audio streams.

WebRTC, on the other hand, offered more neutral responses from respondents, with over 65% of respondents ranking each file a ‘2’ or ‘3’. Roughly 7% of responses on average identified the 3 recordings as identical to their complementary source files. While this does not necessarily represent a negative listening experience, it showed that most listeners can perceive some difference in the audio streams from WebRTC.

5.2.6 Discussion of JackTrip and WebRTC Performance

The analysis of the JackTrip audio recordings show that audio streamed across Jacktrip accurately resembles the source audio file. The waveforms of the transmitted audio are similar to the original, showing that JackTrip correctly models the characteristics of the source audio without any additional filtering or processing during transmission. The listening tests confirmed that a significant amount of respondents observed similarities between the JackTrip and the source audio file, with many perceiving no differences as compared to the noticeable differences reported to be perceived from the WebRTC audio samples. Additionally, the spectrograms show that the true frequencies of the source audio are maintained better using JackTrip than WebRTC. With the exception of the second sine sweep test, JackTrip produced no noticeable network drops or spikes that represent streaming errors in 14 of the 15 recorded trials.

The waveforms of the WebRTC captures do, however, show explicit differences from that of the source audio file, and the listening test reveals that these differences are perceived audibly. Regarding the sine sweep, the WebRTC captures are initially compressed and the audio becomes non-existent at higher frequencies, showing signs of higher frequency filtering. These characteristics correlate with WebRTC’s use of VoIP codecs for video and voice chat scenarios. Additionally, the acoustic guitar audio streamed using WebRTC showed some attenuation of the original signal throughout each of the trials. While music transmitted with WebRTC may not be an issue for some applications, the fact that the

WebRTC captures do not accurately reflect the original source audio makes it unsuitable for the high quality music applications presented in this research.

These tests inherently showed a higher quality of performance from JackTrip compared to WebRTC for real-time audio streaming, and also demonstrated that streaming over the LAN is very reliable and sufficient for IoT-based music applications.

5.2.7 Local JackTrip and WebRTC Latency Measurements

WebRTC presents challenges in implementation without the support of public stun/turn servers. A JackTrip server, in comparison, can be set up directly within a local area network. Standard roundtrip latency measurements for JackTrip can be taken by measuring the delay between the source file and the recorded stream using a loopback server as outlined in section 5.1.4.

JackTrip requires an audio stream input/output (ASIO) driver, named *JackRouter,* to transfer audio from source to destination. It conceptually acts as a virtual audio cable allowing users to route audio through desired input and output paths between a client and server computer. JackTrip additionally has a server configuration that allows a server computer to loopback any audio sent to it, which combined with JackRouter allows the client computer to route the looped back audio into an audio editor where it can be recorded and compared to the source audio it transmits. Average latency speeds were calculated for JackTrip over the local using five repeats of this test.

Table 5.3 Latency measurements over LAN.

JackTrip LAN Latency (ms) Using Loopback Server					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
8	2	8	2	2	4.4

Audio transmission over the local area network using the JackTrip loopback server provided a very low average of 4.4 ms roundtrip latency.

In a real world scenario that requires streaming audio from a client computer to a remote music processor, the processor may need to be connected to an additional audio interface at the remote end. A second test was conducted streaming audio from a client computer to a server using JackTrip, but the audio was routed back to the client through JackTrip using a T.C. Electronic *Impact Twin* audio interface instead of the JackTrip loopback server. Results are below:

Table 5.4 JackTrip Round Trip Latency measurements using an Audio Interface.

JackTrip LAN Latency (ms) Looped Through Audio Interface					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
33	22	33	33	22	28.6

In this scenario, the audio interface did add roughly an additional 20 to 25 ms of latency to the audio stream, however, the average was still below the standard threshold of human echo perception.

While not a true representation of a local area test due to support from external servers, roundtrip WebRTC latencies were also obtained. In contrast to JackTrip, WebRTC relies on the pre-existing audio output devices within a PC as a path to transmit audio through the browser. An audio editor can use the same output device as an input to record audio being looped back to the client, but this causes signal routing errors as the editor will try to simultaneously record the audio it transmits as well as the audio looped back, providing inaccurate latency measurements. To overcome this issue, WebRTC latency measurements were captured by virtually streaming audio from the client computer to the server using the enabled browser, however, the two computers needed to be placed physically nearby each other and the streamed audio was routed back from server to client using a 2 meter audio cable. Any delay added by the cable would be miniscule as a sound signal propagates through a cable at approximately the speed of light (Fonseca and Monteiro, 2003, p. 2).

Table 5.5 WebRTC Latency Measurements.

WebRTC Roundtrip Latency (ms) using a physical loopback					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
85.5	92	109.5	80.5	89.5	91.4

WebRTC latency measurements averaged around 91 ms of delay, possibly presenting some challenges for real-time music applications as it is well above the threshold of echo perception by the human ear.

5.3 Commercial/Commodity Network Testing

JackTrip showed capabilities to support high quality networked audio distribution and objectively outperformed WebRTC for measured and perceived audio quality. As a result, the next tests were organised to determine network configurations that best support audio transmission using JackTrip exclusively. While the LAN-based tests presented an adequate baseline for observing the capability of real-time audio streaming over a computing network, the following set of trials examined JackTrip streaming capabilities in real-world environments independent of the Local Area Network and incorporated external computers housed outside of Anglia Ruskin University, extending distributed music applications into the Wide Area Network (WAN). The tests evaluated Jacktrip audio streaming on the commercial, commodity Internet from a residential home in London, UK to Anglia Ruskin University in Cambridge, UK. A speed test of the client computer (Figure 5.19) showed very high upload speeds, while download speeds were fairly low in comparison to the LAN network (only 4.58 Mbps), presenting a possible bottleneck for high speed audio transfer.

Figure 5.19 External client computer speed test on commercial network.

5.3.1 Investigating JackTrip Performance with Different Buffer Sizes

A number of JackTrip buffer settings were investigated, with buffer sizes of 128, 256, 512 and 1024 being compared for performance regarding dropouts, distortion, and latency. While a small buffer reduces the amount of time audio interfaces like JackTrip need to store and transmit audio packets to ultimately reduce latency, Robjohns (2008) explains that "software problems often stem from the audio interface RAM buffers being too small, and the data running out before the operating system can get back to top them up (playback) or empty them (recording)." Small buffer sizes can cause audio interfaces to run out of data to transmit before having the option to transmit it, and this can result in errors such as clicks and pops, or gaps of silence while the buffer is refreshed.

The first test of the wide area audio streams was conducted with a 128 size buffer and produced many audible errors, representative of loud clicks and large audio spikes in the captured recordings. These are evident as broadband noise periods in the spectrograms and can be seen in the figure below alongside the spectrogram of the source 1 kHz sine wave.

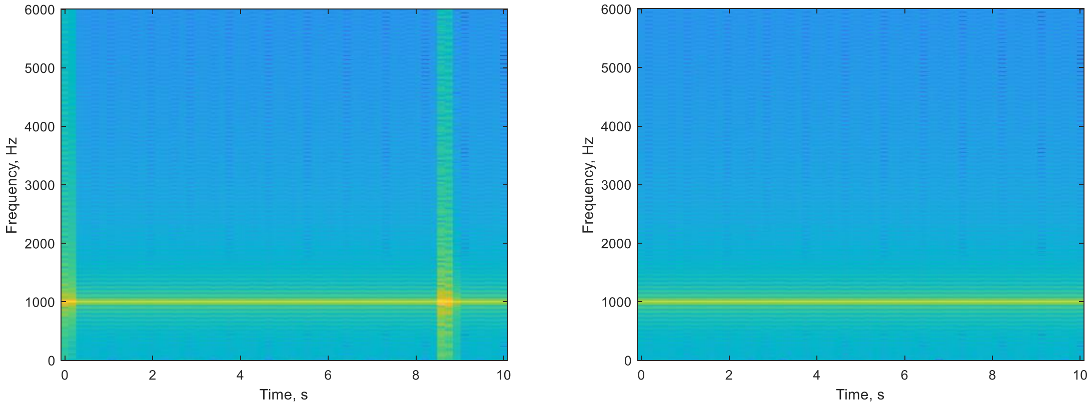


Figure 5.20 Spectrogram of example 1 kHz sine wave capture on commercial network with 128 sample buffer size (left), compared to source audio (right).

As expected, increasing the buffer size to 256 samples provided major improvements in the results as represented by the spectrogram below.

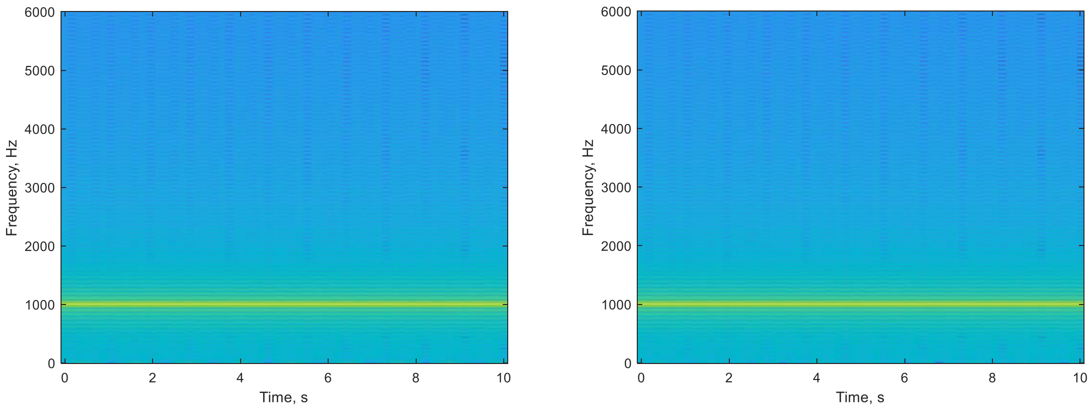


Figure 5.21 Spectrogram of example 1 kHz sine wave capture on commercial network with 256 sample buffer size (left), compared to source audio (right).

Larger buffer sizes showed successful reduction in the errors in the audio streams. Further increases of the buffer size to 512 and 1024, however, prompted warnings of latency over 30 ms from JackTrip as shown in Figure 5.22.

```
Waiting for Peer...
Received Connection for Peer!
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
UDP waiting too long (more than 30ms)...
```

Figure 5.22 JackTrip latency warnings for delays over 30 ms.

The complete set of spectrogram results from the public network streaming tests can be found in the Appendix.

5.3.2 Dropouts at Different Buffer Sizes on Commodity Network

The number of dropouts was evaluated for the sinewave source file transmitted with different JackTrip buffer sizes. The average number of dropouts is calculated from five repeats of each test.

Table 5.6 Dropouts for sinewave tests on commercial network with different JackTrip buffer sizes.

	Number of Dropouts					
Buffer Size	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
128	12	4	23	13	18	14
256	0	0	0	0	0	0
512	0	0	0	0	0	0
1024	0	0	0	0	0	0

The initial 128 sample buffer produced a higher number of dropouts than the tests conducted over the LAN, and dropouts were consistent in each test. As shown by the spectrograms, these drops produced audio spikes and broadband noise that can make the listening experience very unpleasant. Increasing the buffer size drastically improved the results, eliminating the drops over the set of the sinewave tests.

5.3.3 Distortion Measurements at Different Buffer Sizes on Commodity Network

Distortion was calculated for the sinewave recordings of the source file transmitted over the commodity Internet with different JackTrip buffer sizes. The average distortion is calculated from five repeats of each test.

Table 5.7 Distortion measurements for sinewave tests on commercial network with different JackTrip buffer sizes.

	THD+N (%)					
Buffer Size	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
128	5.2090	1.9509	2.4970	2.5573	3.0517	3.0532
256	0.0048	0.0048	0.0048	0.0048	0.0048	0.0048
512	0.0048	0.0048	0.0048	0.0048	0.0048	0.0048
1024	0.0048	0.0048	0.0048	0.0048	0.0048	0.0048

The distortion measurements were high for the 128 sample buffer. When the buffer size was increased, the distortion was minimised and matched the source.

5.3.4 Latency Measurements at Different Buffer Sizes on Commodity Network

Roundtrip latency measurements were taken at different buffer sizes. The average latency was calculated from five repeats of each test.

Table 5.8 Latency measurements on commercial network with different JackTrip buffer sizes.

	Roundtrip Latency (ms)					
Buffer Size	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
128	30	30	30	25	25	28
256	30	30	30	30	30	30
512	54	42	42	42	42	44.4
1024	112	112	112	112	112	112

The 128 and 256 sample buffer sizes both achieved low latency rates. Further increases in buffer sizes helped maintain streaming quality, but introduced greater latency above 30 ms.

5.3.5 Commercial/Commodity Network Audio Streaming Discussion

The commodity network initially displayed challenges supporting real-time audio streaming with low buffer sizes, but significantly improved when the buffer size was increased. It is possible that the lower bandwidth designations and overall network congestion on commercial networks can create more opportunities for dropped packets and errors in the data stream (Sinatra, 2014). Also, while these tests overall performed very well, the results differed from initial results seen in Hardin and Toulson (2019), showing that network congestion on different dates may severely impact performance. Additionally in addressing errors in Hardin and Toulson (2019), the use of larger buffers on commercial networks may create a scenario called *bufferbloat*, where latency increases due to excessive buffering of packets and ultimately reduces network throughput and successful message delivery. Nonetheless, these tests showed that in optimal conditions the current commercial/commodity computing networks have some capability of supporting real-time high quality, low latency music transfer, although comparisons to Hardin and Toulson (2019) prove that results may vary over time.

5.4 High-Speed Network Testing

A National Research and Educational Network (NREN) is a high-speed computing network operated amongst the educational and research communities inside a country and serves as a backbone to interconnect higher educational institutions within the country and to other research networks abroad (Foley, n.d.). NRENs provide higher bandwidths suitable for transferring large data sets and have shown past success in low-latency audio streaming applications with JackTrip, demonstrated by a series of ‘telematic’ music performances hosted by the SoundWIRE Group at Stanford University (SoundWire Group, 2010).

JANET is the designated high-speed NREN supplied by the Joint Information Systems Committee (JISC) in the United Kingdom. The next set of streaming tests evaluated the quality of audio transmission incorporating a JANET-based client computer external to Anglia Ruskin University. The tests were conducted between the University of Westminster in London, UK and Anglia Ruskin University in Cambridge. Measured network speed of the JANET-based client computer is shown in Figure 5.23.

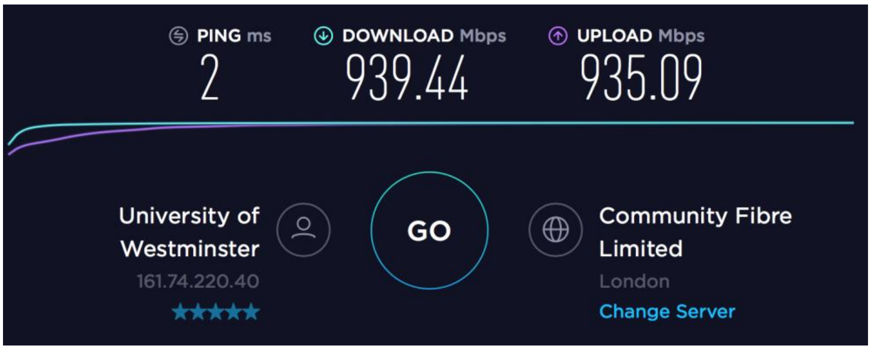


Figure 5.23 Client Computer Speed Test over JANET.

As a 256 sample buffer size was determined to be effective for streaming in previous experiments, this setting was maintained for subsequent testing. Examples of the spectrograms are shown in the figures below:

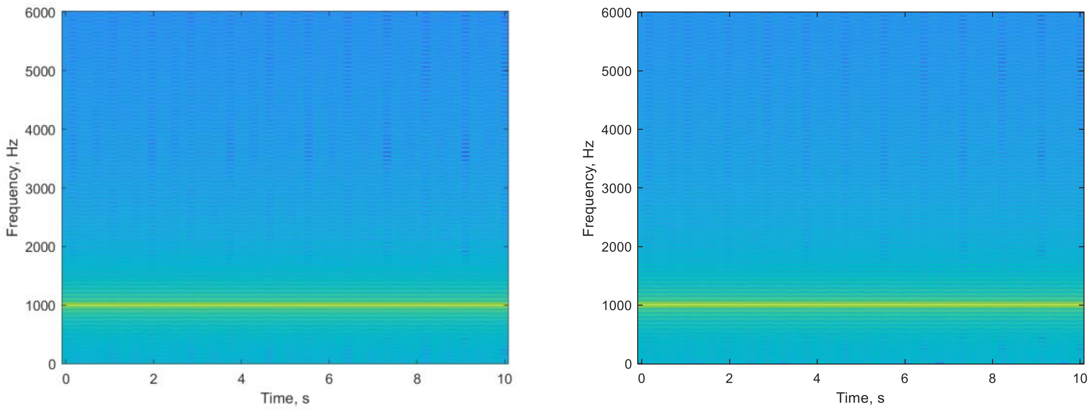


Figure 5.24 Spectrogram of example 1 kHz sine wave capture over JANET network (left), compared to source audio (right).

The complete set of spectrogram results from the JANET network streaming tests can be found in Appendix AA.

5.4.1 Dropouts on Research and Educational Network

The number of dropouts was evaluated for the sinewave source file transmitted using JackTrip over the JANET high-speed research and education network. The average number of dropouts is calculated from five repeats of each test.

Table 5.9 Dropouts for sinewave tests on JANET network.

Number of Dropouts					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
0	0	0	0	0	0

The initial sets of testing showed no signs of drop outs. Future tests were additionally conducted on separate dates to determine reliability and yielded the same results.

5.4.2 Distortion Measurements on Research and Educational Network

Distortion was calculated for the sinewave source audio transmitted over JANET. The average distortion is calculated from five repeats of each test.

Table 5.10 Distortion measurements for sinewave tests on JANET network.

THD+N (%)					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
0.0048	0.0048	0.0048	0.0048	0.0048	0.0048

The THD+N measurements over the JANET-based networked matched that of the source audio file, showing that no distortion occurred during transmission.

5.4.3 Latency Measurements on the Research and Educational Network

Roundtrip latency measurements were taken for audio transmission over JANET. The average latency was calculated from five repeats of each test.

Table 5.11 Latency measurements on JANET network.

Roundtrip Latency (ms)					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
30	25	36	31	25	29.4

Roundtrip latency measurements overall averaged lower than 30 ms and showed adequacy for remote music processing and production.

5.4.4 High-speed Network Audio Streaming Discussion

The high-speed network tests showed great promise in real-time music distribution. Minimal audio errors were observed and round trip latency was kept at a minimum to ensure unperceivable delays in transmission. Outside investigations of technology for networked music performances have additionally supported these results. A separate real-time media platform, the LoLa audio video system, was purposely designed to exploit the bandwidth capacity and robustness of dedicated high performance networks like JANET, requiring at least 100 Mbps throughput (Drioli, Allocchio, and Buso 2013), and has allowed remote musicians to play comfortably together at distances up to 3000 Km (Ferguson, 2015).

These findings show that media distribution over NRENs performs at a higher level than commodity networks, and presents possibilities of implementing an IoT-enabled music system with real-time audio transfer when supported by high-speed LAN and NREN networks.

5.5 Wireless Network Testing

The development of wireless Internet networks has made computing resources widely available for mobile applications and allows numerous complex digital processes to occur fluidly in the real world without the need to be tethered to a specific location. On-the-go processes allowing more engagement in personal and professional workflows have been greatly enhanced by portable laptops, mobile smart phones, and tablets. While hard-wired Ethernet connections effectively enable frameworks for remote networked music applications, the following Wi-Fi tests determine effectiveness for on-the-go music production scenarios.

The Wi-Fi audio streaming tests consists of a client laptop utilising Anglia Ruskin University's internal wireless supported by JANET's high speed network connecting to the existing server computer on the internal wired network.

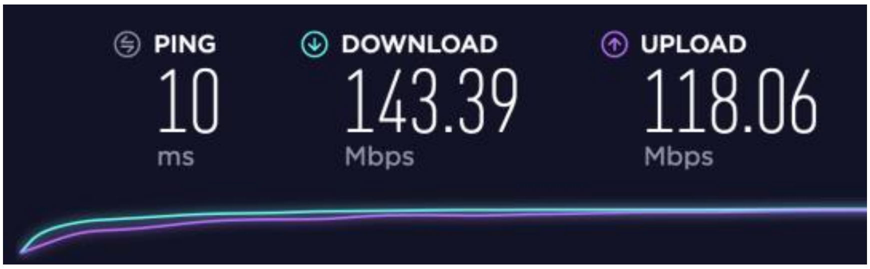


Figure 5.25 Client computer Wi-Fi speed test.

Examples of the audio streaming tests over Wi-Fi are shown below with full results provided in Appendix BB.

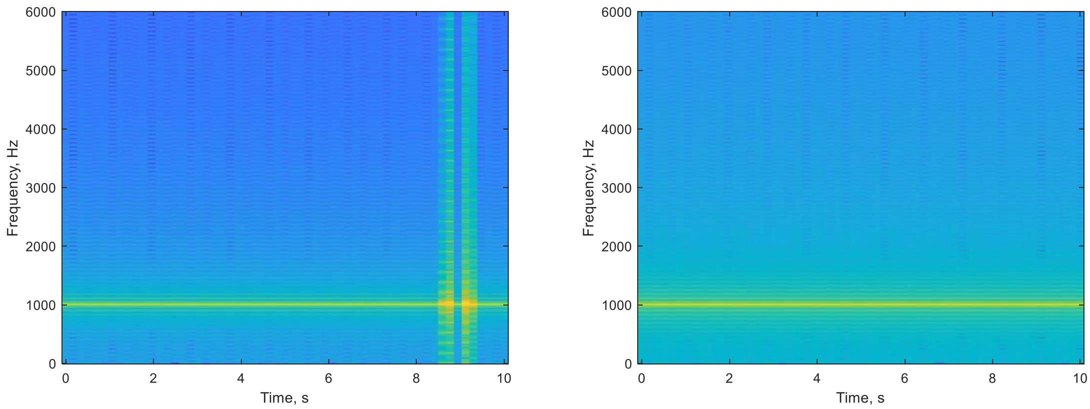


Figure 5.26 Spectrogram of example 1 kHz sine wave capture over Wi-Fi network (left), compared to source audio (right).

5.5.1 Dropouts on Wi-Fi

The number of dropouts was evaluated for the sinewave source file transmitted from JackTrip over Wi-Fi. The average number of dropouts is calculated from five repeats of each test.

Table 5.12 Dropouts for sinewave tests on Wi-Fi.

Number of Dropouts					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
34	60	25	15	24	31.6

The Wi-Fi results faired far worse than streaming audio using physical network connections and yielded the highest number of dropouts of all the tests. Additionally there were no successful, error-free streams over the 5 Wi-Fi trials, showing that real-time wireless audio streaming is highly unreliable.

5.5.2 Wi-Fi Distortion Measurements

Distortion was calculated for the sinewave source file transmitted from JackTrip over Wi-Fi. The average distortion is calculated from five repeats of each test.

Table 5.13 Distortion measurements for sinewave tests on Wi-Fi.

THD+N (%)					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
1.2938	2.8277	3.3925	1.2570	1.6257	2.0793

Distortion measurements over Wi-Fi were very high. This is most likely a result of the broadband errors caused by dropouts that distorted the signal stream.

5.5.3 Wi-Fi Latency Measurements

Roundtrip latency measurements were conducted over Wi-Fi. The average latency was calculated from five repeats of each test.

Table 5.14 Latency measurements over Wi-Fi.

Roundtrip Latency (ms)					
T1	**T2**	**T3**	**T4**	**T5**	**Average**
8	8	14	14	14	11.6

While Wi-Fi connections introduced low starting latencies, JackTrip would often prompt the ‘UDP waiting too long” warnings shown in Figure 5.22 as the recordings occurred. This would sometimes result in the tail end of file ranging from 30 to 100 ms longer than the 10 second source sine wave.

5.5.4 Wi-Fi Audio Streaming Discussion

The Wi-Fi audio streaming tests had very little success compared to the hard-wired Ethernet tests over the same LAN. In addition to clicking sounds present in the recordings and visible spikes in the spectrograms, some recordings had deep buzzing sounds that were not observed in the other network tests. The issue with audio streaming over Wi-Fi, inherently, has much to do with the protocol used for low-latency transmission.

There are two transport protocols used to transmit data packets over the internet: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is the primary protocol used for standard Internet traffic, and the benefits include packet numbering to make sure packets reach their destination in the desired order, and additionally error tracking and retransmission of packets that lost over the network (Hoffman, 2017a). This creates efficiency and greater reliability for Internet communication, including the upload and download of data. However, these methods for data redundancy unfortunately add latency to network communication.

In comparison, UDP offers data packet transmission without the error checking. This means there is no confirmation that data packets have arrived to their destination and no retransmission of missing packets; new packets are continuously delivered with no redundancy for lost data. However, without the overhead of error tracking, what UDP losses in reliability it makes up for in speed (Hoffman, 2017a). Data can be delivered more rapidly across the Internet using the UDP framework and makes it ideal for real-time and low latency audio streaming. As a result UDP is used for audio transmission over JackTrip and it is the default protocol for WebRTC data.

The lack of error checking is not a major issue on a stable Internet connection. The highest network speeds a Cat-5e Ethernet cable provides data exchanges up to 1 Gb/s (10 Gb/s using Cat 6) while currently the fastest wireless router offers max speeds of 866.7 Mb/s (Hoffman, 2017b). Speeds over 100 Mb/s are more than enough to support high quality audio streaming, however, a big difference between Ethernet and wireless is the reliability of the connection. An Ethernet connection provides a straight physical connection between a computing device and the Internet. However, Wi-Fi is a radio signal that is subject to interference from other wireless broadcasting devices. These interferences often result in latency due to competing network traffic and worse, dropped signals (Hoffman, 2017b). While dropped signals are often reacquired very quickly and usually unnoticed, the lack of error checking in UDP transmission tend to create pronounced and noticeable errors in audio streaming when these packets are lost. Consequently, until the reliability and robustness of wireless networks are improved, high-quality, real-time audio transfer still favours wired connections in modern networks.

6. User Insight Evaluation

The user insight evaluation presents statistics, views, and impressions of music producers regarding IoT-enabled audio systems and the opportunity to remotely engage music processing systems interconnected to the Internet. Data is collected through a mixed-method process using questionnaire feedback and expert interviews. The findings portray a more insightful and qualitative exploration into the impacts of the Internet of Things for producers across a variety of musical backgrounds.

6.1 Questionnaire Evaluations Considerations

6.1.1 Design of Questionnaire

The questionnaire is designed using a mixture of closed and open ended questions that firstly seek to understand the personal production practices of sampled music producers and secondly aim at capturing perspectives of an IoT-enabled music system. The first 6 questions collect background information regarding the respondents' approaches and experiences in music production and identify their preference for hardware-driven or digital techniques in normal mixing conditions. The remaining 4 questions collect feedback directly pertaining to IoT-enabled audio applications; gathering deeper perspectives on perceived use cases and how mainstream implementation of these systems could promote greater engagement with audio processing hardware. An optional comments section is also available for additional thoughts not addressed in the previous questions.

The target audience for the user insight questionnaires are:

1. Novice and casual music makers
2. Experienced musicians and music producers

The first target group involves casual and hobbyist musicians, or novice music students who benefit from the modern-day emergence of digital music technology and may or may not utilise the traditional music studio environment for production purposes. The ubiquitous nature of computing devices has made digital music software a preferred method for mixing music by many music makers, and this research explores if greater opportunities to access and utilise music production hardware would factor into increased value and engagement in analogue and physical music processes. The second target audience are audio and music producers who have knowledge and field experience using various platforms, including both analogue hardware and digital technology, to make music. This group can provide greater insight into the potential usefulness of an IoT-enabled audio system and give higher levels of detail into benefits and challenges that can emerge from these systems. An optional respondent category is available for non-musician feedback that is primarily obtained by creative practitioners from a range of backgrounds (artists, film makers, technologists, etc.) who engage in unique and imaginative processes to generate original work. These individuals may have limited experience in music production, but their understanding of novel and innovative approaches to technology and artistic practices can promote outside interpretations on the impact of IoT applications in creative industries for future discussions not reflected in this chapter.

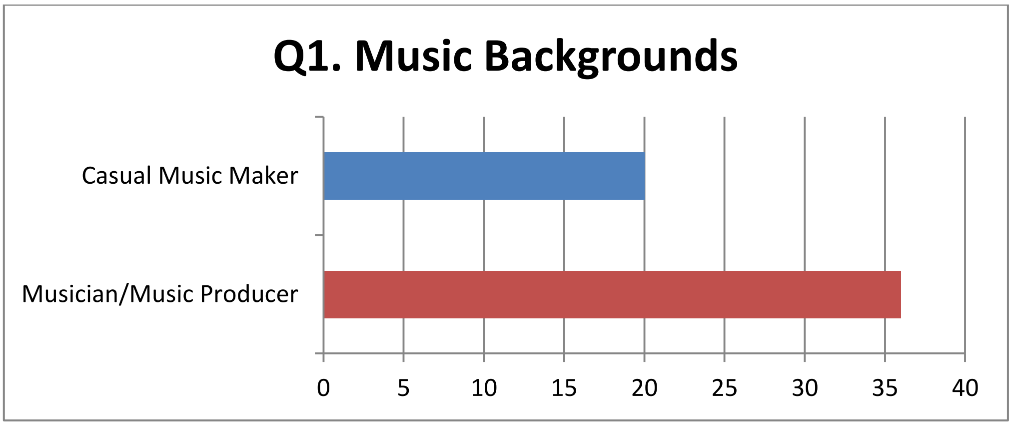
A minimum of 50 respondents were targeted for the questionnaire responses. Tuckett (2004, p. 48) states that "whilst there are no closely defined rules for sample size (Baum 2002; Patton 1990), sampling in qualitative research usually relies on small numbers with the aim of studying in depth and detail (Miles & Huberman 1994; Patton 1990). For the purpose of seeking richness in data about a particular phenomenon, in this case new music production opportunities established by IoT-enabled technology, "the sample is derived purposefully rather than randomly" from collective groups of music producers (Tucket, 2004, p. 48 via Reed et al. 1996; Mays & Pope 1995; Ezzy 2002). Tuckett (2004, p. 49) via Lincoln and Guba (1985) recommends that "sampling continues until the researcher recognises no new data were forthcoming – a point of data or information redundancy."

The sample group was largely composed of music production students within the audio music technology and creative music technology courses at Anglia Ruskin University as well as local music producers in the Cambridge, UK area. The proximity to the researcher and institution provided optimal opportunities for demonstration and evaluation, and the targeted music courses brought about a variance of producers from different skill sets, backgrounds, age ranges, and genders. When the online data collection commenced, known music producers and their colleagues where specifically targeted, as well as members of music professional groups such as the Audio Engineering Society. During the data collection process *data saturation* began to appear when approaching 40 respondents. At this point there was no new information emerging and it was believed that the collected data satisfied the aims of the research goals. A copy of the questionnaire can be found in Appendix EE.

6.1.2 Questionnaire Results and Analysis

The questionnaire produced a total of 56 responses, where an acceptable saturation of data trends was observed. The questions are listed below with a brief analysis of the responses:

Q1. Describe Your Musical Status



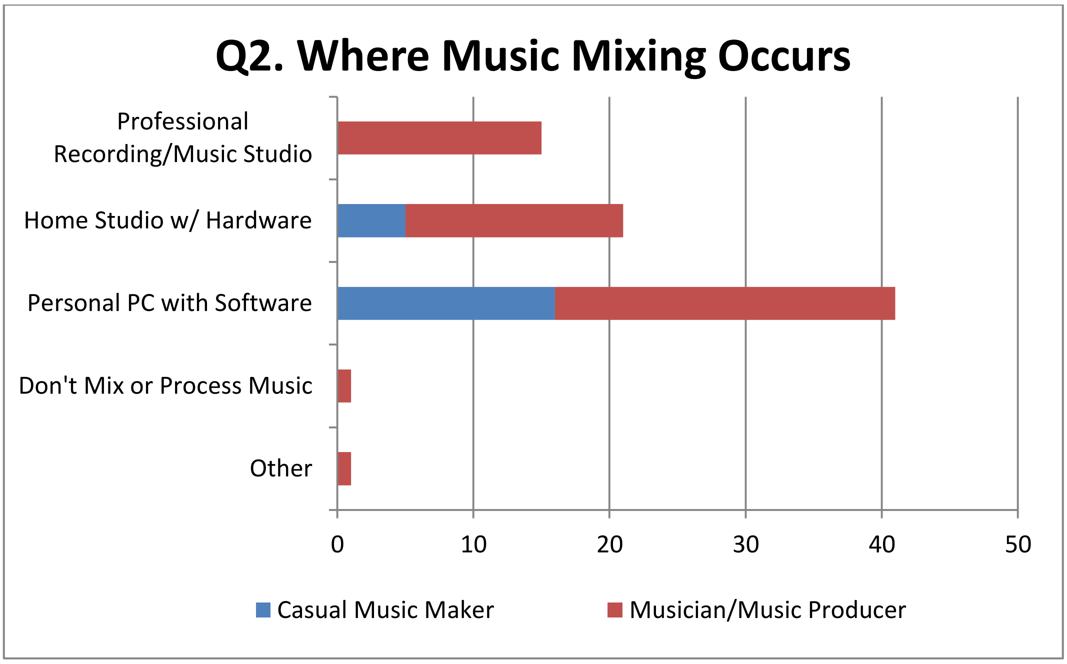
	Casual Musician	Music Producer	Total
# of Participants	20 *(36%)*	36 *(64%)*	56

Figure 6.1 Music backgrounds of questionnaire participants.

56 responses collected total: 20 casual musicians makers, 36 music producers.

Note: 1 participant identified as a non-musician or "Other Creative Practitioner." However, due to having some experience in music processing and enrolment in a university audio degree programme, it was best to include the respondent's feedback with the casual music maker data.

Q2. If You Mix and Process Your Own Music, Where Does This Normally Occur? (Multiple Answers)



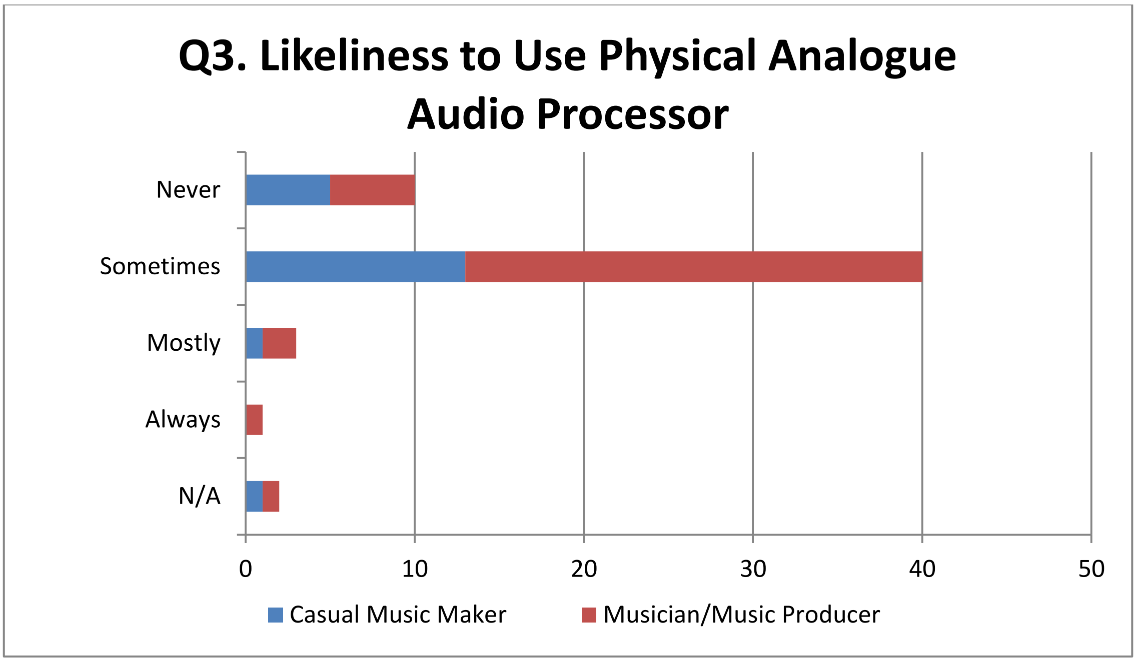
	Casual Music Maker	Musician/Music Producer	Total
Other		1 *(2%)*	1 *(2%)*
Don't Mix or Process Music		1 *(2%)*	1 *(2%)*
Personal PC with Software	16 *(28%)*	25 *(45%)*	41 *(73%)*
Home Studio w/ Hardware	5 *(9%)*	16 *(28%)*	21 *(37%)*
Professional Recording/Music Studio		15 *(27%)*	15 *(27%)*

Figure 6.2 Where and how respondents primarily mix music.

Respondents were allowed to select multiple answers regarding where and how they produce music. A majority of the respondents reported producing music within a personal space, with 37% reporting to use a home studio with processing hardware and 73% choosing to use a personal PC with software. While many respondents used both a mixture of hardware and software in their production workflows, nearly 3 quarters admitted to using software techniques while around one-third reported incorporating hardware.

Only 1 musician reported not mixing music which is principally due to them identifying as an instrument-playing musician instead of music producer. As a result, this respondent chose the ‘Non-applicable’ option for subsequent questions relating to their method of production. Additionally one Music producer selected “Other,” stating “mixing and mastering at home. Some tracking in the studio (Drums/Vocals).”

Q3. If You Mix Music, How Likely Are You to Use Physical Audio Processing Systems to Process Music in Comparison to Software Equivalents (e.g Hardware Compression vs. Software Compression)

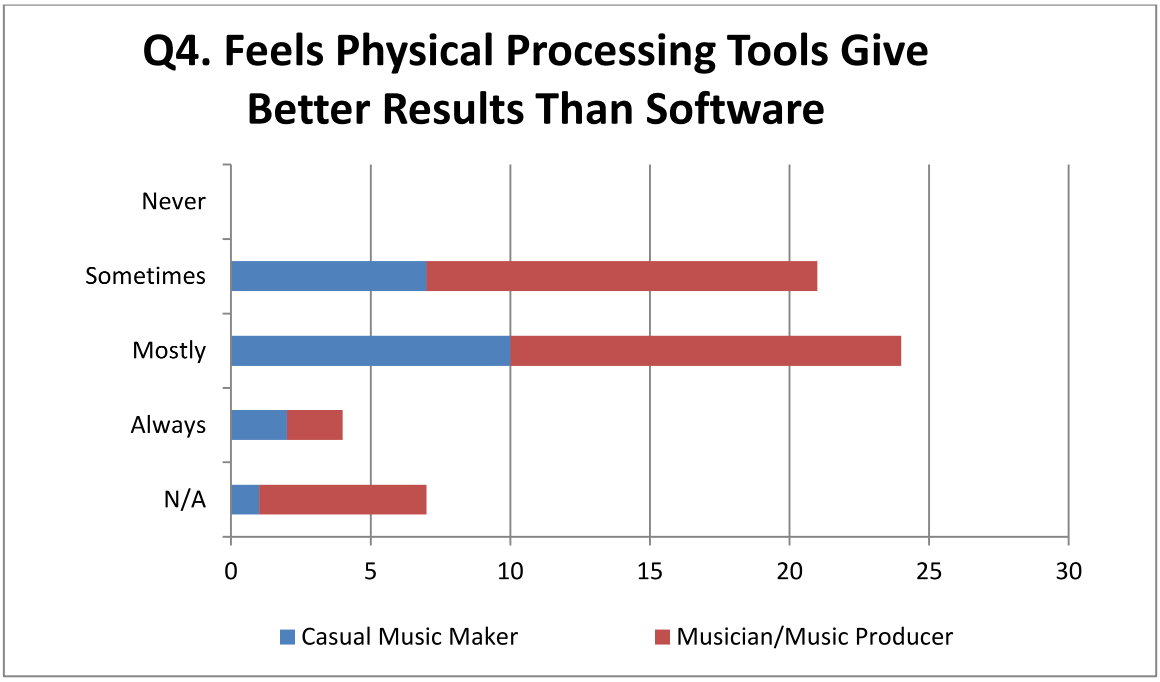


	Casual Music Maker	Musician/Music Producer	Total
Never	5 *(9%)*	5 *(9%)*	10 *(18%)*
Sometimes	13 *(23%)*	27 *(48%)*	40 *(71%)*
Mostly	1 *(2%)*	2 *(4%)*	3 *(5%)*
Always		1 (2%)	1 *(2%)*
N/A	1 (2%)	1 (2%)	2 *(4%)*

Figure 6.3 Participants' likeliness to use physical analogue processors to currently mix music.

78% of respondents reported using hardware in some capacity, with 71% acknowledging that they 'Sometimes' use it for production purposes. The ratio of respondents who used hardware was about 13% higher for music producers than casual musicians. Less than one-fifth of the overall respondents reported never using hardware at all.

Q4. Do You Feel That Physical Processing Hardware Give Better Results Than Their Software Equivalent



	Casual Music Maker	Musician/Music Producer	Total
Never			
Sometimes	7 *(12%)*	14 *(25%)*	21 *(37%)*
Mostly	10 *(18%)*	14 *(25%)*	24 *(43%)*
Always	2 *(4%)*	2 *(4%)*	4 *(7%)*
N/A	1 *(2%)*	6 *(11%)*	7 *(13%)*

Figure 6.4 Respondents' feelings of analogue processing tools providing better results than software equivalents.

87% of respondents felt there are times when musical hardware give better results than software and none of the respondents reported that hardware 'Never' give better results. The ratio of casual music makers who felt that hardware 'Mostly' give better results (50% of casual music makers) also showed to be higher than the music producers (39% of music producers). Some optional feedback regarding these choices fell into the following areas:

Analogue characteristics difficult to replicate

While software has made drastic improvements over the decades to accurately replicate software systems and in many cases have become a convenient and preferred tool over hardware, some felt that software cannot always replicate the subtleties of hardware, particularly when it comes to tonality and sound characteristics of rare and vintage devices. One respondent mentioned “software has caught up on the physical but it still can’t replace all the previous physical units, particularly with tone” (Music producer - ‘Mostly’). Additionally, another respondent added that:

> *The sound of some hardware cannot be easily replicated in the sterile ITB environment. An example of this is the sound of resampling multiple times through a hardware sampler (emu samplers in particular!).* (Casual music maker - ‘Sometimes’)

Appreciation for physical interfaces of hardware

For some respondents, there is simply an appreciation of having a “hands-on” experience with physical equipment that is not encountered with software. One respondent mentioned “I just personally prefer a more tactile approach to production” (Music producer - ‘Mostly’).

Appreciation of audio characteristics of hardware

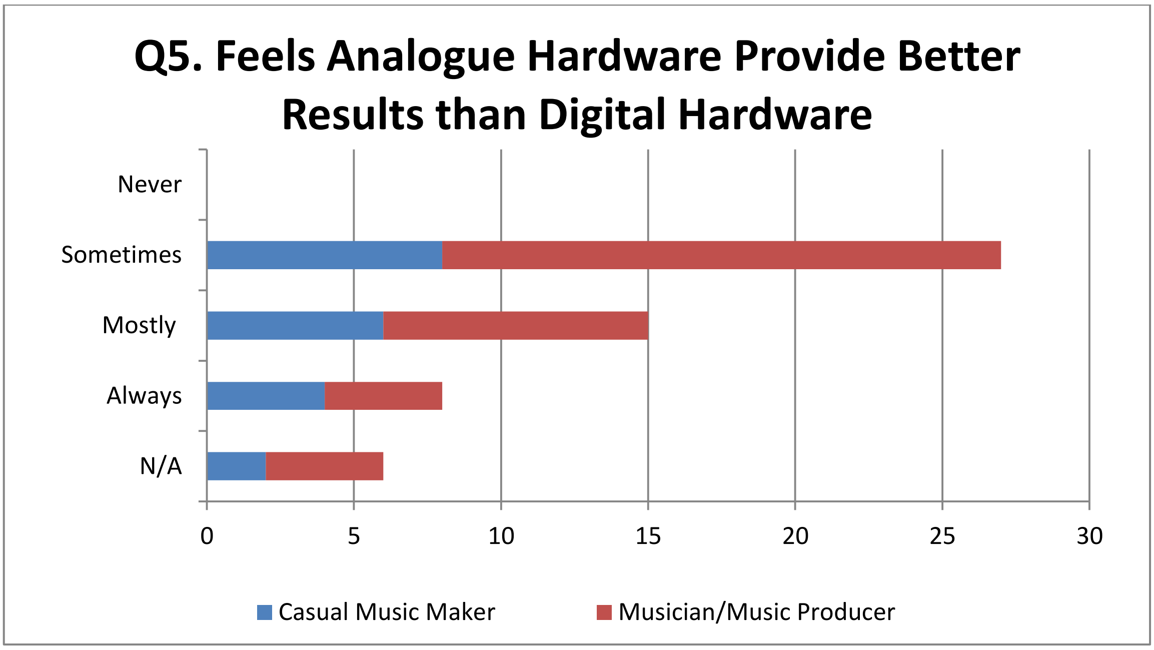
Aside from some respondents feeling software struggles to replicate hardware, others felt that hardware provides distinct benefits to software. In comparing hardware and software, one respondent mentioned “better is not quite the right word though. Different and mostly richer, along with direct creative interaction with the mix as it prints” (Music producer - ‘Mostly’). A common argument is that hardware produces more nature and organic sounds, as detailed by one respondent:

> *From my musical experience, physical processing tools have given a better result as they have a more organic sound, in comparison to their software equivalent which digitalises everything and makes it sound artificial, robotic from my point of view.* (Casual music maker - ‘Always’)

About 13% of the respondents answered ‘N/A,’ which typically meant they did not feel that one technique gives better results than the other overall. As one music producer reported:

> *Depends on the modelling? Some well-designed digital replicas not only sound great but often they can also extend the abilities of the original hardware (think extra M/S decoder modes, wet/dry balance controls on some compressors that didn't originally have them opening up parallel processing within the plug). Of course there are also lots of digital plug-ins that fall short of the mark and I am still yet to find any [software] that replicates the tactile control of hardware processors. So if it is tweakable synths your after you are not going to get the responsiveness of a Moog using any current MIDI controllers (even the high end NI stuff still isn't great).*

Q5. With Regards to Physical Hardware, Do You Feel Analogue Components Provide Better Results than Digital Counterparts?

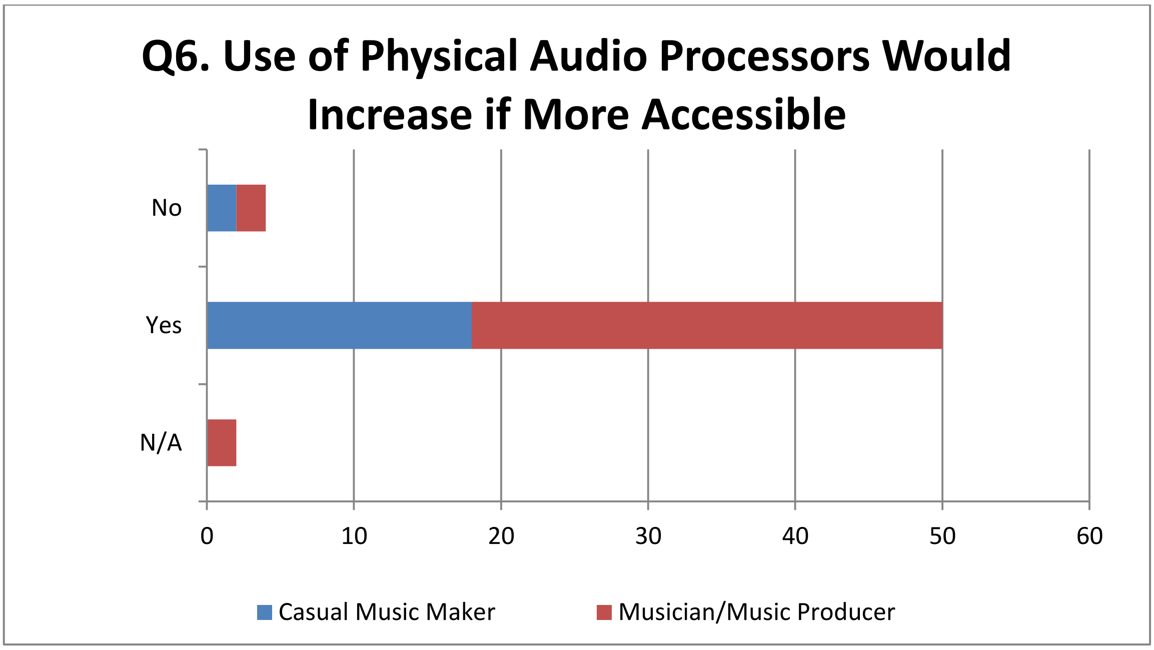


	Casual Music Maker	Musician/Music Producer	Total
Never			
Sometimes	8 *(14%)*	19 *(34%)*	27 *(48%)*
Mostly	6 *(11%)*	9 *(16%)*	15 *(27%)*
Always	4 *(7%)*	4 *(7%)*	8 *(14%)*
N/A	2 *(4%)*	4 *(7%)*	6 *(11%)*

Figure 6.5 Respondents' feelings of hardware processing tools providing better results than software.

Regarding physical hardware, 89% of respondents felt there were times when analogue hardware provided better results than digital hardware, and 11% reported having no preference. There were not any respondents who felt that analogue hardware 'Never' produce better results than digital hardware.

Q.6 Would Your Use of Physical Audio Processing Systems Increase If They Were More Accessible?

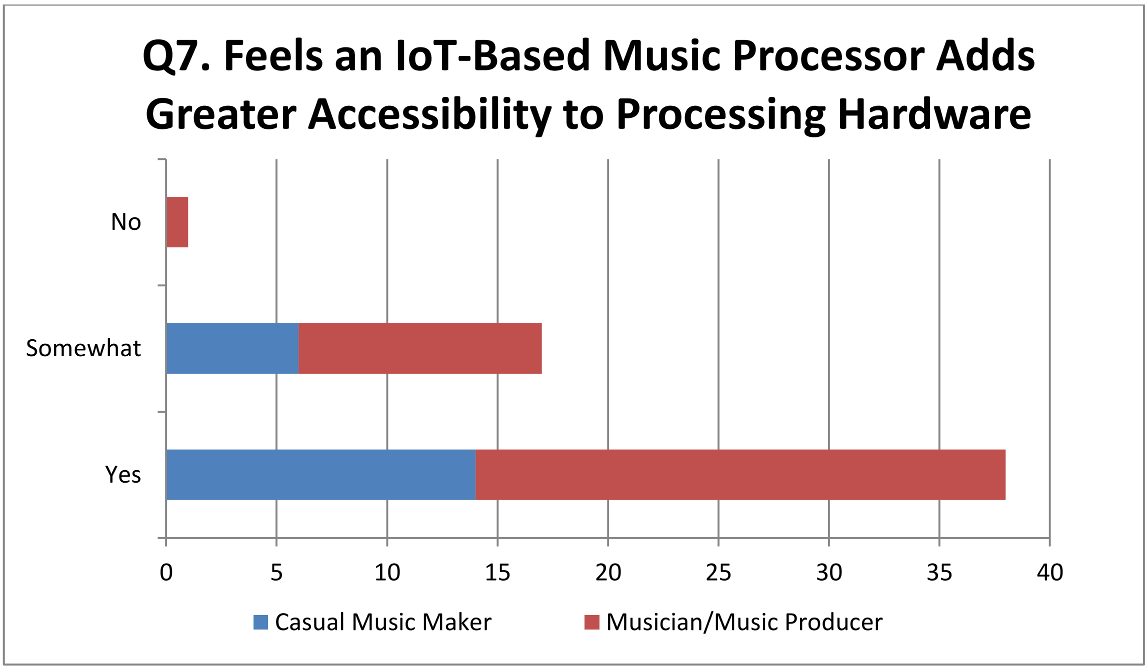


	Casual Music Maker	Musician/Music Producer	Total
No	2 *(4%)*	2 *(4%)*	4 *(7%)*
Yes	18 *(32%)*	32 *(57%)*	50 *(89%)*
N/A		2 *(4%)*	2 *(4%)*

Figure 6.6 Comparison of participants who would use analogue audio processing systems if they were more accessible.

89% of respondents reported that they would use more analogue audio processors if these devices were overall more accessible. Of the 4 ‘No’ responses, 3 of the respondents (2 casual music makers and 1 music producer) reported only using a personal PC with software to produce music and the last music producer felt that plug-ins have reached a level where they are comparable to hardware.

Q7. Do You Feel an IoT-Based Music Processing System Adds Greater Accessibility to Analogue or Professional Audio Processing Hardware



	Casual Music Maker	Musician/Music Producer	Total
No		1 *(2%)*	1 *(2%)*
Somewhat	6 *(11%)*	11 *(20%)*	17 *(30%)*
Yes	14 *(25%)*	24 *(43%)*	38 *(68%)*

Figure 6.7 Participants' feelings that IoT adds greater accessibility to analogue audio processing systems.

All respondents aside from one felt that an IoT-enabled music system adds some accessibility to music processing hardware. Close to one-third of the respondents felt that IoT 'Somewhat' adds more accessibility while around two-thirds reported 'Yes' it does add more accessibility. Similarly, this is mirrored in both categorical groups, where 30% of casual music makers and music producers reported 'Somewhat' and 68% of casual music makers and music producers reported 'Yes.'

Q8. Briefly Describe Your Impression of an IoT-Based Music Processing System:

Question 8 presented the first opportunity within the questionnaire to collect open-ended, qualitative feedback. An analysis of the responses provided 36 comments offering positive or supportive impressions of an IoT-enabled music system and 15 comments displaying mixed feelings or potential concerns of the technology.

A breakdown of response themes immerging from positive and supportive respondent impressions is shared below:

Overall/Generic Supportive Feedback

Many respondents provided general, supportive comments that were not directed towards any specific research question, expressing sentiments such as "very interesting technology with a lot of potential, provides a good utilization of the IoT for something productive and attractive to consumers" (Casual music maker) and "I think this is an incredible idea and I would definitely use it" (Musician producer). The responses typically showed general expressions of interest in an IoT-enabled music system, while some offered excitement about future possibilities, such as:

> *Amazing concept that with enough research and development could make a real difference to the world of music production.* (Casual music maker)

Accessibility/Engagement

A high number of comments relayed thoughts about IoT promoting greater accessibility and engagement into music processes. One music producer states:

> *Great way to access equipment around the world to develop personal work/projects. Otherwise without one may never get to experience certain types of physical equipment.*

Others expanded on opportunities IoT presented in eliminating barriers from access to particular equipment. A casual music maker stated:

> *Good step forward for the future of music as it takes away the location barrier when it comes to audio recording*.

Additionally a couple of respondents presented arguments of IoT extending musical processes to individuals with special needs, incurring some ***health benefits***. This is reflected in comments such as: “I think it can be useful and bring more option to people with limited capabilities” (Casual music maker), and “I see that it has potential for curious producers but most notably, people with disabilities” (Music producer).

Creative Benefits

A few respondents spoke on using IoT as a unique tool for adding creativity and creative techniques in music. One respondent reported:

> *It is a very good idea. I personally would use it more creating sounds instead of mixing/mastering.* (Music producer)

However, another respondent saw opportunities outside of standard music production, commenting on how IoT can be applied to other audio fields:

> *For film, to process audio to occur in a particular space and location of space to give a producer an idea of what to emulate with plug-ins. Perhaps, to run a guitar DI channel through a one of a kind guitar amp in studio conditions.* (Music producer)

Enterprise Benefit

Respondents relayed ways in which IoT can economically support individuals in their endeavours to create music, stating “fantastic idea and very useful to music producers with limited income or equipment” (Music producer) and “well, it looks to be making the expensive hardware accessible for less well-off people. I can get behind that.” (Music producer). These comments generally related to IoT granting opportunities for individuals with low incomes to access equipment that would normally be out of their financial price ranges.

Cultural Benefit

A last theme showing support of an IoT-enabled music system in the questionnaires was cultural benefits. Most notably this is associated with granting new opportunities for “bedroom musicians” to engage more with physical hardware. One music producer mentioned:

> *I believe that, if accomplished, it would be an amazing breakthrough as a business and as a product. The idea of being able to obtain all these hardware or physical sound obtained properties from my "bedroom" studio would be great. I happen to match the description of the bedroom studio and being able to use synthesizers and other tools would become an essential in my production.*

Additionally, it may provide new generation of musicians opportunities to expand their creativity, as presented by one music producer:

> *Fantastic. IoT Based processing can create a new platform for the new generation of producers to expand their creativity whether in a bedroom studio or professional studio. It will enable producers to find their own sound while using different responses in places such as frequency responses.*

While many positive themes emerged from the questionnaire data sets, several respondents presented feedback that showed mixed feelings or concerns arising from an IoT-enabled music system. The major thematic concerns are listed below:

Device Use Regulation

There was an overall concern about how the use of IoT-enabled music devices could be regulated if they were more accessible and how overbooking of a particular piece of equipment could become a constant issue. One casual music maker mentions:

> *I do wonder though about the demand there would be for the best gear, as it can only be controlled by one computer at a time, how would the time you would be allowed to use the gear in question be regulated?*

Similarly, there were concerns about wait time that could emerge to use ubiquitous audio hardware, such as:

> *So long as the hardware is widely accessible, i.e. not needing to wait in a queue of 1000 people to use a compressor unit, then I believe it could be very useful and increase the quality of sound.* (Casual music maker)

Internet Reliability

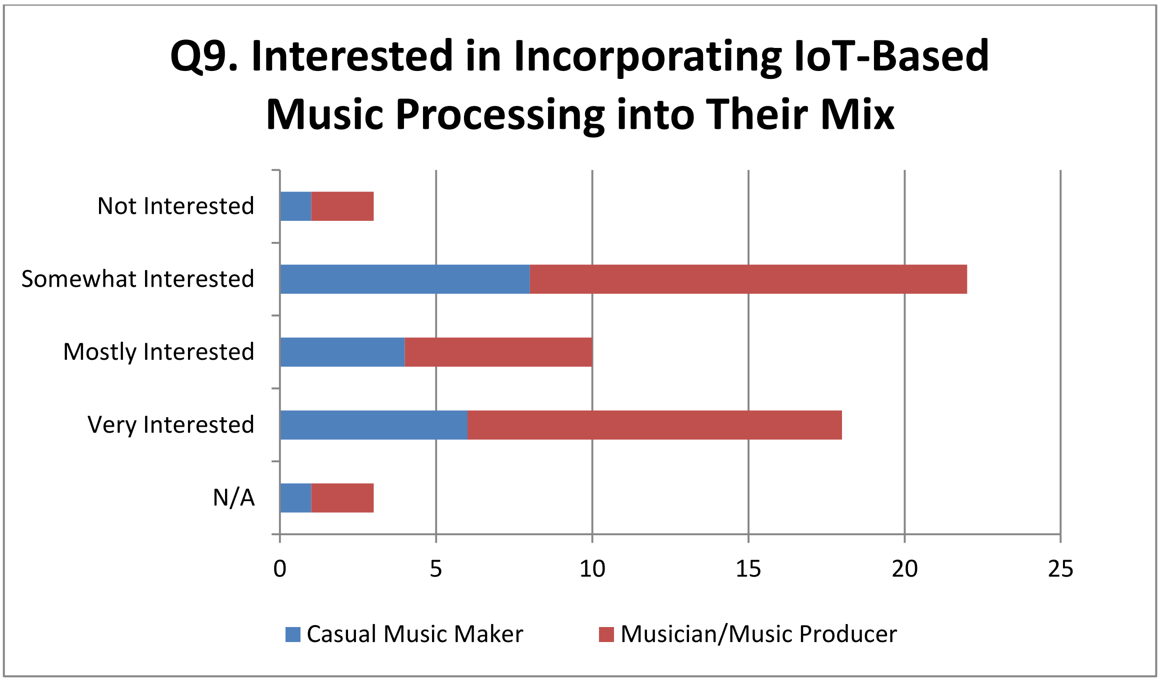
Another concern arising from IoT-enabled music systems is the reliance on a stable Internet connection. Some felt that unreliable connections could jeopardise the quality of the experience, offering statements such as “honestly would rather have the physical hardware in front of me because internet connection has to be good” (Casual music maker) and “awesome idea, but latency and encoding/decoding seem like big hurdles” (Music producer).

Quality of Experience

Associated with the concerns network stability, a few respondents felt that the reliance on the Internet could result in a loss of quality for the production experience, both regarding interaction and musical output. One music producer mentioned “Intriguing. Concerned on losing quality of sound” (Music producer), while another expanded further:

> *The idea conceptually is great and for static processes that don't get automated across the production process they could definitely have a place. My major concern would be the responsiveness and interface for using such devices to animate the sounds in real-time as per standard DAW automation techniques.* (Music producer)

Q9. If Available, How Interested Would You Be In Incorporating IoT-Based Music Processing Systems Into Your Own Music Workflows?



	Casual Music Maker	Musician/Music Producer	Total
Not Interested	1 *(2%)*	2 *(4%)*	3 *(5.5%)*
Somewhat Interested	8 *(14%)*	14 *(25%)*	22 *(39%)*
Mostly Interested	4 *(7%)*	6 *(11%)*	10 *(18%)*
Very Interested	6 *(11%)*	12 *(21%)*	18 *(32%)*
N/A	1 *(2%)*	2 *(4%)*	3 *(5.5%)*

Figure 6.8 Participants' interests in incorporating IoT-based hardware processors into their music making process.

89% of respondents showed interest in incorporating IoT-based hardware into their music workflows, with about 39% reporting 'Somewhat interested' and 50% reporting 'Mostly Interested' or 'Very Interested.' Most of the 'Not Interested' or 'N/A' choices arose from respondents who primarily produced music using a personal PC with software or respondents who did not see a major benefit from using hardware over software audio systems.

Q10. Are There Any Pros and Cons You Can Envision From IoT Extensions to Music and/or Other Creative Fields?

The final question presented another opportunity for open-ended feedback from respondents regarding pros and cons they envision about IoT-enabled music production, and also allowed them to extend their perspectives to other creative fields if desired. Similar to question 8, the responses can be broken down into themes that relate to either pros or cons associated with the technology.

The following are some themes drawn from responses sharing **PROS** of IoT music applications:

Greater Accessibility and Flexibility

A high number of responses were collected that highlighted an IoT system providing greater accessibility to music production equipment that individuals would rarely encounter otherwise. Sample responses included "great chance to network and use equipment you wouldn't normally have the chance to use" (Music producer) and "a pro for me would be that you don't necessarily need to own that exact device which would be great" (Casual music maker).

Again, there are also references to health benefits, as one respondent stated "help limited accessibility" (Casual music maker) as he later referred to the opportunity of IoT to assist individuals with physical impairments to produce music.

Enterprise Benefits

A number of respondents reflected enterprise benefits in terms of being able to access and afford professional quality production devices that are often considered expensive and unaffordable. Responses reflecting this included "cheaper alternative for musicians/creators to use gear they can't afford" (Music producer) and "wider accessibility to high end, less affordable hardware" (Casual music maker).

Cultural Benefits

As alluded in Question 8, respondents felt IoT applications in music production enabled predominately digital software users to also benefit from access to analogue and hardware tools. One respondent mentioned that an IoT-enabled music system "could help many digital only producers access hardware" (Casual music maker) while another felt these systems can allow people “access to the public to Pro-level kit and have access to the pro's niche level kit” (Casual music maker).

Going further, respondents felt IoT showed potential to help bridge the divide between digital and analogue music production. One respondent mentioned that “this is a very innovative way to bridge the gap between analog and digital when producing music” (Music producer), while another expanded:

> *The trend of the world is digitalization. And even though it can be sad to lose the beauty of the real world, the accessibility would give the average user would overcome any con.* (Music producer)

Educational Benefits

Educational benefits of IoT-enabled music applications presented an unexpected theme that became prevalent in both the questionnaires and the professional interviews. This is reflected in comments by the respondents concerning opportunities to demo and compare different devices that they do not personally own. Sample feedback includes “greater accessibility and an opportunity to compare otherwise unaccessible equipment” (Casual music maker) and “a pro is allowing bedroom productions to try out and access hardware they are unable to come across” (Music producer).

Creative Benefits

Some respondents shared insights into how IoT can bring new creative techniques to music and additionally areas expanded outside of music production. Some areas where respondents felt IoT benefits can influence creativity included:

- “Performance (Live), disability, robotics where new ways of signal transfers need testing” (Music producer)
- “Film - creating realistic reverb or echo controlled by a midi interface controlling all hardware in a studio” (Music producer)
- “New genres or mixing styles/producing styles/new trends of music” (Music producer)

New Workflows

A last theme reflecting pros of IoT music applications is the development of new workflows in music production. As shown statistically in Question 9, many respondents felt that IoT applications could be useful tools for influencing their current music production processes. One musician mentioned greater flexibility in producing music, stating:

> *Good flexibility for musicians. If you leave the studio but think you want to change something after you get home, providing you have a stable connection, you can access the physical equipment at any time.* (Musician producer)

While another reflected on greater options for production, stating:

> *The applications to post processing are possibly endless and the system is probably perfect for this purpose (i.e. using outboard compressors, reverb, etc.).* (Casual music maker)

In addition to pros of IoT-enabled music applications, many respondents shared feedback regarding concerns or perceived **CONS** of these systems if they became universally available. A selection of themes is presented below:

Reliability of Internet/Reliability on Performance of Real-time Interaction

Concerns about Internet stability, latency, and less than ideal, real-time interaction produced a number of comments regarding potential cons of an IoT-enabled music system. A sample response mentioned that "latency would be a big issue, especially with live recording and tracking, this may never be able to be solved, however…" (Casual music maker), while another respondent elaborated that:

> *Analogue gear has to work in real-time, there is no possibility of time or phase compensation at the equipment. This would need to be implemented at the DAW end if transfer speeds were slow enough to require it, and this would limit any real time interaction as the mix prints.* (Music producer)

Additionally, one respondent spoke on concerns for regions with less developed Internet structures, stating:

> *If one was to take it to the less fortunate countries who don't have direct access to internet or limited access to internet. Would be more difficult to reach those areas.* (Musician producer)

Limited Resources/Oversaturated Market Use of IoT Music Devices

Another major concern seen across the data is the limitation of available IoT-enabled music hardware devices due to oversaturation of demand if the devices became widely available to the public. One respondent asks:

> *I do wonder though about the demand there would be for the best gear, as it can only be controlled by one computer at a time, how would the time you would be allowed to use the gear in question be regulated?* (Casual music maker)

While another reflects:

> *The problems I see are (i) that only one user can access any one piece of equipment at any particular time, thus limiting the number of users to number of physical units...* (Music producer)

Unintended Consequences

Additionally, while some insights aren't repeated throughout the data, there are revelations how the positive aspects of IoT music applications can promote some unforeseen negatives. One being greater isolation of some music makers, as one music producer states:

> *Like many internet based innovations, this has the power to improve and widen the creative musical horizons for many people, yet it also has the potential to isolate people as it reduces the need to physically move around. As long as this is recognised and treated accordingly, then this innovation will do much good in the music industry.*

Another music producer speaks on possible learning curves adopting the technology, mentioning

> *An answer to one problem usually bring its own questions demanding answers. It can save time meanwhile creating more options and versatility which in itself demands knowledge and technique, which requires more time invested to become fully utilizable.*

Lastly, one music producer introduces the idea that IoT-enabled music devices could disrupt the software plug-in market, or conversely negatively impact the demand for new hardware devices. In a unique perspective, the respondent states

If this truly did take off, there is a possibility of it killing the physical hardware market. However, you would also be suffering with the issue of direct comparison to plug-ins and the utility of using these "true physical" systems through digital means kind of screws over both sides.

6.2 Interview Evaluations

6.2.1 Design of Interviews

The interviews targeted a smaller subset of individuals who have professional expertise in audio and music production or sound engineering. The list of interviewees is as followed:

1. Todd Reitzell – Musician and sound engineer. Founder and managing director of an international creative agency.
2. Dr. Bill Campbell – University lecturer, sound engineer, and audio producer for music, film, and gaming.
3. Dan Wilde – UK singer, songwriter, and music producer.
4. Mat Skidmore – Freelance live recording and mix engineer.
5. Simon Gogerly – University lecturer, Grammy award winner, and leading UK music producer and mix engineer.
6. Alan Branch – Grammy award winning audio engineer, producer, musician, and writer.
7. Gary Bromham – Professional music producer, mix engineer, guitarist, and songwriter.

The interviewees are esteemed by their peers and have made significant contributions to music through their years of experience in the field, having earned accolades and awards in music and sound production as a result of their outputs and actively sharing their knowledge with audio and music students. The selected sample group was ideal based on their accessibility, experience, and proximity to the researcher, however the homogeneity of the group reflects the overarching demographic of the music industry, a topic further exemplified by research from Smith, Choueiti, and Pieper (2019).

The interviews are qualitative in nature, and provide more subjective views regarding the incorporation of IoT for music production. As identified by Goddard III and Villanova (2006), the interview questions provide the following benefits:

- "Allowing respondents to reveal otherwise concealed attitudes,"
- "Revealing problems and their potential solutions through discussion,"

- “Encouraging free expression,”
- “Discovery of personal information, attitudes, beliefs, and perceptions that a paper-and-pencil survey might not uncover,”
- “Allowing interviewers to probe or follow up on survey items.”

The interviews are semi-structured; an outline of the questions was arranged prior to the interviews in order to set a plan of discussion, but the responses and feedback of the interviewees help assist driving further discussion (Stuckey, 2013). The benefits of semi-structured interviews are that they give more freedom to the interviewee to express their views (Stuckey, 2013) and more opportunities to the interviewer to “prompt and probe deeper into the given situation” (Kajornboon, 2005) without being tied to a rigid format mandated by structured interviews. The developed interview questions are framed as open-ended variations of the questionnaire questions and aim at providing additional insight into needs and gaps within music production that a possible IoT solution can serve. The questions can be found in Appendix FF.

6.2.2 Thematic Analysis of Interview Data

The analysis of the interview responses involves scanning the transcripts and observing consistent topics or themes that are prevalent throughout each of the discussions. The themes “capture something important about the data in relation to the research question, and [represent] some level of patterned response or meaning within the data set” (Braun and Clark, 2006, p.82). As discussed in the methodology, theme categories were driven directly by the second and third research questions:

RQ2: How can IoT-enabled music systems facilitate new music production engagement, workflows, and collaboration methods?

RQ3: What cultural, enterprise, and creative benefits do IoT- based music systems present?

The list of prevalent themes in the interviews is numbered down below:

1. Predominant production environment
2. Preferences for analogue and physical hardware over digital software
3. Preferences for digital software in production.
4. Engagement with IoT-enabled music hardware
5. Creative benefits of IoT music application
6. Cultural benefits of IoT music applications

7. Enterprise benefits of IoT music applications
8. Educational Benefits of IoT music applications
9. Concerns and risks associated with IoT music applications

6.2.3 Interview Analysis

As discussed in 6.2.1, the interview questions mirror those of the questionnaires, therefore the first 6 questions collect information regarding the interviewees' backgrounds and current preferences for music production techniques, and the last 4 questions gather insights into the interviewees' views on IoT being adopted to facilitate music production. In this section the themes will be presented as headers with relevant commentary from the interview respondents supporting the discussion of the themes.

Predominant Production Environment

The first two questions were posed in order to acquire an understanding of the production backgrounds of the interviewees, specifically learning about their experiences relating to audio and music processing and the typical spaces in which they conduct their work. The questions immediately revealed that while all of the interviewees spend some time working inside of a professional studio space, a majority of their time is dedicated to a personal or in-home developed studio. For some, the personal space offers the best accommodation for the initial stages of production before moving into a professional studio that has a greater range of resources to finalise a mix. Campbell reflects that:

> *So for convenience I'll work in-the-box at home but I obviously don't have the facilities to really get a good mix, so I would go into the studio to finalise in a proper environment and then use hardware to supplement it.*

Respondents reported that the personal space, although not always ideal, offers comforts of working at a desired pace without the sometimes harsher time constraints of professional spaces. Additionally the personal space provides economic advantages as it is not as costly as professional spaces, and those savings can be passed onto the customer as a result. Wilde elaborates on this point, stating:

> *It's not the ideal space but most of what I do is from home because it's… there are no overheads there and I'm trying to provide a service to people that, you know, doesn't cost them the world.*

A final point highlighted the fact that when a producer reaches a higher professional status within the music industry, their clientele almost expects them to have a personal working environment available in some capacity. Gogerly adds “the way that the music business has changed over the years clients now expect you to have your own studio to mix in” and Branch supports his sentiment stating, “I think sometimes we need spaces in studios just more for business and personnel and everything else.”

Preferences for Analogue and Physical Hardware over Digital Software

Questions three through five asked whether the interviewees use physical hardware, digital software, or a combination of the two in their normal production processes and probed deeper into why these were the cases. The interviews offered a range of responses with insights into the appeal of both hardware and software techniques for music processing. While mixing audio and music primarily tended to be in-the-box amongst the interviewees, they opened up about the appeal of analogue and physical hardware and why some will never tend to completely discard analogue techniques. The lure for hardware can be categorised into 2 areas: *Quality & Performance* of the technology and *User Experience*.

Quality and Performance

One argument regarding the performance of physical and analogue devices shared by both interview and questionnaire respondents is that there are qualities about physical music hardware that separate them from digital software, and that some devices provide greater warmth and sonic depth than their software counterparts. Bromham believes that analogue compression, saturation, and distortion “provide excellent coloration solutions for digital audio which can sometimes be 2 dimensional and slightly sterile.” With regards to specific types of hardware like reverb, he feels “the sound is more encompassing and immersive than a plugin which often sounds like it is sitting on top of the mix!”

Similarly, another common view shared among the respondents is that there are attributes that software replicas may not be able to accurately recreate about hardware. Skidmore remarks, “yeah there definitely seems to be something happening in that analogue box that you don’t get [in software],” while Branch and Bromham give definitive insights emphasising the appreciation of the ‘non-linear’ properties of analogue equipment. Branch stresses this in his statement:

> *But some stuff, you know synthesisers or certain valve compressors that have got kind of non-linearities or certain things about it, there are times when you plug in a real hardware piece of gear and you just know that it sounds real.*

In fact, one of the appreciations of hardware is that different devices can give unique, unpredictable results every time, and some producers believe this characteristic can positively influence the aesthetic of a recording. Gogerly provides a more in-depth view of this phenomenon, stating that:

> *But to a degree I almost sort of like the slight unpredictability of real hardware, there is something about, it's like playing an instrument. The difference between playing a real instrument and playing a plug-in instrument is that the real instrument will react differently and sound differently depending on what you're playing whereas the plug-in, once you record your midi performance, will play back exactly the same every time.*

While most of the interviewees were satisfied finalising mixes in software, including implementing standard touches such as equalisation, a common preference among the respondents was capturing or recording the best quality audio using desired hardware before bringing the recording into the mix stage.

User Experience

In addition to feeling that they sometimes obtain a more pleasurable auditory experience from physical hardware, another draw to hardware among the interview respondents is the tactile experience. Both Campbell and Gogerly reflect on instances where a producer already knows the workings of specific devices, with Campbell stating, "rather than spending hours trolling through settings on a plug-in, I can go straight to a piece of hardware that I know works and sounds good. It's quicker in some respects than doing everything in-the-box I find."

The process of manually trying to restore a desired setting on a hardware device is arguably a negative trait of hardware compared to software, but there are occasions when the tactile experience of rotating a knob or moving a fader is preferred than digitally interacting with a software interface or plug-in. Skidmore admits that "there's something kind of nice and tactile about actually using hardware equipment," and sometimes it's even the nostalgic factor of engaging a process in a traditional manner, as explained by Wilde:

> *Now it could just be that, you know, that could just be sort of in my head because I'm twisting some knobs on a hardware thing and I'm thinking 'yeah, I'm like a proper producer now' not clicking around with a mouse...*

Lastly, in many cases producers have a greater appreciation of hardware devices they discover and take the time to become familiar with and learn the inner-workings about. The physicality of the gear and personal interaction between the person and device can build an emotional response. Branch reflects more by stating:

> *I sometimes think there's a point where you just fall in love with that piece of hardware. You know it, you trust it, that kind of thing, so there's that element of just knowing an individual piece and no other software can come close.*

Preferences for digital software in production

On the opposite spectrum, all of the interviewees admitted to using in-the-box software primarily for mixing audio, attributing the key factor to convenience. The ability to produce music in a personal space at one's convenience is a modern development that offers a number of advantages to traditional, studio-based production schemes. Specifically two factors drew a lot of appeal for the use of software in both the questionnaires and the interviews: *recallability* and *cost savings*.

Recallability

The biggest theme regarding the advantage of software in music production amongst the interview respondents is the option to recall presets. With software, a producer can save the settings of their work and return to a production at a later time, restoring the exact settings without losing their initial work. Reitzell remarks on the value of this, mentioning:

> *I'm someone who I like to be able to make some settings and load those settings and you know, recall it. Rather than getting frustrated never being able to dial in that sound again.*

Particularly with mixing, where all of the respondents reported opting for software over hardware, the chance to save and recall work is of significant importance. Skidmore expresses this point saying "for me, the most important thing with mixing has become recallability."

Cost Savings

The convenience and efficiency of digital mixing in a personal space inherently comes with additional financial savings. In most cases, a software plug-in or digital model of a well-known music processing device will cost significantly lesser than the actual hardware product. Reitzell expands

> *But all of this hassle comes with digital, comes at a price. It's less expensive, typically less expensive. I mean rather than spending six or eight-hundred or a thousand currency units on a great compressor or EQ you can buy the equivalent at 150 currency units, and that's what I do!*

Additionally, the savings of not having to purchase expensive equipment or rent extra time in a professional studio can be passed onto the customer. Skidmore recounts his business model of splitting his production process between a studio and his own workspace because it results in greater savings for his clients:

> *Because of the nature I think of the industry now a days, what tends to happens is… so I'll work with a band, we'll record in the studio for a period of days, and then I tend to mix on my own. And so rather than sort of having the recording session and then the mix session together and at the studio… just 'cause it will save money for the band so we don't have to hire a studio space.*

Engagement with IoT-enabled music hardware

When asked whether their use of physical and analogue equipment would increase if they were more accessible, all but one respondent confirmed that they would add IoT-enabled hardware to their production workflows. Reitzell felt that IoT would not impact him much for his standard process of mixing for film, but most others felt it would be an added benefit for their production techniques. Gogerly adds his thoughts stating:

> *I would definitely embrace that in a big way I think because it's almost like having an extended studio where you can access pieces of equipment that you don't have or you don't own and you can't afford to buy necessarily or just don't have physical access to like plates for example.*

Similarly, Bromham reflects on IoT-enabled music systems being an extension to the process of in-studio equipment hires:

> *I am amazed that no one has done this before as I think it's an excellent idea. When I started producing music, it was common to hire in equipment on sessions and I see this as an extension of this principle. It means that someone that would*

not otherwise have access to a well-known piece of equipment, due to cost or size considerations, might subsequently have access to this world!

Additionally, while Wilde felt that IoT-enabled devices would not largely influence his processes of mixing in-the-box since he felt some processes like equalisation is more convenient in software, he reported desired uses for tracking and recording:

So I think tracking, like on the way in, recording… If it was I'd use it on the way in, tracking, and I might use it for mix bus and if ever I do like masters for people then I'd use it.

Lastly, most of the respondents relayed specific interests in utilising IoT technology for unique and untraditional processes. A popular topic included creating real-time reverb from acoustic spaces, such as Reitzell mentioning how he would value sending organ sounds into a chapel or cathedral, admiring this kind of "location independent processing." IoT may also provide unconventional ways of incorporating instruments or older sound producing devices into a mix. Skidmore reflects more on this:

I like the idea of instruments. So say, maybe things like synthesisers or that kind of thing is quite interesting where it's something which, isn't widely available or isn't widely recreated as a VST synth or something. Those kind of things could be good.

Creative benefits of IoT music applications

Expanding upon using IoT for non-traditional music practices, most of the respondents showed excitement for creative benefits that could emerge from IoT-enabled music systems. Skidmore felt that:

Anything that can give access to more kinds of equipment and sounds is useful for creativity… especially if you have a specific thing that you think you're after but you don't have access to it, those kinds of times it could be really effective.

And as mentioned, almost all of the respondents found the possibility of incorporating unique acoustic spaces into a real-time reverb scenario a very attractive creative application. Campbell reflects on some famous tracks being recorded by popular musicians in unique venues, and how the musicians effectively "played the space." The space became as much a product of the recorded sounds as the vocals and instruments, and ultimately added to the overall listening experience. Campbell expands:

> *So that speaking on a live scenario, where you probably wouldn't get that same effect… you wouldn't get the musician playing in exactly the same way if you're doing it through convolution. So in that sense, if you would have the real-time Internet of Things where you would actually have a drummer playing and hearing themselves playing in an amazing space that would probably be something that would be spectacular in itself.*

Additionally, a unique benefit of IoT is that it can promote innovation amongst its user and encourage originality, and as seen by many technical developments in the recent years, this is highly desirable for individuals in creative fields. Branch shares his impression:

> *When something actually, let's say, something original comes along, it's something different I can't get currently that gives me a different sound or, not saying better sound, but it gives me a different sound, something I can't achieve now, that's great because now you're searching for originality and something new. I can plug in, you know, to something and use new plug-ins that can enhance my song, I want to see it.*

Cultural benefits of IoT music applications

A key benefit of IoT-enabled music applications mentioned in the both the interviews and the questionnaires is the expansion of production processes by extending networked analogue and physical hardware to musicians who currently have little or no access to these devices. The questionnaires gave insight into how bedroom musicians can benefit from accessing devices outside of the limits of their personal spaces, but Wilde also presents a case for the nostalgia of analogue technology and how IoT can allow people who have had to shift to digital music production techniques over time reconnect and re-engage with older, hardware-based processes that may still hold sentimental value. He also contends a notion of nostalgia for things we haven't experienced, and how younger music producers can still romanticise about vintage hardware, similar to the growing appreciation seen for vinyl records among modern-day music listeners who may not have grown up with the technology. He closes with a personal relation, "…'cause people have nostalgia for things that they've never experienced too, don't they?"

Gogerly and Branch also refer to IoT-enabled music opportunities being attractive to people with more technical leaning backgrounds because it allows more experimentation with music processes while also granting options to understand and utilise an emerging technology trend. The idea of using a browser to interact with an infrastructure of interconnected devices could pose an exciting prospect for musicians who like to experiment as well as beta test new processes.

Finally, creativity inspired by IoT music applications can encourage growth in cross-cultural collaborations. Branch passionately explains:

> *I think the beautiful thing of the Internet of Things in a creative context is creativity or let's say musical art; I think in its general sense brings people together. It brings beauty and passion and all those exciting things as us as music makers kind of fills their inner beings.*

IoT offers different societies opportunities to become closer by sharing and combining their artistic techniques in addition to spreading a greater appreciation of their individual heritages. Branch closes his original point stating:

> *…So I think maybe the Internet of Things, if it enables creativity in art to become close together around the world maybe there's a broader sense of things, a beauty of joining, of becoming different cultures and different societies, an appreciation amongst each other rather than say the music scene in LA and music scene in London… I still think cultural communities would have that, music would have its kind of heritage I suppose but maybe we could create more together because of that.*

Enterprise benefits of IoT music applications

Another key benefit across the interviews was enterprise benefits of IoT. Bromham mentioned that:

> *The concept and process almost certainly have a place in post-production and mastering where select pieces of hardware might easily cost far more than those used in traditional recording and mixing environments.*

Relatedly, one argument is that IoT can allow production studios as well as individuals who collect musical hardware opportunities to monetise their equipment. Wilde remarks, "I think there are people that have amassed and collected all this nice gear and they can't be using it 24 hours a day and I'm sure they wouldn't want to be," suggesting that IoT technology incorporated into their collection of devices would allow individuals to set a scheme to rent and hire their unused equipment to the public. This could potentially create a new source of income for people, especially those who collect rare and highly-coveted vintage gear, while allowing customers more opportunities to find devices they desire. Wilde follows up to his argument stating:

> *… I think everybody even if you really enjoy your job, most people would say if they could work a little bit less and still make money from their assets or whatever, they would do that. So I think, yeah, there would be a lot of people who would say, right ok this allows me to do a little bit less production and just have these hours where I rent out my equipment.*

Similarly, Gogerly envisions an infrastructure where people can access a data base of networked music devices and see which particular devices are available and who and when they can be hired from. Gogerly elaborates:

> *But on a global scale I supposed if you have some kind of directory of users set up you could say… say you have a list of 100 people who've got Culture Vultures and you can see that 30 of them at that time weren't using them you could use one, especially if they were probably in a completely different time zone because they may be in bed at that time when you were doing your mix and they can leave all their stuff set up so other people could use it in the middle of the night which would be really cool.*

Additionally, another enterprise benefit of IoT includes the actual preservation of hardware devices themselves. If very expensive devices were set up to only allow digital access without physical human interaction it can save them from a lot of the standard wear-and-tear associated with typical use and misuse, inadvertently resulting in more savings by not having to spend additional funds on repairs and replacements. Reitzell expands:

> *So you spend thousands and thousands on a mixing desk and as soon as someone gets their little greasy fingers on it we start beating up. So if it could be completely robot controlled that would mean that it would take less abuse and might last longer.*

Lastly, associated with the robotising of hardware, IoT can help implement new workflows in production spaces. Reitzell explains a possibility of optimising a recording session explaining:

> *I'm an engineer in a control room recording someone and instead of going through 4 sound proof doors to get to a knob that's in a room with the performer, I can do it using IoT. This is a massive time saving for everyone, time saver for the engineer.*

It doesn't disturb the performer and you can more quickly, well with less disruption adjust a parameter that might be causing, I dunno, possible ear damage or possible equipment damage.

Wilde adds to this point mentioning how many digital processes can be controlled by a networked device like an iPad, "but for all the physical stuff I'm using, like the analogue stuff, there's nothing, I can't change it. So that would be quite useful, even to control your own gear." IoT can provide digital benefits of network control to analogue technology, influence new methods of engagement and interaction with hardware-based production practices.

Educational Benefits of IoT music applications

A last IoT-enabled music benefit that emerged across the interview data sets were educational benefits. Particularly among the respondents in academia, some interviewees envisioned how IoT can allow testing to occur for research purposes or be used as an instructional tool for students.

In one case, Campbell identifies how IoT music devices can be used as a comparative tool for evaluating different hardware devices and comparing them to equivalent software applications and plug-ins. He related to his postgraduate research experience stating:

If the Internet of Things would have been available to me it would have been nice to do real-time dynamic range compression with people sitting in the room, but you could compare digital to analogue or something like that, you know.

In this case, IoT would allow real-time listening tests to be conducted with actual hardware in remote locations with live participants. He expands:

…if you had like an Internet of Things where you had mass availability to these things you could actually get people from all over to be able to do these experiments for you or this research to test their abilities.

In a second case, Skidmore relates to IoT-enabled music devices being complementary tools for trainings and tutorials on the use of real musical devices. He explains:

Particularly I think when you're learning and you're sort of watching videos on YouTube of people with this actual hardware equipment using these things it would be interesting to go, 'ok well how when they turn that knob up to 4 on the

> *gate how does it differ to my plug-in?' And actually being able to use a real bit of hardware and actually hear the differences I think that would be interesting.*

But going further, this case can be expanded to lessons and courses within academic institutions. While currently working with students in different modules for an audio and music technology programme, Skidmore adds:

> *…and in fact from a university kind of perspective being able to actually show students an actual piece of equipment and actually run audio through it… but obviously without the university having to own it I think that could be very useful.*

Concerns and risks associated with IoT music applications

The Interviews presented many themes recognising areas where IoT can positively impact music production. However, addressing pros and cons, the data presented areas where concerns arose from the respondents regarding a widespread implementation of this musical processes.

Similar to the questionnaire data, some of the respondents felt that unlimited access to ubiquitous music hardware could create higher demands, consequently lower the supplies of available devices. In some cases, devices that were normally free and accessible to specific users may become harder to access as a result. Campbell presents the case:

> *I also think that you would have… probably if you had like one facility that had a Pultec available, it would probably be booked up all the time that you couldn't get access to it when you wanted to have access to it.*

These restrictions could affect the standard operations of local music studios. Musicians who frequented a nearby studio expecting regular access to desired equipment may have to be set aside for the growing demands of world-wide consumers who also have the desire to access the same hardware. Wilde expands:

> *It would be interesting to see how that affected studio availability I suppose. Like for somebody that actually wants to go and hire out the studio, you know, the studio might be like, 'Well sorry we're booked up because now, we're no longer serving the people of this area, it's the whole world.'*

Additionally the entire process of regulating the use of the networked devices would need to be amended. If these devices became high in demand there would have to be policies in place governing appropriate use in order to allow everyone reasonable access. A proper computing infrastructure would need to be in place to monitor use and only allow or restrict access based upon the agreements with the studio.

A last source of concern is that a growing access to musical devices could actually devalue music production and the creative process. Branch presents a unique argument:

> *I can certainly see patterns emerging, cons, like the easier we make it to make music… there is a plethora of terrible music you know. It's harder for very good music to rise to the top, so is there a con if we make it simpler to do things with the Internet of Things?*

The opportunity to have a wide arrange of hardware in addition to software readily available with little effort can actually stunt the creative process and allow more production of lower quality music by less talented producers. Similar implications have occurred with other contemporary music processes emerging in recent times, as explained by Gogerly:

> *The only downsides I can see to that is the aspect of, when you know software instruments came in and a lot of plug-ins came in… this even happened before [with] hardware, you know rack mount equipment where you'd get hundreds of programs in them and you can spend so much time just literally going through listening to presets that you end up losing the plot entirely about what you were trying to do creatively.*

However, Branch feels that although these processes have the opportunity to be abused and produce poor results, they ultimately stem from a place of creativity and human prowess, and as long as people continue to construct good ideas and remain proactive, creativity will continue to emerge as a result. He offers a conclusive argument:

> *I can only think if there is some type of process that took creativity away from us it still has to have good ideas, it still has to have our human intellect creating something new, fresh, and original with that, whatever it is… if it crosses over into the side it's just doing stuff for us, we're just watching it happen aren't we, we're not creating it anymore.*

7. Critical Assessment of Research

Chapter 7 provides a critical assessment of the analysis conducted in the research, giving further insight and context to the practical creative work, the evaluation of technology through the audio streaming tests, and the feedback considerations from music producers regarding opportunities and challenges of IoT-enabled music systems.

7.1 Realising an IoT-enabled Music System

This research set out to understand, demonstrate, and evaluate how the Internet of Things could impact new music development processes by allowing remote interaction with networked physical and analogue music technology. As part of the research, a proof-of-concept, IoT-enabled music application was conceptualised and prototyped to facilitate the investigations and address the following research questions:

RQ1: What are the current capabilities of IoT infrastructures to support distributed audio system networks, and what improvements can be identified and evaluated?

RQ2: How can IoT-enabled music systems facilitate new music production engagement, workflows, and collaboration methods?

RQ3: What cultural, enterprise, and creative benefits do IoT- based music systems present?

Two areas were determined to be beneficial components to the practical development of the IoT-enabled music system: the ability to virtually control physical music hardware through the Internet and the discovery of tools to deliver uncompressed, low-latency audio to these networked devices. The convergence of these two components in the design and build stage successfully aided in realising a unique system for Internet-driven music production.

The design, build, and implementation of the IoT-enabled music system took place over a two-and-a-half year span, starting with the creation of a webpage interface to send data between remotely connected computers and concluding in a complete user experience allowing virtual control of two audio hardware devices through the Internet. Using the practice-driven methodology detailed in Patterson et. al. (2015), the practical build of the system occurred over multiple iterative stages of research and design, and the progression of each stage built upon the successful employment of the prior. The iterative design and build cycle gave greater insight into existing technology that could effectively enable desirable processes at each developmental stage, and additionally identified some areas (such as network speed and latency, responsiveness, security) where improvements can occur. Continuous advancements in both existing technology and the Internet will make future IoT music systems ever more capable, but this research proved that present-day IoT resources can facilitate remote music production techniques and provokes the argument that wide-spread implementation of these systems may one day occur.

Near the conclusion of this research a new company, mix:analog, completed and released the most comprehensive realisation of an IoT-enabled music system that allows control of bespoke analogue audio processors and real-time VU and audio monitoring through their consolidated web application. The successful implementation of mix:analog's application further highlights the importance and necessity of networked audio systems for remote music production, and presents growing possibilities of how combined efforts between musicians and engineers can lead to more robust and effective IoT-enabled music soultions. Where De Carvalho (2012) and Dixon (2016) see digital software tools effectively reducing attributes of the music studio to a laptop, IoT can additionally help virtually extend professional studios devices to these laptops and personal spaces. This research has also shown that these types of networked architectures can help incorporate unconventional creative qualities, such as remotely-connected material environments that can produce distributed, natural real-time reverb, into standard music production practices.

7.1.1 Democratising IoT-based Development with Low-Cost Computing Resources and Open Source Code

Research Question 1 inquires about IoT infrastructures that can support IoT-enabled music applications, and this research notably illustrated that these systems can be realised with publicly available tools and non-proprietary resources. Having flexibility in product development methods inherently gives more creative input to individuals and helps expand technical design without a sole reliance on traditional, commercial-driven applications and corporate-based platforms.

Websockets played an important role in delivering real-time control information to networked computing devices, and its HTML5 skeletal code made it convenient to adopt a portable, web-based interface. Microcontrollers continue to become cheaper and increasingly more powerful, and allowed a wide array of computing processes to be performed to manipulate remote music processing devices in fractions of time. WebRTC provided flexibility into how media elements can be programmed and merged into the practical design to enrich the user experience, and JackTrip offered a high-performing audio streaming application that is freely accessible and user friendly. All of these components worked together to create a unified music experience without the need for a full assortment of assets typically exclusive to corporate environments. Consequently, this shows that other complex creative systems, not limited to music, can be fostered and delivered by broader groups of technically capable individuals, helping open new processes and opportunities for creative IoT development.

7.2 Critical Analysis of Research Evaluations

7.2.1 Discussion of Real-time Audio Streaming Evaluations

Further focus on Research Question 1 also showed two audio streaming platforms, JackTrip and WebRTC, offering prospects of high-quality audio transmission with low-latency speeds over the Internet. After extensive testing and evaluation of the two platforms, it was observed that they catered to slightly differing audio scenarios. The audio streams transmitted using JackTrip consistently maintained the entire frequency spectrum of the source audio and proved suitable for transferring full-range music and instrument sounds over the Internet. In contrast, the WebRTC recordings were found to show some filtering and compression of the original source signal. As discussed in Chapter 5, WebRTC employs techniques to preserve and enhance the human voice, thus making it more tailored for virtual chat applications and browser based video conferencing. Concerning a networked music production scenario, JackTrip provided the most desirable results, and this was reflected in the mixed-methods analysis of Chapter 5 where JackTrip showed lower measurements of distortion, dropouts, and latency compared to WebRTC in repeated trials of audio streaming tests. These quantitative measurements were complemented by general listener feedback from participants who subjectively compared the similarities of the recorded audio streams from the two platforms to their source audio files. JackTrip was highly favoured among the respondents, with many perceiving minor to no differences between the source audio file and the JackTrip stream recording.

After direct comparisons of JackTrip and WebRTC over the local area network showed that JackTrip objectively outperformed WebRTC in cases of real-time music streaming scenarios suitable for this research, the next step aimed at determining the types of networks that best supported JackTrip audio transfer. Subsequent tests evaluated JackTrip streaming implemented over four network scenarios: 1.) two computers transmitting audio between each other using hard-wired Ethernet connections within a local area network, 2.) remote computers transmitting audio using Ethernet over a wide area network using the commercial commodity Internet, 3.) remote computers transmitting audio using Ethernet over a wide area network utilising a high-speed national research and educational network (NREN), and 4.) computers using wireless Wi-Fi connections over the local area network. In the case of Wi-Fi networks, the lack of suitable bandwidth and possible network congestion yielded unstable streaming results as dropouts and audio errors were regularly present in all of the trials. The commodity network presented mixed results, and while the data in Chapter 5 showed minimal errors in the audio tests, separate testing dates shown in Hardin and Toulson (2019) provided a higher number of errors across all sets of streaming tests. This may reflect similar issues to Wi-Fi, where dates of higher network congestion can lead to unreliable service. Additionally the case of Wi-Fi streaming resulted in higher latency times than the Ethernet streams, and increased buffer sizes applied in the commercial network streams also accounted for greater latency. The LAN streaming tests performed exceptionally well, showing low latency speeds below the threshold of human echo perception (30 ms) with very rare to no cases of dropouts or errors over the entire series of tests. More importantly, repeated WAN trials utilising NRENs provided consistent successful results mirroring the reliability of the LAN performance. Similar to the successful results of other real-time streaming applications exploiting NREN networks for media transfer (Drioli, Allocchio, and Buso, 2013; Ferguson, 2015), the high-speed network tests offered proof that real-time audio distribution can effectively occur over extended distances.

7.2.2 Opportunities for Improved Performance of Real-time Audio Streaming

NRENs proved to be robust enough to support remote IoT music applications with reliable audio transmission. However, the lack of stability on modern, commodity Internet networks challenged a case for universal adoption, especially since NRENs are often not accessible to the general public. However, a developing communication infrastructure that may provide more reliable, real-time audio streaming over the commercial Internet is the 5G network.

Some key features of 5G as detailed by ITU-R specifications is that it will provide downlink peak data rates of 20 Gbit/s and uplink peak data rates of 10 Gbit/s (SG05, I.T.U.R., 2017). The minimal requirement latency speeds call for 4 ms for enhanced mobile broadband (eMBB) and 1 ms for ultra-reliable and low-latency communications (URLLC). Baratè, Haus, and Ludovico (2018) explain that these figures translate into 10 to 1000 times higher data transfer speed, and 5 times the reduced end-to-end latency of current commercial networks.

In relation to music, attempts to conduct distributed music concerts utilising 5G networks have been successfully accomplished. In 2018, the telecommunications company, *Ericsson*, set out to conduct live, real-time music concerts between London, England and Berlin, Germany in order to demonstrate the effectiveness and possible opportunities of 5G communication technology (Patzold, 2018). In one of the concerts, musician and Professor of Wireless Communication, Mischa Dohler, played the piano at the Brandenburg Tor in Berlin while his daughter, Noa, sung accompanying vocals at the Guildhall in London. Using a 5G network established by Ericsson to transmit media between the two locations, negligible delay was observed between the performance venues, with Dohler reporting around 20 ms of end-to-end latency between the two sites and no noticeable errors in the audio stream (Dohler, 2018).

5G is still currently in its early stages, but its promise to deliver high bandwidth data transfer with low-latency speeds makes an IoT music application with errorless, real-time media transmission a real possibility in the future.

7.2.3 Discussion on User Insight Evaluations

The user insight evaluations presented documented feedback from music producers answering Research Questions 1 & 2, and helped identify new production workflows that can emerge from IoT-enabled music systems as well as benefits and challenges these systems can offer. The feedback responses provided greater awareness of modern production techniques used by musicians, and meanwhile shared a modern appeal and affinity for the use of physical and analogue music processing systems compared to digital software. The feedback similarly helped develop an understanding whether remote IoT-based engagement with musical devices could influence greater hardware-based music production practices. It was observed that despite their professional experiences (respondents self-identifying as music producers or casual music makers), a vast majority of the questionnaire respondents, nearly 75%, reported using a personal computer with digital music software in their standard production processes. Comparatively, all of the professional producers and sound engineers in the interviews defaulted to using software as their primary tool to mix music. While the use of digital software and patches is the preferred case for most modern musicians, the evaluations revealed that physical and analogue hardware still have value in music production. A majority of the survey respondents acknowledged scenarios where hardware provided better results than their software equivalents. Furthermore, almost 90% of respondents reported they would use more physical devices if they were more accessible and would equally be inclined to incorporate IoT-enabled music systems in their personal workflows if they were widely implemented. The appreciation for hardware in the respondent data coincides with sympathies expressed in section 2.3.3 of the literature review and reiterates the views of the interviews shared by Reidy (2014).

The next two sections dive deeper into the survey feedback by providing a greater discussion of the perceived benefits and potential challenges of IoT-enabled music systems collected within the user insight evaluations.

7.2.4 Primary Perceived Benefits of an IoT-enabled Music System

The thematic analysis of the respondent data from both the questionnaires and interviews presented various areas where music producers favoured incorporating an IoT music system into music production. A primary benefit discussed among respondents is the increased opportunity to engage music technology and develop personal production techniques with remote hardware. Responses reflected how IoT can provide more flexibility in mixing and recording by delivering a greater range of analogue techniques in conjunction to the digital processes prominently used in personal settings. With respect to analogue production, the research revealed that a desire for analogue techniques is still prevalent among modern music producers. While many respondents felt software production techniques have vastly improved over time and are, in some instances, preferred to hardware, some felt that the in-the-box environment is too artificial and sterile and presents difficulty in replicating the subtle tonalities of analogue equipment. Similarly, the non-linear characteristics of analogue offers appeal in that hardware devices can generate completely unique sound characteristics with each use, likening the experience to playing a real, physical instrument. Lastly, nostalgia may be a perceived factor in the desirability of analogue, with some arguments mentioning it provides a more authentic music experience that resists modern culture (Brennan, 2018), and that it similarly gives younger generations new technology complementary to their digital lifestyles (Patel, 2017). In the evaluations, Wilde additionally references sentimental values attached to reconnecting and engaging with hardware-driven processes of the past.

Furthermore, the driving theme of the user insight evaluations centred on accessibility and IoT granting music producers access to rare and desirable music processing hardware not normally available in their standard production environments. The evaluations presented a strong case for IoT in music production arguing that it serves to break down the location barrier between the music producer and physical music technology, adding greater variety to music production and directly impacting consumers and influencing their production methods. Within the evaluations it was discussed that IoT may allow individuals with limited resources to expand their experiences making music outside of their traditional means. Modern music processes are typically split between a professional 'in-studio' and personal 'in-home' experience, with both offering advantages and aesthetics that drive productivity in their respective spaces. The opportunity to virtually extend aspects of the studio into a personal space can create a higher level of collaboration that can reshape modern work ethics. Additionally, for people who have geographical, financial, or physical limitations that restrict their access to desired music hardware, or those who primarily work in spaces where these devices are not readily available, remotely accessible hardware has been cited as a possible tool to expand creativity as it grants those users opportunities to produce more distinctive and expressive sounds. Lastly, IoT can help bridge the gap between analogue and digital music technology by giving new generations of producers the opportunities to find unique sounds and generate a wider selection of sonic content for more innovative music making. For a growing number of musicians, as identified by Jonze (2010), who feel that the "bedroom" or their personal space is essential to their production aesthetics, the user insight feedback shared cases where IoT can grant these musicians the opportunity to observe, learn, and experience the qualities of hardware systems that they would not normally investigate from their preferred working conditions.

Finally, independent collectors and commercial entities that provide musical hardware to consumers can benefit from IoT-enabled devices since virtually-accessible hardware grants them additional vehicles to distribute their products and services to customers. Respondent feedback alluded to new enterprise and market schemes that could be arranged by companies and collectors of rare and bespoke hardware through remote consumption of their equipment. This can effectively result in new mechanisms to generate income. Drawing back to the model of the Audio Hunt (https://www.theaudiohunt.com/) where audio files are distributed amongst users and processed by the device owner for the customer, IoT can expand upon this idea by allowing customers to connect directly to the remote devices and independently facilitate their own mixing sessions without depending on the owner's time and availability. These virtually-hireable music systems have great potential to benefit the hardware owners; however, this research subsequently revealed hopes that wide spread accessibility could result in saved costs for the consumers as well.

7.2.5 Primary Perceived Challenges of an IoT-enabled Music System

Any major technological development comes with inherent risks, and one of the major concerns surrounding the implementation of an IoT based music system is its perceived reliability. A high percentage of concerns surrounding IoT music systems centred around audio quality and the reliance on stable Internet connections. Some felt that current Internet speeds render these systems unusable, and that the constant need for a steady network connection is a deterrent when hands-on hardware or software works efficiently, independent of the Internet. Several respondents hinted at latency being a concern for tracking and real-time engagement, while others felt the Internet would invoke compression or other destructive processing techniques that would degrade the overall audio experience. The audio streaming evaluation was able to address some of these issues, showing that high-quality, real-time audio transfer can occur with less than 30 ms of latency and negligible distortion to the source audio stream on robust and high-speed computing networks. However, the same quality and performance standards would require enhancements to the commercially available Internet to be widely adopted.

Additionally, regulation of IoT accessible devices is another concern that was regularly mentioned by respondents in the research evaluations. Making musical hardware more accessible to the public could inadvertently decrease how accessible they are to their regular users. Compared to software patches, there are only a finite number of physical music units that exist, and depending on the brand and model, some devices are more highly sought after than others. If desirable hardware can be easily accessed at anytime from anywhere in the world there could be an overcrowding of the equipment and potentially stifle musical experiences and engagement for producers. Several respondents proposed implementing a virtual booking system that could help regulate use. However, concerns emerge over long queues to operate the devices and may eventually dissuade users from seeking physical devices altogether due to further drops in accessibility. These factors can only be fully assessed after an IoT music system is broadly employed and consumer trends observed, but prior planning and analysis of foreseeable complications are imperative to proper execution.

8. Conclusion

This chapter concludes the thesis by offering a concise discussion of the main contributions to knowledge as related to the research questions, and summarises the analysis undertaken in addressing the goals and aims of the research. Additionally, proposed future work is presented stemming from the findings of the work.

8.1 Summary of Main Contributions to Knowledge

To recap, the original contributions to knowledge obtained in this research are:

- A first original, detailed analysis of open source IoT technologies with respect to creative music applications.

- The creative development of a unique and innovative IoT demonstrator unit for use in music production scenarios, enabling and realising concepts including the ‘Internet connected reverb chamber’ and ‘Internet connected hardware units’ for music production, and hence enabling the first detailed evaluation of the concept of the ‘virtually-extended music studio’.

- The first analysis of Internet-controlled hardware alongside Internet streaming protocols for real-time, two-way audio streaming and real-time processing via the Internet.

- The first and most detailed critical analysis specifically of JackTrip and WebRTC streaming protocols in supporting high quality, real-time audio transfer across a number of modern computing networks.

- The first case of documented feedback from practitioners and experts in music production addressing impressions, principally highlighting perceived opportunities and concerns, of IoT-enabled music production systems.

- Original and unique implementation of enhanced mixed-method methodologies to critically investigate practical uses of the Internet of Things opportunities within a creative industry, focusing specifically on music production.

The primary contribution identified in the research is its in-depth academic investigation identifying and exploring new opportunities that may arise from the convergence of the Internet of Things and music production. Innovative audio applications are regularly being evaluated through virtual processes, and in 2001 an annual conference regarding *New Interfaces of Musical Expression* has brought together experts across the world to share knowledge and expertise for expressing music using digital interfacing and human-computer interaction (Poupyrev, Lyons, and Fels, 2001). In relation, this research uses an analytical approach to investigate new paradigms emerging from IoT-driven music production, and incorporates practical design, testing, and evaluation of untraditional, virtual methods to promote meaningful engagement with music technology.

In addressing the research questions raised in Chapter 3 of the thesis, three knowledge areas have seen direct contributions:

IoT Capabilities to Augment Physical Music Production Processes

Modern IoT frameworks facilitate ubiquitous communication and data exchange, granting network-based interactivity with physical music hardware. This is demonstrated in the research by the creation of unique user experiences that merge IoT control protocols and the transfer of high resolution multimedia data, effectively pushing the idea of a virtualised studio where distributed music devices can be engaged remotely from a centralised environment. Testing and experimentation demonstrated that interconnected microcontroller units can serve as resourceful tools to operate remote audio systems, while HTML5 Websockets allowed web browsers to function as portable vehicles allowing engagement and control of physical systems. The use of Jacktrip and low-latency audio streaming platforms allows audio to be transmitted across vast distances without perceivable losses of integrity, eliminating the dependence on physical audio connections and cables.

IoT-enabled Music Systems Influencing New Workflows for Music Production

Networked music hardware allows methods for distributed music composition to become readily available to the wider population. Through IoT it is shown that unique opportunities are presented to better bridge the musician to interconnected music production devices by removing the need of physical presence. A shift in musical engagement promoted by the Internet of Things adds new layers of complexity to the music making process resulting in more opportunities to discover and develop unique sound experiences. Similarly, having options to mix and record music using a combination of hardware and software resources offers greater flexibility in production techniques.

Potential Benefits Emerging from an IoT-enabled Music System

IoT and the virtually-extended music studio helps expand the professional studio experience to a culture of musicians who work independently outside of traditional production spaces, and additionally to those who may have physical limitations restricting them from accessing desired devices and technology. The qualitative evaluations provided prospective insights into how access to rare and exclusive devices around the world can provide creative opportunities for more inventive development of personal projects and musical works that may not be experienced if solely reliant on digital software techniques. These opportunities are not limited to hardware, but material spaces with unique acoustics can be adapted into creative musical workflows as echo chambers and natural, real-time reverb processors. Additionally, access to a variety of production tools allows producers to be more informed about the range of technology available to them and sounds they can acquire, potentially improving the overall quality of music they can produce. Finally, IoT-enabled music hardware has the opportunity to create new business models for hardware exchange and hire, which could generate greater value for neglected technology and spaces while potentially raising demands for hardware and lower costs for users.

While this research presents explicit contributions to knowledge for music production, an important impact comes from the consideration that IoT-driven processes can reflect greater cultural benefits for the larger creative industry.

IoT Extension to the Greater Creative Industry

Complex IoT processes have become more accessible throughout time, and a number of features in this research were made possible due to low cost electronics and open source tools that are free and available to the public. Open source resources can provide opportunities for understanding how a range of IoT applications are developed and evaluated, and aid in the development and reproductions of future creative projects. The use of networked technology also provides more opportunities for collaboration amongst individuals and provides non-traditional methods for working with technology to express creative works.

The research evaluations addressed how IoT can be expanded both artistically and innovatively outside of music production. Questionnaire respondents felt that IoT can offer benefits to health and safety, with some respondents posing ideas of how hospitals can better monitor patients with networked wearables. Another opportunity mentioned is how IoT can provide better regulation of speed laws by equipping vehicles with GPS monitoring hardware that rewards or penalises drivers, effectively gamifying traffic safety. Respondents additionally saw scenarios where people with disabilities or limited mobility could have more opportunities to interact with remote music systems and have more handicap friendly interfaces available to interact with standard hardware. In more distinctively creative applications, respondents felt that IoT can impact real-time interaction with networked robots, possibly in a live performance setting for generative performances including dance and visual art, or even the control of networked instruments. Strong cases were also made for sound-based applications outside of traditional music production, such as applying the real-time reverb scenarios using acoustic spaces to create more true-to-life sound effects and experiences for film and gaming.

The development and progression of this research sheds light on how anyone with imagination and original ideas can benefit from the IoT architecture. Artists can profit from collaborative relationships and open source data exchange that enables the use, modification, and development of embedded systems for greater engagement with both desired technology and their target audiences. IoT technology is not limited solely to industrial and economic driven applications, but can be utilised to benefit the broader, diverse interests of society as well.

8.2 Future Work

Being able to access and control remote objects is part of the allure of the Internet of Things, however, IoT is not limited to simply interconnecting devices over the Internet. An important product of IoT applications is the large amount of information produced by sensing and actuating devices, and these complex data streams can be collected and manipulated into performing other in-depth, data-driven processes. Future advances in this research could be achieved through added intelligence to IoT processes, such as incorporating machine learning and smart algorithms, which can help optimise functionality and improve interaction with the interconnected musical devices. One scenario demonstrating this is the idea of smart network path selection, where round-trip data packets are sent from a client connection to the server to determine which network path offers the best speed and most stable link for reliable data transmission.

Expanding upon added intelligence, one of the most important factors for survey respondents who reported preferred use of in-the-box software for music production is the ability to recall presets and previous settings of a mix. A practical benefit of software is the ability to save and restore settings of previous productions for future recreation, taking away some of the headache of recalling mix parameters associated with analogue hardware. Control information transmitted between IoT devices can be reinterpreted into practical actions used to designate settings and functionality of a device. One example of this is using numerical data values delivered by IoT interfaces to set the position of a servo motor. Servo motors are popular types of actuators that allow precise control of linear or rotational movement, and their positional feedback can be used to dictate stepped movements and desired motor positions. In the mentioned scenario, a numerical value sent from a control input of an interface, like a virtual knob, can be translated into the exact rotational position of a servo motor. Being able to manoeuvre the physical motor to a precise location with virtual data allows predetermined positions to be re-obtained, taking away the guess work of trying to manually reset the motor position by hand. More importantly, the digital position data can be saved and stored into system memory that can be recalled and used for automation in future instances of a production.

A last future work of this research is the consolidation of a complete web-based interface. With control aspects implemented into a webpage with Websockets, WebRTC had the greatest potential for adding both high-resolution audio transfer and a real-time video stream of the remote hardware directly into the browser with its HTML5 backbone. The audio streaming evaluations in Chapter 5 deemed WebRTC unsuitable for music transfer in this research, but as the research neared conclusion, companies like mix:analog have had some success incorporating both lossless and lossy real-time audio streaming into an IoT-enabled music application for real-time monitoring. mix:analog requires the initial upload of an audio file to their servers versus streaming the audio directly from an audio editor on the client’s computer as proposed in this research, but they have worked with technology like Web Audio API to deliver real-time audio monitoring of the processed music and visual feedback in the form of VU meters on a graphical representation of the controlled musical device. This technique has worked well for their application and removed the need for both a live video feed and round-trip audio stream. However, similar to JackTrip utilising the JACK Audio Connection Kit for audio distribution in its application, the JACK Audio API could potentially be configured into HTML code or a web plug-in allowing both the control interface and audio streaming to be provided directly into the browser. This would give users more freedom to connect their preferred audio editors to the API and greater control over the audio stream for recording and tracking.

References

Literature References

AES Standards Committee, 2018a. AES standard for audio applications of networks - Open Control Architecture - Part 1: Framework. *Audio Engineering Society, Inc*.

AES Standards Committee, 2018b. AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability. *Audio Engineering Society, Inc*.

Bandyopadhyay, D., and Sen, J., 2011. Internet of Things – Applications and Challenges in Technology and Standardization. [pdf] Available at: <http://arxiv.org/pdf/1105.1693v1.pdf> [Accessed 17 April 2016]

Baratè, A., Haus, G. and Ludovico, L.A., 2018, June. Advanced Experience of Music through 5G Technologies. In *IOP Conference Series: Materials Science and Engineering*(Vol. 364, No. 1, p. 012021). IOP Publishing.

Baskar, S., 2017. 6 Reasons Why Designing UX for IoT is So Difficult. *Machine Design*. [online] 28 July. Available at: <http://www.machinedesign.com/industrial-automation/6-reasons-why-designing-ux-iot-so-difficult> [Accessed 20 February 2018]

Bonaccorsi, A. and Rossi, C., 2003. Why open source software can succeed. *Research policy*, 32(7), pp.1243-1258.

Brennen, B., 2019. *Opting Out of Digital Media*. New York: Routledge.

Borenstein, G., 2008. A Brief History of the Music Industry over the Last 10 Years. *Urban Honking*, [blog] 12 January. Available at: <http://urbanhonking.com/ideasfordozens/2008/01/12/a_brief_history_of_the_music_i/> [Accessed 5 June 2016]

Bouillot, N. and Cooperstock, J.R., 2009, June. Challenges and Performance of High-Fidelity Audio Streaming for Interactive Performances. In NIME (pp. 135-140).

Bouillot, N., Cohen, E., Cooperstock, J.R., Floros, A., Fonseca, N., Foss, R., Goodman, M., Grant, J., Gross, K., Harris, S. and Harshbarger, B., 2009. Aes white paper: Best practices in network audio. *Journal of the Audio Engineering Society*, 57(9), pp.729-741.

Braun, V. and Clarke, V., 2006. Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), pp.77-101.

Cáceres, J.P. and Chafe, C., 2010. JackTrip: Under the hood of an engine for network audio. *Journal of New Music Research*, 39(3), pp.183-187.

Cage, J., 1990. John Cage: Autobiographical Statement. *JohnCage.org* [online] Available at: <http://www.johncage.org/autobiographical_statement.html> [Accessed 23 May 2016]

Chalcraft, E., 2012. Underwater by David Bowen. *Dezeen Magazine*. [online] Available at: <http://www.dezeen.com/2012/11/04/underwater-installation-by-david-bowen/> [Accessed 27 May 2016]

Char, K., u. Internet of Things System Design with Integrated Wireless MCUs. *Silicon Labs*. [PDF] available at <http://www.silabs.com/Support%20Documents/TechnicalDocs/Internet-of-Things-System-Design-with-Integrated-Wireless-MCUs.pdf> [Accessed 3 March 2016]

Creswell, J.W. and Plano Clark, V.L., 2007. *Designing and Conducting Mixed Methods Research*. Thousand Oaks: Sage.

Curtis, S., 2016. Survey Research. In: B. Wheeler and K, Murphy, ed. 2016. *Music Therapy Research: Third Edition.* Dallas: Barcelona Publishers. Ch. 27.

De Carvalho, A.T., 2012. The discourse of home recording: authority of pros and the sovereignty of big studios. *Journal of the Art of Record Production*, 7. [online] Available at <https://www.arpjournal.com/asarpwp/the-discourse-of-home-recording-authority-of-%E2%80%9Cpros%E2%80%9D-and-the-sovereignty-of-the-big-studios/> [Accessed 18 February 2019]

Dixon, A., 2016. 31 December. How Has The Recording Studio Affected The Ways In Which Music Is Created? *Classic Album Sundays*. [blog] Available at: <http://classicalbumsundays.com/how-has-the-recording-studio-affected-the-ways-in-which-music-is-created/> [Accessed 11 May 2018]

Dohler, M., 2018. World's First 5G Distributed Concert London - Berlin. [video online] Available at: <https://youtu.be/mB_w-ml-dZY> [Accessed 17 April 2019].

Drioli, C., Allocchio, C. and Buso, N., 2013. Networked performances and natural interaction via LOLA: Low latency high quality A/V streaming system. In *Information*

Technologies for Performing Arts, Media Access, and Entertainment. Berlin: Springer, pp. 240-250.

EPSRC, 2016. *New Internet of Things Research Hub announced*. [press release] 06 January 2016. Available at: <https://www.epsrc.ac.uk/newsevents/news/iotresearchhub/> [Accessed 28 March 2017]

Evens, A., 2002. Sound ideas. In: Massumi, B., 2005. *A shock to thought: Expression after Deleuze and Guattari*. London: Routledge, pp. 171-187.

Evren, A., 2015. Tracing the History of Music Streaming, and Why It's More Important Now Than Ever. *Bit-of-News*, [blog] 17 June. Available at: <http://news.bitofnews.com/tracing-the-history-of-music-streaming-and-why-its-more-important-now-than-ever/> [Accessed 4 June 2016]

Fabiani, M., Friberg, A., and Bresin, R., 2013. *Systems for Interactive Control of Computer Generated Music Performance*. In: Kirke, A. and Miranda, E., 2013. *Guide to Computing for Expressive Music Performance*. London: Springer-Verlag London 2013, Ch. 2.

Ferguson, P., 2013, May. Using Low-Latency Net-Based Solutions to Extend the Audio and Video Capabilities of a Studio Complex. In *Audio Engineering Society Convention 134*. Audio Engineering Society.

Ferguson, P., 2015. Real-time long-distance music collaboration using the Internet. In: Hepworth-Sawyer, R., Hodgson, J., Toulson, R. and Paterson, J.L., 2016. *Innovation in music II*. Future Technology Press, UK, pp. 174-178.

Fleischman, E., 1998. WAVE and AVI codec registries. *Microsoft Corporation*. [online] Available at: <https://tools.ietf.org/html/rfc2361> [Accessed 3 March 2018]

Foley, M., n.d. National Research and Education Networks. *GEANT Association*. [pdf] Available at: <https://www.casefornrens.org/Resources_and_Tools/Document_Library/Documents/What%20is%20an%20NREN.pdf> [Accessed 22 December 2018]

Fonseca, N. and Monteiro, E., 2003, March. Latency in Audio Ethernet Networks. In *Audio Engineering Society Convention* 114. Audio Engineering Society.

Gartner, 2015. Gartner Says 6.4 Billion Connected "Things" Will be in Use in 2016, Up 30 Percent From 2015. [press release] 10 November 2015, Available at: <http://www.gartner.com/newsroom/id/3165317> [Accessed 18 April 2016]

Goddard III, R.D. and Villanova, P., 2006. Designing surveys and questionnaires for research. *The psychology research handbook: A guide for graduate students and research assistants*, pp.114-125.

Gresham-Lancaster, S., 1998. The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music. *Leonardo Music Journal*, pp. 39-44

Grote, F., Anderson, K., and Knees, P., 2015. Collaborating with Intelligent Machines: Interfaces for Creative Sound. *CHI 2015 Workshop*, pp. 2345-2348.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M., 2013. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. [pdf] Available at <http://arxiv.org/ftp/arxiv/papers/1207/1207.0203.pdf> [Accessed 24 February 2016]

Hardin, M. and Toulson, R., 2019, March. Quantitative Analysis of Streaming Protocols for Enabling Internet of Things (IoT) Audio Hardware. In *Audio Engineering Society Convention 146*. Audio Engineering Society.

Hayes, A., 2018. How Does Spotify Make Money? *Investopedia*. [blog] Available at: <https://www.investopedia.com/articles/investing/120314/spotify-makes-internet-music-make-money.asp> [Accessed 28 April 2019]

Hendrick, S, 2018. The Impact of Open Source Software on Developing IoT Solutions. *RT Insights* [online] Available at: <https://www.rtinsights.com/the-impact-of-open-source-software-on-developing-iot-solutions/> [Accessed 10 April 2018]

Hoffman, C., 2017a. What's the Difference Between TCP and UDP? *How-To Geek.* [blog] 7 June. Available at: <https://www.howtogeek.com/190014/htg-explains-what-is-the-difference-between-tcp-and-udp/> [Accessed 15 February 2018]

Hoffman, C., 2017b. Wi-Fi vs. Ethernet: How Much Better Is a Wired Connection? *How-To Geek.* [blog] 7 June. Available at: <https://www.howtogeek.com/217463/wi-fi-vs.-ethernet-how-much-better-is-a-wired-connection/> [Accessed 15 February 2018]

Jonze, T., 2010. The bedroom artists who prefer creative solitude. *The Guardian.* [online] Available at: <https://www.theguardian.com/music/2010/apr/04/tim-jonze-indie-music> [Accessed 23 April 2018]

Kajornboon, A.B., 2005. Using interviews as research instruments. *E-journal for Research Teachers*, *2*(1), pp.1-9.

Keith, C., 2010. Agile game development with Scrum. New Jersey: Pearson Education.

Kinsella, B., 2018. New Voicebot Report Says Nearly 20% of U.S. Adults Have Smart Speakers. *Voicebot,* 7 March. [online] Available at: <https://voicebot.ai/2018/03/07/new-voicebot-report-says-nearly-20-u-s-adults-smart-speakers/> [Accessed 12 March 2019]

Lacey, L., 2015. How the Internet of Things Could Impact Music Composition, Production & Performance. *Ask.Audio*. [online] Available at <https://ask.audio/articles/how-the-internet-of-things-could-impact-music-composition-production-performance> [Accessed 21 March 2016]

Marshall, M.N., 1996. Sampling for qualitative research. *Family practice*, *13*(6), pp.522-526.

Mattern, F. and Floerkemeier, C., 2010. From the Internet of Computers to the Internet of Things. In *From active data management to event-based systems and more* (pp. 242-259). Springer, Berlin, Heidelberg.

Mayton, B.D., Dublon, G., Joliat, N. and Paradiso, J.A., 2012, May. Patchwork: Multi-User Network Control of a Massive Modular Synthesizer. In *NIME*.

Miorandi, D., Sicari, S., De Pellegrini, F. and Chlamtac, I., 2012. Internet of things: Vision, applications and research challenges. *Ad hoc networks*, *10*(7), pp.1497-1516.

Moir, Z., Ferguson, P., and Smith, G., 2019. Real-Time Remote, Interactive Recording Sessions: Music Production Without Boundaries. In Hepworth-Sawyer, R., Hodgson, J., and Marrington, M., 2019. *Producing Music*. New York: Routledge. Ch. 12.

Moore, B.C. and Tan, C.T., 2003. Perceived naturalness of spectrally distorted speech and music. *The Journal of the Acoustical Society of America*, *114*(1), pp.408-419.

Moore, B.C., Tan, C.T., Zacharov, N. and Mattila, V.V., 2004. Measuring and predicting the perceived quality of music and speech subjected to combined linear and nonlinear distortion. *Journal of the Audio Engineering Society*, *52*(12), pp.1228-1244.

Morris, C., 2016. Vinyl Record Sales Are At A 28-Year High. *Fortune*. [blog] 16 April. Available at <http://fortune.com/2016/04/16/vinyl-sales-record-store-day/> [Accessed 17 February 2019]

Newmarker, C. and Buntz, B., 2016. 11 Innovative IoT Use Cases. *Internet of Things Institute*. [blog] 6 December. Available at: <https://www.iotworldtoday.com/2016/12/07/11-innovative-iot-use-cases/> [Accessed 19 April 2017]

O'Brien, C., 2015. 10 gadgets that demonstrate the potential of the Internet of Music Things. *Venture Beat*, [blog] 15 June. Available at: <http://venturebeat.com/2015/06/15/10-gadgets-that-demonstrate-the-potential-of-the-internet-of-music-things/> [Accessed 7 June 2016]

Patel, N., 2017. The Resurgence of Analog Makes Sense in a Post-Digital Era. *Inverse.com*. [online] Available at: <https://www.inverse.com/article/27543-david-sax-revenge-of-analog-digital> [Accessed 1 January 2020]

Paterson, J., Toulson, E.R., Lexer, S., Webster, T., Massey, S. and Ritter, J., 2015. Interactive digital music: enhancing listener engagement with commercial music. In: Hepworth-Sawyer, R., Hodgson, J., Toulson, R. and Paterson, J.L., 2016. *Innovation in music II*. Future Technology Press, UK, pp. 193-209.

Patzold, M., 2018. It's Time to Go Big with 5G [Mobile Radio]. *IEEE Vehicular Technology Magazine*, *13*(4), pp.4-10.

Poupyrev, I., Lyons, M.J. and Fels, S., 2001, March. New interfaces for musical expression. In *CHI'01* Extended Abstracts on *Human Factors in Computing Systems* (pp. 491-492). ACM.

Prism Sound, 2018. Glossary definition of 'Total Harmonic Distortion Plus Noise. *Prism*. [online] Available at: <http://www.prismsound.com/define.php?term=Total_Harmonic_Distortion_Plus_Noise> [Accessed 4 February 2018]

Prosuk, A., 2017. What is Total Harmonic Distortion (THD)? *Siemens*. [blog] Available at: <https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/What-is-Total-Harmonic-Distortion-THD/ta-p/440629> [Accessed 2 May 2019]

Rec, I.T.U.T., 1996. P. 800: Methods for subjective determination of transmission quality. *International Telecommunication Union, Geneva*.

Reddell, T., 2003. Laptopia: the spatial poetics of networked laptop performance. *Contemporary Music Review*, *22*(4), pp.11-23.

Reidy, T., 2014. Retro electronics still popular – but why not just use modern software? *The Guardian*. [online] Available at: <https://www.theguardian.com/music/2014/feb/15/old-electronic-instruments-popular-software> [Accessed 23 April 2018]

Robjohns, H., 2008. Digital Problems, Practical Solutions: Getting The Best From Digital Audio. *Sound on Sound*. [blog] Available at: <https://www.soundonsound.com/techniques/digital-problems-practical-solutions> [Accessed 29 January 2018]

Robjohns, H. and White, P., 2019. Glossary Of Technical Terms. *Sound on Sound*. [blog] Available at: <https://www.soundonsound.com/sound-advice/glossary-technical-terms> [Accessed 5 July 2019]

Rouse, M., 2016b. Latency. *WhatIs.com* [online] Available at: <https://whatis.techtarget.com/definition/latency> [Accessed 23 December 2018]

Schober, M., 2006. Virtual environments for creative work in collaborative music-making. *Virtual Reality.* Volume 10, Issue 2, pp 85-94. London: Springer-Verlag London Limited.

Series, B., 2014. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*.

SG05, I.T.U.R., 2017. Draft New Report ITU-R M.[IMT-2020. TECH PERF REQ]-Minimum requirements related to technical performance for IMT-2020 radio interface(s). *ITU-R SG05 Contribution 40*.

Shakhovskoy, J. and Toulson, R., 2015. Future Music Formats: Evaluating The 'Album App.' *Journal on the Art of Record Production*, Issue 10. [online] Available at: <http://arpjournal.com/future-music-formats-evaluating-the-album-app/> [Accessed 29 June 2016]

siteadmin, 2013, 4 March. What future is there for independent recording studios? *Mill Hill Music Complex.* [blog] Available at: <https://www.millhillmusiccomplex.co.uk/what-future-is-there-for-independent-recording-studios/> [Accessed 23 April 2018]

Sinatra, M., 2014. Research and Commodity Networks: What are they and what's the difference? *NANOG on the Road*, 20 May. [pdf] Available at: <https://www.nanog.org/sites/default/files/Research-Commodity.pdf> [Accessed 21 December 2018]

Smith, S.L., Choueiti, M. and Pieper, K., 2019. Inclusion in the recording studio? Gender and race/ethnicity of artists, songwriters and producers across 700 popular songs from 2012-2018. *USC Annenberg Inclusion Initiative.*

SoundWIRE Group, 2010. SoundWIRE Group at CCRMA. *CCRMA*. [online] Available at: <https://ccrma.stanford.edu/groups/soundwire/> [Accessed 22 April 2019]

Stankovic, J., 2014. Research Directions for the Internet of Things. *IEEE Internet of Things Journal*. [online] Available at: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6774858&tag=1> [Accessed 29 February 2016]

Statistical Services Centre, 2001. *Approaches to the Analysis of Survey Data*. Reading: Statistical Services Centre, University of Reading. Available at: <https://www.ilri.org/biometrics/TrainingResources/Documents/University%20of%20Reading/Guides/Guides%20on%20Analysis/ApprochAnalysis.pdf> [Accessed 11 September 2018]

Strickland, J., 2008. Does digital sound better than analog? *HowStuffWorks.com*. [online] Available at: <http://electronics.howstuffworks.com/digital-versus-analog.htm> [Accessed 21 March 2017]

Stuckey, H.L., 2013. Three types of interviews: Qualitative research methods in social health. *Journal of Social Health and Diabetes*, *1*(2), p.56.

[The User], 2000. Silophone. *Undefine*. [website] Available at: <http://www.undefine.ca/en/projects/silophone/> [Accessed 30 October 2017]

Temme, S., 1992. Audio distortion measurements. *Application Note, Bruel & Kjar*.

Thiesse, F., and Michahelles, F., 2009. Building the Internet of Things Using RFID: The RFID Ecosystem Experience. *IEEE Computer Society*. [pdf] Available at: <http://sam-raymer.com/pub/welbourne-ieeeic09.pdf> [Accessed 17 April 2016]

Toole, F.E., 1982. Listening tests-turning opinion into fact. *Journal of the Audio Engineering Society*, 30(6), pp.431-445.

Toulson, E. R., Campbell, W. and Paterson, J., 2014. Evaluating harmonic and intermodulation distortion of mixed signals processed with dynamic range compression, in: Innovation In Music 2013, Future Technology Press, ISBN 9780956151681.

Tuckett, A., 2004. Qualitative research sampling-the very real complexities. *Nurse Researcher*. 12(1): 47-61

Vaismoradi, M., Jones, J., Turunen, H. and Snelgrove, S., 2016. Theme development in qualitative content analysis and thematic analysis. *Journal of Nursing Education and Practice*, *6*(5), pp.100-110.

Vermesan, O. et al., 2009. The Internet of things: Strategic Research Roadmap. [online] Available at: <http://sintef.biz/upload/IKT/9022/CERP-IoT%20SRA_IoT_v11_pdf.pdf> [Accessed 13 April 2016]

Visnjic, F., 2013. Long Distance Art – One artist, two robots and three paintings in Vienna, Berlin and London. *Creative Applications Network*. [online] Available at: <http://www.creativeapplications.net/vvvv/long-distance-art-one-artist-two-robots-and-three-paintings-in-vienna-berlin-and-london/> [Accessed 22 May 2016]

Voldhaug, J.E., Hellerud, E., and Svensson, P., 2006, May. Evaluation of packet loss distortion in audio signals. In Audio Engineering Society Convention 120. *Audio Engineering Society*.

WebRTC, 2011a. *WebRTC*. [online] Available at: <https://webrtc.org/> [Accessed 31 July 2017]

WebRTC, 2011b. *Frequent Questions*. [online] Available at: <https://webrtc.org/faq/> [Accessed 31 July 2017]

Weiser, M., Gold, R., Brown, J.S., 1999. The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal*. Vol 38, No. 4. [pdf] Available at: <http://www.cs.cmu.edu/~./jasonh/courses/ubicomp-sp2007/papers/03-weiser-origins.pdf> [Accessed 20 April 2016]

Whalley, I., 2015. Developing Telematic Electroacoustic Music: Complex networks, machine intelligence and affective data stream sonification. *Organised Sound*, 20, pp 90-98 [online] Available at <http://journals.cambridge.org/download.php?file=%2F2008_9D29F88E5E96FBE9E208F0745469B579_journals__OSO_OSO20_01_S1355771814000478a.pdf&cover=Y&code=23f51f9807891a661339879b90c07d0c> [Accessed 9 March 2015]

Worby, R., 2009. Tune on, tune in: John Cage's symphony for 12 radios. *The Guardian*. [online] Available at: <http://www.theguardian.com/music/2009/aug/06/john-cage-symphony-for-radios> [Accessed 23 May 2016]

Wright, M., 2005. Open Sound Control: an enabling technology for musical networking. *Organised Sound*, *10*(3), pp.193-200.

Zolfagharifard, E., 2013. One artist, two robots and three cities: Painter creates a work of art in Vienna - while robot arms create the same picture in London and Berlin at the SAME time. *Daily Mail*. [online] Available at: <http://www.dailymail.co.uk/sciencetech/article-2433471/Artist-Alex-Kiesslings-Long-Distance-uses-robots-copy-painting.html> [Accessed 22 May 2015]

Image References

DJ City, n.d. *PV8*. [image online] Available at: <https://cdn.djcity.com.au/wp-content/uploads/2016/07/22122415/115033_26028.jpg> [Accessed 11 November 2017].

ekawa, 2016. Guitar Hero: Live - Through the Fire and Flames (Expert 97%). [image online] Available at: < https://www.youtube.com/watch?v=cmB2ucBlrhQ> [Accessed 11 March 2019].

Design References

ARMmbed, n.d.a FRDM-K64F. *ARMmbed.* [online] Available at: <https://developer.mbed.org/platforms/FRDM-K64F/> [Accessed 1 July 2016]

ARMmbed, n.d.b. Interfacing Using RPC. *ARMmbed.* [online] Available at: <https://developer.mbed.org/cookbook/Interfacing-Using-RPC> [Accessed 19 October 2015]

ARMmbed, n.d.c. Interfacing with Python. *ARMmbed.* [online] Available at: <https://developer.mbed.org/cookbook/Interfacing-with-Python> [Accessed 19 October 2015]

ARMmbed, n.d.d. DC Motor. *ARMmbed.* [online] Available at: <https://developer.mbed.org/components/DC-Motor/> [Accessed 4 April 2017]

ARMmbed, 2012. Websocket Tutorial. *ARMmbed*. [online] Available at: <https://developer.mbed.org/cookbook/Websockets-Server> [Accessed 19 October 2015]

Cáceres, J., 2007a. About Soundwire. *CCRMA – Stanford University,* [webpage] Available at: <https://ccrma.stanford.edu/groups/soundwire/about/> [Accessed 6 June 2016]

Cáceres, J., 2007b. JackTrip Documentation. *CCRMA – Stanford University,* [webpage] Available at: < https://ccrma.stanford.edu/groups/soundwire/software/jacktrip/> [Accessed 31 July 2016]

Castrounis, A,. 2015. What is WebRTC and How Does It Work? *Innoarchitech*. [online] Available at: <https://www.innoarchitech.com/what-is-webrtc-and-how-does-it-work/> [Accessed 9 August 2017]

Davis, P. et al., 2001. JACK-AUDIO-CONNECTION-KIT. *Jackaudio.org*. [online] Available at: <http://jackaudio.org/api/> [Accessed 24 July 2017]

Dutton, S. 2012. Getting Started with WebRTC. *HTML5 Rocks*. [online] Available at: <https://www.html5rocks.com/en/tutorials/webrtc/basics/> [Accessed 4 August 2017]

Ford, S., 2010. Motor_HelloWorld. *ARMmbed*. [online] Available at: <https://developer.mbed.org/users/simon/code/Motor_HelloWorld/file/7bbc230e00d6/main.cpp> [Accessed 4 March 2017]

Gleason, K., 2015. WebRTC Live Video Stream Broadcasting from One-to-Many. *PubNub.* [online] Available at: <https://www.pubnub.com/blog/2015-08-27-webrtc-live-video-stream-broadcasting-from-one-to-many/> [Accessed 7 February 2017]

Hasler, C., 2010. Motordriver. *ARMmbed*. [online] Available at: <https://developer.mbed.org/users/littlexc/code/Motordriver/> [Accessed 4 March 2017]

HiveMQ, n.d. MQTT Essentials: Publish/Subscribe. *HiveMQ.* [online] Available at: <http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe> [Accessed 5 October 2015]

Kaazing, n.d. About HTML5 Websocket. *Websocket.org*. [online] Available at: <https://www.websocket.org/aboutwebsocket.html> [Accessed 1 July 2016]

Lubbers, P and Greco, R, n.d. HTML Websocket: A Quantum Leap in Scalability for the Web. *Websocket.org*. [online] Available at: <http://websocket.org/quantum.html> [Accessed 23 December 2016]

mbed official, 2012. EthernetInterface. *ARMmbed.org*. [online] Available at: <https://developer.mbed.org/users/mbed_official/code/EthernetInterface/docs/183490eb1b4a//classEthernetInterface.html> [Accessed 17 November 2015]

Mokrani, S., 2012a. Wifly_HelloWorld. *ARMmbed*. [online] Available at: <https://developer.mbed.org/users/samux/code/Wifly_HelloWorld/file/867d16e948eb/main.cpp> [Accessed 9 December 2015]

Mokrani, S., 2012b. WebsocketClient. *ARMmbed.org*. [online] Available at: <https://developer.mbed.org/users/samux/code/WebSocketClient/> [Accessed 17 February 2016]

MQTT, 2009. *Frequently Asked Questions*. [online] Available at: <http://mqtt.org/faq> [Accessed 21 December 2016]

Rouse, M., 2016a. Remote Procedure Call (RPC). *TechTarget*. [online] Available at: <http://searchmicroservices.techtarget.com/definition/Remote-Procedure-Call-RPC> [Accessed 2 February 2017]

Terrien, A., 2015. JQuery Knob. *AnthonyTerrien.com*. [online] Available at: <http://anthonyterrien.com/knob/> [Accessed 12 October 2016]

Audio Samples

Buzbe20, 2017. Ukele jingle.wav. *freesound*. [sound recording] Available at: <http://freesound.org/people/buzbe20/sounds/389758/> [Accessed 16 June 2017]

Hughesj333, 2010. 1000hz.wav. *freesound*. [sound recording] Available at: <http://freesound.org/people/hughesj333/sounds/99202/> [Accessed 16 June 2017]

kate_dilemma, 2016. Female Vocal 2 100bpm. *Freesound*. [sound recording] Available at: <https://freesound.org/people/kate_dilemma/sounds/364214/> [Accessed 15 January 2018]

kwahmah_02, 2015. 0Hz to 22500kHz in 30 seconds.flac. *freesound*. [sound recording] Available at: <http://freesound.org/people/kwahmah_02/sounds/262333/> [Accessed 19 July 2017]

MTG, 2016. Violin – A major. *freesound*. [sound recording] Available at <https://freesound.org/people/MTG/sounds/356188/> [Accessed 18 November 2017]

peridactyloptrix, 2013. African tribal drums, beat 3. *freesound*. [sound recording] Available at: <http://freesound.org/people/buzbe20/sounds/389758/> [Accessed 16 June 2017]

Tyops, 201. Dramatic Beat. *freesound*. [sound recording] Available at: <http://freesound.org/people/tyops/sounds/392620/> [Accessed 16 June 2017]

Appendices

Appendix A – Wifly mbed Code (Modified from Mokrani (2012))

```
#include "mbed.h"
#include "WiflyInterface.h"
#include "Websocket.h"
Serial pc(USBTX, USBRX);

/* wifly interface:
*       - p28 and p27 are for the serial communication
*       - p25 is for the reset pin
*       - p26 is for the connection status
*       - "virginmedia" is the ssid of the network
*       - "password" is the password
*       - true means that the security of the network is WPA
*/
Websocket ws("ws://***.***.*.***:****/ws");
WiflyInterface wifly(p28, p27, p25, p26, "*****", "************", true);

int main() {
    wifly.init(); //Use DHCP
    while (!wifly.connect());
    printf("IP Address is %s\n\r", wifly.getIPAddress());
    ws.connect();
    while (1) {
            ws.send("WebSocket Hello World from Marques over Wifly");
            wait(1.0);
    }
}
```

Appendix B – /FRDM_Ethernet_Helloworld/main.cpp

```
#include "mbed.h"
#include "EthernetInterface.h"

Serial pc(USBTX, USBRX);

int main() {
    EthernetInterface eth;
    eth.init(); //Use DHCP
    eth.connect();
    printf("IP Address is %s\n\r", eth.getIPAddress());
    // Prints I.P. Address to terminal upon connection
}
```

Appendix C – led_RPCSerial.html

```
<!doctype html>
<html>
  <head>
    <title>LED WebSockets Hello World</title>
    <meta charset="utf-8" />
    <style type="text/css">
      body {
        text-align: center;
        min-width: 500px;
      }

      #red{
        background-color: red;
        font-size: 1.25em;
        font-weight: bold;
      }

      #green{
        background-color: green;
        font-size: 1.25em;
        font-weight: bold;
      }

      #blue{
        background-color: blue;
        font-size: 1.25em;
        font-weight: bold;
      }
    </style>
    <script src="http://code.jquery.com/jquery.min.js"></script>
    <script>

      // log function
      log = function(data){
        $("div#terminal").prepend("</br>" +data);
        console.log(data);
      };

      $(document).ready(function () {
        $("div#message_details").hide()

        var ws;

        $("#open").click(function(evt) {
          evt.preventDefault();

          var host = $("#host").val();
          var port = $("#port").val();
          var uri = $("#uri").val();

          // create websocket instance
```

```
        ws = new WebSocket("ws://" + host + ":" + port + uri);

        // Handle incoming websocket message callback
        // *will only print if self.write_message() is in python code
        ws.onmessage = function(evt) {
          log("Message Received: " + evt.data)
          };

        // Close Websocket callback
        ws.onclose = function(evt) {
          log("***Connection Closed***");
          alert("Connection close");
          $("#host").css("background", "#ff0000");
          $("#port").css("background", "#ff0000");
          $("#uri").css("background",  "#ff0000");
          $("div#message_details").empty();

          };

        // Open Websocket callback
        ws.onopen = function(evt) {
          $("#host").css("background", "#00ff00");
          $("#port").css("background", "#00ff00");
          $("#uri").css("background", "#00ff00");
          $("div#message_details").show();
          log("***Connection Opened***");
        };
      });

      $("#close").click(function(evt) {
          ws.close();
      });

      $("#red").click(function(evt) {
          log("Sending Message: "+$("#red").val());
          ws.send($("#red").val());
      });

      $("#blue").click(function(evt) {
          log("Sending Message: "+$("#blue").val());
          ws.send($("#blue").val());
      });

      $("#green").click(function(evt) {
          log("Sending Message: "+$("#green").val());
          ws.send($("#green").val());
      });
    });
  </script>
</head>

<body>
  <h1>WebSockets Hello World</h1>
  <div id="connection_details">
```

```
      <label for="host">host:</label>
      <input type="text" id="host" value="localhost"
style="background:#ff0000;"/><br />
      <label for="port">port:</label>
      <input type="text" id="port" value="8888"
style="background:#ff0000;"/><br />
      <label for="uri">uri:</label>
      <input type="text" id="uri" value="/ws"
style="background:#ff0000;"/><br />
      <input type="submit" id="open" value="open" /></br>
      <input type="submit" id="close" value="close" />
    </div>

    <div id="led_buttons">
        </br>
        <h2>Select the LED you wish to turn on</h2>
        <button type="button" id="red" onclick="console.log('red')"
value="myled1">Red</button>
        <button type="button" id="green" onclick="console.log('green')"
value="myled2">Green</button>
        <button type="button" id="blue" onclick="console.log('blue')"
value="myled3">Blue</button>
    </div>

    <div id="terminal">

    </div>
  </body>
</html>
```

Appendix D – led_rpc.py

```
#!/usr/bin/env python

import tornado.httpserver
import tornado.websocket
import tornado.ioloop
import tornado.web
import socket

class WSHandler(tornado.websocket.WebSocketHandler):
    num = 0
    num_red = 0
    num_green = 0
    num_blue = 0
    def open(self):
        print 'new connection'

    def on_message(self, message):

        name = str(message)
        print name

        # Changes Write State Everytime Button Pressed
        if name == "myled1":
            if WSHandler.num_red == 0:
                WSHandler.num_red = 1
                myled1.write(WSHandler.num_red)
            elif WSHandler.num_red == 1:
                WSHandler.num_red = 0
                myled1.write(WSHandler.num_red)

        elif name == "myled2":
            if WSHandler.num_green == 0:
                WSHandler.num_green = 1
                myled2.write(WSHandler.num_green)
            elif WSHandler.num_green == 1:
                WSHandler.num_green = 0
                myled2.write(WSHandler.num_green)

        elif name == "myled3":
            if WSHandler.num_blue == 0:
                WSHandler.num_blue = 1
                myled3.write(WSHandler.num_blue)
            elif WSHandler.num_blue == 1:
                WSHandler.num_blue = 0
                myled3.write(WSHandler.num_blue)

    def on_close(self):
        print 'connection closed'

    def check_origin(self, origin):
        return True
```

```
from mbedRPC_new import *
serdev = 2 # For LPC1768
#serdev = 4 # For k64F
mbed = SerialRPC(serdev, 9600)
print 'test something'

#Turn on/off LED
myled1 = RpcDigitalOut(mbed, "myled1")
myled2 = RpcDigitalOut(mbed, "myled2")
myled3 = RpcDigitalOut(mbed, "myled3")



application = tornado.web.Application([
    (r'/ws', WSHandler),
])


if __name__ == "__main__":
    http_server = tornado.httpserver.HTTPServer(application)
    http_server.listen(8888)
    myIP = socket.gethostbyname(socket.gethostname())
    print '*** Websocket Server Started at %s***' % myIP
    tornado.ioloop.IOLoop.instance().start()
```

Appendix E - /RPC_Serial_LED/main.cpp

```
#include "mbed.h"
#include "mbed_rpc.h"

RpcDigitalOut myled1(LED1,"myled1");
RpcDigitalOut myled2(LED2,"myled2");
RpcDigitalOut myled3(LED3,"myled3");
RpcDigitalOut myled4(LED4,"myled4");

Serial pc(USBTX, USBRX);
int main() {
    //The mbed RPC classes are now wrapped to create an RPC enabled
version - see RpcClasses.h so don't add to base class

    // receive commands, and send back the responses
    char buf[256], outbuf[256];
    while(1) {
        pc.gets(buf, 256);
        //Call the static call method on the RPC class
        RPC::call(buf, outbuf);
        pc.printf("%s\n", outbuf);
    }
}
```

Appendix F – mult_slider_gui.html

```
<!doctype html>
<html>
  <head>
    <title>LED WebSockets Hello World</title>
    <meta charset="utf-8" />
    <style type="text/css">
      body {
        text-align: center;
        min-width: 500px;
      }

      #red, #green, #blue, #btn1, #btn2{
        font-size: 1.1em;
        font-weight: bold;
      }

      /*
      #red{
        background-color: red;
      }

      #green{
        background-color: green;
      }

      #blue{
        background-color: blue;
      }
      */
    </style>

    <script src="http://code.jquery.com/jquery.min.js"></script>
    <script src="js/foundation/foundation.js"></script>
    <script src="js/foundation/foundation.slider.js"></script>

    <!-- Script for sliders-->
    <link rel="stylesheet"
href="//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css">
    <script src="//code.jquery.com/jquery-1.10.2.js"></script>
    <script src="//code.jquery.com/ui/1.11.4/jquery-ui.js"></script>
    <link rel="stylesheet" href="/resources/demos/style.css">

    <script>

      // log function
      log = function(data){
        $("div#terminal").prepend("</br>" +data);
        console.log(data);
      };

      $(document).ready(function () {
```

```
$("div#message_details").hide()

var ws;

$("#open").click(function(evt) {
  evt.preventDefault();

  var host = $("#host").val();
  var port = $("#port").val();
  var uri = $("#uri").val();

  // create websocket instance
  ws = new WebSocket("ws://" + host + ":" + port + uri);

  // Handle incoming websocket message callback
  // *will only print if self.write_message() is in python code
  ws.onmessage = function(evt) {
    log("Message Received: " + evt.data)
    };

  // Close Websocket callback
  ws.onclose = function(evt) {
    log("***Connection Closed***");
    alert("Connection close");
    $("#host").css("background", "#ff0000");
    $("#port").css("background", "#ff0000");
    $("#uri").css("background",  "#ff0000");
    $("div#message_details").empty();
    };

  // Open Websocket callback
  ws.onopen = function(evt) {
    $("#host").css("background", "#00ff00");
    $("#port").css("background", "#00ff00");
    $("#uri").css("background", "#00ff00");
    $("div#message_details").show();
    log("***Connection Opened***");
  };
});

$("#close").click(function(evt) {
    ws.close();
});

/*Value Changed for RED GREEN and BLUE for K64F*/
$("#red").click(function(evt) {
    log("Sending Message: "+$("#red").val());
    ws.send($("#red").val());
});

$("#blue").click(function(evt) {
    log("Sending Message: "+$("#blue").val());
    ws.send($("#blue").val());
});
```

```
        $("#green").click(function(evt) {
            log("Sending Message: "+$("#green").val());
            ws.send($("#green").val());
        });

        $("#btn1").click(function(evt) {
            //log("Sending Message: "+$("#green").val());
            ws.send($("#btn1").val());
        });

        $("#btn2").click(function(evt) {
            //log("Sending Message: "+$("#green").val());
            ws.send($("#btn2").val());
        });

        $("#qty").click(function(evt) {
            //log("Sending Message: "+(typeof $("#qty").val()));
            ws.send($("#qty").val());
        });

        $("#send").click(function(evt) {
            ws.send($("#qty").val());
        });

        $("#slider").on("input change", function(){
            ws.send($("#slider").val());
        });

      });

    </script>
  </head>

  <body>
    <h1>WebSockets Hello World</h1>
    <div id="connection_details">
      <label for="host">host:</label>
      <input type="text" id="host" value="localhost"
style="background:#ff0000;"/><br />
      <label for="port">port:</label>
      <input type="text" id="port" value="8888"
style="background:#ff0000;"/><br />
      <label for="uri">uri:</label>
      <input type="text" id="uri" value="/ws"
style="background:#ff0000;"/><br />
      <input type="submit" id="open" value="open" /></br>
      <input type="submit" id="close" value="close" />
    </div>

    <!--
    <div id="message_details">
        </br></br>
        <label for="message">message:</label>
        <input type="text" id="message" value="Hello World!"/><br />
```

```
            <input type="submit" id="send" value="send" />
        </div>
        -->

        <div id="led_buttons">

            <!-- NOTE: Values changed for K64F 6/1/15 (myled1 = red, myled2
= blue)-->
            </br>
            <h2>Select the LED you wish to turn on</h2>
            <!-- REMOVED FOR PWM K64F LEDs
            <button type="button" id="red" onclick="console.log('red')"
value="myled1">Red/LED1</button>
            <button type="button" id="green" onclick="console.log('green')"
value="myled2">Green/LED2</button>
            <button type="button" id="blue" onclick="console.log('blue')"
value="myled3">Blue/LED3</button>
            </br></br>
            -->

            <button type="button" id="btn1" onclick="console.log('btn1')"
value="btn1">Light 1</button>
            <button type="button" id="btn2" onclick="console.log('btn2')"
value="btn2">Light 2</button>
            </br>

            <h3>LED Slider</h3>
            <input type="range" orient="vertical" id="slider" step=".01"
min="0"  max="1">
        </div>

        <div id="terminal">
        </div>

    </body>
</html>
```

Appendix G – mult_led_server.py

```
#!/usr/bin/env python

import tornado.httpserver
import tornado.websocket
import tornado.ioloop
import tornado.web
import socket

class WSHandler(tornado.websocket.WebSocketHandler):
    ws_clients = []
    num = 0
    num_red = 0
    num_green = 0
    num_blue = 0

    button = 10

    def open(self):
        if self not in WSHandler.ws_clients:
            WSHandler.ws_clients.append(self)
        print 'new connection'

    def on_message(self, message):
        for c in WSHandler.ws_clients:
            name = str(message)

            c.write_message(name)
            print "Incoming message: " + name
            #print "Message type: " + str(type(message))
            print "Button Value: " + str(WSHandler.button)


            ## Selection of LED By button

            if name == "btn1":
                WSHandler.button = 1
            elif name == "btn2":
                WSHandler.button = 2
            elif name == "btn3":
                WSHandler.button = 3
            ## Writes only numerical values when slider is moved

            if (WSHandler.button == 1 and name !="btn1"):
                ext_led1.write(name)
                #c.write_message(name)
            if (WSHandler.button == 2 and name !="btn2"):
                ext_led2.write(name)
                #c.write_message(name)

    def on_close(self):
        if self in WSHandler.ws_clients:
```

```
            WSHandler.ws_clients.remove(self)
        print 'connection closed'

    def check_origin(self, origin):
        return True


from mbedRPC_new import *
#serdev = 2 # For LPC1768
serdev = 3 # For k64F (4 MB/3 MB Pro)
mbed = SerialRPC(serdev, 9600)
print 'Load mbedRPC'

#Turn on/off LED
#myled1 = RpcDigitalOut(mbed, "myled1")
#myled2 = RpcDigitalOut(mbed, "myled2")
#myled3 = RpcDigitalOut(mbed, "myled3")
ext_led1 = RpcPwmOut(mbed, "ext_led1")
ext_led2 = RpcPwmOut(mbed, "ext_led2")

application = tornado.web.Application([
    (r'/ws', WSHandler),
])

if __name__ == "__main__":
    http_server = tornado.httpserver.HTTPServer(application)
    http_server.listen(8888)
    myIP = socket.gethostbyname(socket.gethostname())
    print '*** Websocket Server Started at %s***' % myIP
    tornado.ioloop.IOLoop.instance().start()
```

Appendix H – /RPC_Serial_LED/main.cpp

```
#include "mbed.h"
#include "mbed_rpc.h"

RpcDigitalOut myled1(LED1,"myled1");
RpcDigitalOut myled2(LED2,"myled2");
RpcDigitalOut myled3(LED3,"myled3");
RpcDigitalOut myled4(LED4,"myled4");

RpcPwmOut ext_led1(p21, "ext_led1");
RpcPwmOut ext_led2(p22, "ext_led2");

Serial pc(USBTX, USBRX);
int main() {

    // receive commands, and send back the responses
    char buf[256], outbuf[256];
    while(1) {
        pc.gets(buf, 256);
        //Call the static call method on the RPC class
        RPC::call(buf, outbuf);
        pc.printf("%s\n", outbuf);
    }
}
```

Appendix I – ws_server.py

```
#!/usr/bin/env python

import tornado.httpserver
import tornado.websocket
import tornado.ioloop
import tornado.web
import socket

class WSHandler(tornado.websocket.WebSocketHandler):
    ws_clients = []
    def open(self):
        if self not in WSHandler.ws_clients:
            WSHandler.ws_clients.append(self)
        print 'new connection'

    def on_message(self, message):
        for c in WSHandler.ws_clients:
            name = str(message)

            c.write_message(name)
            print "Incoming message: " + name
            #print "Message type: " + str(type(message))
            #print "Button Value: " + str(WSHandler.button)

    def on_close(self):
        if self in WSHandler.ws_clients:
            WSHandler.ws_clients.remove(self)
        print 'connection closed'

    def check_origin(self, origin):
        return True

application = tornado.web.Application([
    (r'/ws', WSHandler),
])

if __name__ == "__main__":
    http_server = tornado.httpserver.HTTPServer(application)
    http_server.listen(8888)
    myIP = socket.gethostbyname(socket.gethostname())
    print '*** Websocket Server Started at %s***' % myIP
    tornado.ioloop.IOLoop.instance().start()
```

Appendix J - /FRDM_WS_Eth_Ctrl/main.cpp

```
#include "mbed.h"
#include "EthernetInterface.h"
#include "Websocket.h"

Serial pc(USBTX, USBRX);
//BusOut l(LED1, LED2, LED3, LED4);
PwmOut led(PTD1);

//websocket: configuration
Websocket ws("ws://**.**.***.**:****/ws");

int main() {
    EthernetInterface eth;
    eth.init(); //Use DHCP
    eth.connect();
    printf("IP Address is %s\n\r", eth.getIPAddress());
    char recv[] = "";

    ws.connect();
    while (1) {
        if(ws.read(recv)){
            printf("Received message: %s\r\n", recv);
            float temp = atof(recv);
            led = temp;
        }
    }
}
```

Appendix K - /Motordriver_HelloWorld_Serial/main.cpp (Modified from (ARMmbed,n.d.d))

```
#include "mbed.h"
#include "motordriver.h"

Serial pc(USBTX, USBRX); // tx, rx
Motor m(p23, p6, p5, 1); // pwm, fwd, rev

int main() {
    int val = 0;
    pc.printf("\rPress 'u' to move motor fwd up, 'd' to reverse\n\r");
    float s;
    while(1) {
        char c = pc.getc();
        if(c == 'u'){
            s = 1.00;
            m.speed(s);
            wait(.02);
            m.stop(.1);
            val++;
            pc.printf("\rval = %d\n", val);
        }
        if(c == 'd'){
            s=-1.0;
            m.speed(s);
            wait(.02);
            m.stop(.1);
            val--;
            pc.printf("\rval = %d\n", val);
        }
    }
}
```

Appendix L – Additional Rotating Knob Snippet (Terrien, 2015)

```
$(function() {
	$(".dial").knob({
		"min": 1,
		"max": 100,
		"angleArc": 360,
		"angleOffset": -155,
		"width": 200,
		"height": 200,
		"thickness": .85,
		"stopper": "false",
		"displayInput": false,
		"fgColor": "black",
		"bgColor": "grey",
		"cursor": 3,
		"change": function (v) {ws.send(v);}
	});
});
```

Appendix M - /FRDM_WS_Motordriver/main.cpp

```
#include "mbed.h"

#include <string>

#include "EthernetInterface.h"

#include "Websocket.h"

#include "motordriver.h"

#include "math.h"


Serial pc(USBTX, USBRX);

PwmOut led(PTD1);

Motor m(PTC10, PTB2, PTB3, 1); // pwm, fwd, rev


//websocket: configuration

Websocket ws("ws://**.**.**.***:****/ws"); // ARU

//Websocket ws("ws://***.***.*.***:****/ws"); //Los Angeles

//Websocket ws("ws://***.**.**.**:****/ws"); // Railyard


int main() {

    EthernetInterface eth;

    float temp = 0;

    float s = 1.00;

    m.speed(s);

    wait(1);


    eth.init(); //Use DHCP

    eth.connect();

    printf("IP Address is %s\n\r", eth.getIPAddress());

    char recv[] = "";
```

```
    ws.connect();
    ws.send(eth.getIPAddress());

    while (1) {
        if(ws.read(recv)){
            printf("Received message: %s\r\n", recv);
            float val = atof(recv);
            if (val > temp){
                s = 1.00;
                m.speed(s);
                wait(.018);
                m.stop(.9);
                temp = val;
            }
            else if (val < temp){
                s = -1.00;
                m.speed(s);
                wait(.018);
                m.stop(.9);
                temp = val;
            }
        }
    }
}
```

Appendix N – Initial Jacktrip Traceroutes

LA (Residence) -> UK (ARU)

```
Marquess-MBP:~ marqueshardin$ traceroute 193.63.211.160
traceroute to 193.63.211.160 (193.63.211.160), 64 hops max, 52 byte packets
 1  dsldevice (192.168.1.254)  1.655 ms  0.975 ms  1.441 ms
 2  99-44-60-3.lightspeed.irvnca.sbcglobal.net (99.44.60.3)  467.130 ms  448.748 ms
457.020 ms
 3  75.20.1.10 (75.20.1.10)  449.552 ms *  453.583 ms
 4  12.83.38.149 (12.83.38.149)  453.707 ms * *
 5  ggr2.la2ca.ip.att.net (12.122.129.97)  452.661 ms  453.441 ms  456.557 ms
 6  * * *
 7  * * *
 8  * 213.155.135.68 (213.155.135.68)  589.469 ms
    ldn-bb2-link.telia.net (213.248.65.93)  600.145 ms
 9  * * *
10  * * *
11  * ae29.londpg-sbr2.ja.net (146.97.33.2)  596.852 ms  613.129 ms
12  ae30.londtw-sbr2.ja.net (146.97.33.6)  595.448 ms  603.582 ms  607.456 ms
13  146.97.38.18 (146.97.38.18)  600.065 ms * *
14  * * *
15  * * *
16  dmz160.dante.org.uk (193.63.211.160)  615.553 ms * *
```

UK (ARU) -> LA (Residence)

```
fst-cat-mbpr-domenico:~ domenico$ traceroute 99.44.63.47
traceroute to 99.44.63.47 (99.44.63.47), 64 hops max, 52 byte packets
 1  193.63.211.129 (193.63.211.129)  2.508 ms  1.354 ms  1.466 ms
 2  ge-11-0-8.camb-rbr1.eastern.ja.net (146.97.130.37)  2.164 ms  2.309 ms
1.884 ms
 3  146.97.65.118 (146.97.65.118)  2.189 ms  15.424 ms  46.655 ms
 4  146.97.38.17 (146.97.38.17)  5.110 ms  5.000 ms  5.042 ms
 5  ae29.londtn-sbr1.ja.net (146.97.33.10)  5.136 ms  7.947 ms  5.179 ms
 6  lag-108.ear1.london15.level3.net (212.187.173.53)  5.408 ms  5.397 ms
5.508 ms
 7  * * *
 8  att-level3-60g.nyc.level3.net (4.68.63.142)  76.928 ms  75.278 ms  74.345 ms
 9  cr1.n54ny.ip.att.net (12.122.131.102)  154.354 ms  154.905 ms  155.959 ms
10  cr2.cgcil.ip.att.net (12.122.1.2)  325.549 ms  153.992 ms  157.568 ms
11  cr1.cgcil.ip.att.net (12.122.2.53)  156.911 ms  154.955 ms  154.877 ms
12  cr2.dvmco.ip.att.net (12.122.31.85)  166.436 ms  154.428 ms  153.916 ms
13  cr1.slkut.ip.att.net (12.122.30.25)  154.724 ms  155.228 ms  155.779 ms
14  * * *
15  gar5.lsrca.ip.att.net (12.122.85.37)  153.039 ms  159.027 ms  155.523 ms
16  * * *
17  75.20.1.9 (75.20.1.9)  154.514 ms *  154.613 ms
```

LA (USC - Internet2) -> UK (ARU)

```
traceroute to 193.63.211.169 (193.63.211.169), 64 hops max, 52 byte packets
 1  guest-wireless-upc-1605-gw-13 (10.120.80.2)  3.163 ms  3.141 ms  1.938 ms
 2  130.152.183.2 (130.152.183.2)  4.204 ms  4.322 ms  10.610 ms
 3  dc-riv-agg4--losnettos-10ge.cenic.net (137.164.23.193)  10.375 ms  10.145 ms  13.825
ms
 4  dc-lax-agg7--riv-agg4-100ge.cenic.net (137.164.11.2)  7.646 ms  7.208 ms  8.181 ms
 5  dc-lax-agg6--lax-agg7-100g.cenic.net (137.164.11.10)  8.753 ms  11.327 ms  8.244 ms
 6  hpr-lax-hpr--lax-agg6.cenic.net (137.164.26.245)  8.076 ms  41.513 ms  8.872 ms
 7  hpr-i2--lax-hpr2-r&e.cenic.net (137.164.26.201)  8.855 ms  7.257 ms  8.043 ms
 8  et-1-0-0.111.rtr.hous.net.internet2.edu (198.71.45.20)  39.222 ms  39.305 ms  44.419 ms
 9  et-10-0-0.105.rtr.atla.net.internet2.edu (198.71.45.12)  63.858 ms  63.677 ms  85.863 ms
10  et-9-0-0.104.rtr.wash.net.internet2.edu (198.71.45.7)  76.587 ms  77.426 ms  81.315 ms
11  internet2-gw.mx1.lon.uk.geant.net (62.40.124.44)  154.225 ms  154.885 ms  152.738 ms
12  oc-router-gw-geant.geant.net (62.40.125.250)  160.702 ms  159.227 ms  160.854 ms
13  * * *
14  dmz169.dante.org.uk (193.63.211.169)  171.240 ms  165.911 ms  167.980 ms
guest-wireless-upc-1605-10-120-085-107
```

UK (ARU) -> LA (USC - Internet2)

```
fst-cat-mbpr-domenico:~ domenico$ traceroute 207.151.35.21
traceroute to 207.151.35.21 (207.151.35.21), 64 hops max, 52 byte packets
 1  193.63.211.129 (193.63.211.129)  1.474 ms  1.633 ms  3.852 ms
 2  ge-11-0-8.camb-rbr1.eastern.ja.net (146.97.130.37)  2.379 ms  1.859 ms  1.929 ms
 3  146.97.65.118 (146.97.65.118)  1.847 ms  2.132 ms  2.364 ms
 4  146.97.38.17 (146.97.38.17)  7.737 ms  5.154 ms  5.041 ms
 5  ae30.londpg-sbr2.ja.net (146.97.33.5)  5.997 ms  5.676 ms  7.772 ms
 6  ae29.londhx-sbr1.ja.net (146.97.33.1)  6.077 ms  5.769 ms  7.407 ms
 7  janet.mx1.lon.uk.geant.net (62.40.124.197)  9.267 ms  10.068 ms  9.822 ms
 8  internet2-gw.mx1.lon.uk.geant.net (62.40.124.45)  83.169 ms  83.813 ms  83.192 ms
 9  et-5-0-0.104.rtr.atla.net.internet2.edu (198.71.45.6)  96.442 ms  96.101 ms  96.437 ms
10  et-10-2-0.105.rtr.hous.net.internet2.edu (198.71.45.13)  120.563 ms  126.958 ms  124.625 ms
11  et-7-1-0.4070.rtsw.losa.net.internet2.edu (198.71.45.21)  153.109 ms  153.498 ms  152.665 ms
12  hpr-lax-hpr2--i2-r&e.cenic.net (137.164.26.200)  153.347 ms  153.440 ms  153.170 ms
13  hpr-lax-agg6--lax-hpr.cenic.net (137.164.26.246)  153.544 ms  153.713 ms  154.697 ms
14  lax-agg6--losnettos-dc.cenic.net (137.164.23.226)  157.267 ms  157.406 ms  157.021 ms
15  130.152.181.189 (130.152.181.189)  157.094 ms
    ln-usc3-riv2015.ln.net (130.152.181.145)  157.883 ms
    ln-usc-gsr-vlan2005.ln.net (130.152.181.73)  157.360 ms
16  * * *
17  *
```

Appendix O – broadcast.html (Modified from Gleason (2015))

```
<!DOCTYPE html>

<html>
<head>
    <title>WebRTC Vid Stream</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"
/>
    <meta name= "viewport" content="width=device-width, initial-scale=1"
/>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"><
/script>
    <script src="https://cdn.pubnub.com/pubnub-3.7.14.min.js"></script>
    <script src="https://cdn.pubnub.com/webrtc/webrtc.js"></script>
    <script src="https://cdn.pubnub.com/webrtc/rtc-
controller.js"></script>
</head>

<style>
      #watch{
            display: none;
      }
      #inStream{
            display: none;
      }
</style>

<body>

      <div id="vid-box"><!-- Stream goes here --></div>

      <form name="streamForm" id="stream" action="#" onsubmit="return
stream(this);">
            <input type="text" name="streamname" id="streamname"
placeholder="Pick a stream name!" />
            <input type="submit" name="stream_submit" value="Stream">
            </form>

      <form name="watchForm" id="watch" action="#" onsubmit="return
watch(this);">
            <div id="stream-info">Watching: <span id="here-
now">0</span></div>
            <input type="text" name="number" placeholder="Enter stream
to join!" />
            <input type="submit" value="Watch"/>
      </form>

      <div id="inStream">
```

```
            <button id="end" onclick="end()">Done</button> <br>
            Generate Embed: <button
onclick="genEmbed(400,600)">Tall</button><button
onclick="genEmbed(600,400)">Wide</button><button
onclick="genEmbed(500,500)">Square</button><br>
            <div id="embed-code"></div>
      </div>

      <br/>
      <button id="mute" onclick="mute()">Mute</button>

    <script>
        var video_out  = document.getElementById("vid-box");
            var embed_code = document.getElementById("embed-code");
            var here_now   = document.getElementById('here-now');
            var streamName;

            function stream(form) {
                  streamName = form.streamname.value ||
Math.floor(Math.random()*100)+''; // Random stream if not provided
                  var phone = window.phone = PHONE({
                        number        : streamName, // listen on
username else random
                        publish_key   : 'pub-c-********-****-****-****-
a8e8cc477574', // Your Pub Key
                        subscribe_key : 'sub-c-********-****-****-****-
02ee2ddab7fe', // Your Sub Key
                        oneway        : true,   // One-Way streaming
enabled
                        broadcast     : true,   // True since you are the
broadcaster
                        media         : {audio : false, video : true}
                  });

                  var ctrl = window.ctrl = CONTROLLER(phone);
                  ctrl.ready(function(){
                        form.streamname.style.background="#55ff5b";
                        form.stream_submit.hidden="true";
                        ctrl.addLocalStream(video_out);
                        ctrl.stream();    // Begin streaming video
                  });

                  ctrl.streamPresence(function(m){
here_now.innerHTML=m.occupancy; });
                  return false;  // So form does not submit
            }

            function watch(form){
                  var num = form.number.value;  // Stream to join
                  var phone = window.phone = PHONE({
                        number        : "Viewer" +
Math.floor(Math.random()*100), // Random name
                        publish_key   : 'pub-c-********-****-****-****-
a8e8cc477574', // Your Pub Key
```

```
                        subscribe_key : 'sub-c-********-****-****-****-
02ee2ddab7fe', // Your Sub Key
                        oneway        : true,   // One way streaming
enabled
                        media         : {audio : false, video : true}
                  });

                  var ctrl = window.ctrl = CONTROLLER(phone, true);
                  ctrl.ready(function(){
                        ctrl.isStreaming(num, function(isOn){
                              //if (isOn)
                              ctrl.joinStream(num);
                              //else alert("User is not streaming!");
                        });
                  });
                  ctrl.receive(function(session){
                        session.connected(function(session){
                              video_out.appendChild(session.video);
                        });
                  });
                  ctrl.streamPresence(function(m){
                        here_now.innerHTML=m.occupancy;
                  });
                  return false;  // Prevent form from submitting
            }

            function genEmbed(w,h){
                  if (!streamName) return;  // If global var not set,
not streaming
                  var url = "http://<your-webstie>/embed.html?stream=" +
streamName;
                  var embed    = document.createElement('iframe');
                  embed.src    = url;
                  embed.width  = w;
                  embed.height = h;
                  embed.setAttribute("frameborder", 0);
                  embed_code.innerHTML = 'Embed Code: ';

      embed_code.appendChild(document.createTextNode(embed.outerHTML));
            }
            function mute(){
                  var audio = ctrl.toggleAudio();
                  if (!audio) $("#mute").html("Unmute");
                  else $("#mute").html("Mute");
            }

    </script>
</body>
</html>
```

Appendix P – embed.html (Modified from Gleason (2015))

```
<!DOCTYPE html>

<html>
<head>
    <title>WebRTC Vid Stream</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"
/>
    <meta name= "viewport" content="width=device-width, initial-scale=1"
/>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"><
/script>
    <script src="https://cdn.pubnub.com/pubnub-3.7.14.min.js"></script>
    <script src="https://cdn.pubnub.com/webrtc/webrtc.js"></script>
    <script src="https://cdn.pubnub.com/webrtc/rtc-
controller.js"></script>


      <style>
            #vid-box{
            width: 100%;
            height: 100%;
            text-align: center;
            }

            #vid-box video{
            width: 100%;
            height: 100%;
            }

      #stream-info{
            position: absolute;
            bottom: 3vh;
            right: 5vw;
      }
      </style>

</head>

<body>
      <div id="vid-box"></div>
      <div id="stream-info"><span id="here-now"></span></div>

      <script src="https://cdn.pubnub.com/pubnub.min.js"></script>
      <script
src="http://kevingleason.me/SimpleRTC/js/webrtc.js"></script>
      <script src="http://kevingleason.me/SimpleRTC/js/rtc-
controller.js"></script>
```

```
    <script>
        (function() {

                var urlargs     = urlparams();
                var video_out   = document.getElementById("vid-box");
                var stream_info = document.getElementById("stream-
info");
                var here_now    = document.getElementById("here-now");

                // Handle error if stream is not in urlargs.

                if (!('stream' in urlargs)) {
                      handleNoStream();
                      return;
                }

                // Get URL params
                function urlparams() {
                      var params = {};
                      if (location.href.indexOf('?') < 0){
                            return params;
                      }

                      PUBNUB.each(
                            location.href.split('?')[1].split('&'),
                            function(data) { var d = data.split('=');
params[d[0]] = d[1]; }
                      );
                      return params;
                }

                function handleNoStream(){
                      video_out.innerHTML="<h2>That stream no longer
exists!</h2>";
                      stream_info.hidden=true;
                }

                var stream = urlargs.stream;

                var phone = window.phone = PHONE({
                      number        : "EmbedViewer" +
Math.floor(Math.random()*100), // random viewer name
                      publish_key   : 'pub-c-********-****-****-****-
a8e8cc477574', // Your Pub Key
                      subscribe_key : 'sub-c-********-****-****-****-
02ee2ddab7fe', // Your Sub Key
                      oneway        : true,
                      media         : {audio : false, video : true}
                });

                var ctrl = window.ctrl = CONTROLLER(phone);
                ctrl.ready(function(){
                      ctrl.isStreaming(stream, function(isOn){
```

```
                        //if (isOn)
                        ctrl.joinStream(stream);
                        //else handleNoStream();
                  });
            });

            ctrl.receive(function(session){
                  session.connected(function(session){
                        stream_info.hidden=false;
                        video_out.appendChild(session.video);
                  });
                  session.ended(function(session){
                        handleNoStream();
                  });
            });

            ctrl.streamPresence(function(m){
                  here_now.innerHTML = m.occupancy;
            });

            ctrl.unable(function(){ handleNoStream(); });

      }());
  </script>

</body>
</html>
```

Appendix Q – MATLAB Script to Generate Pure 1 kHz Sine Wave

```
clc;
clear all;

Amp=.25;
Fs=44100;
ts=44100/44100000;
tf=0:1/Fs:10-1/Fs;
y=Amp*sin(2*pi*1000*tf);
plot(tf,y);

yd=length(y)/Fs;
t=linspace(0,yd,length(y));


figure % plot waveform
plot(t,y,'b');
title('pure sine');
ylabel('16-bit data');
xlabel('Time, s')
axis([0 yd -1.1 1.1])

figure %plot spectrogram
F=[0:10:6000]; % frequencies for which to calculate spectrogram for
S = abs(spectrogram(y,8192,512,F,Fs));
[r,c] = size(S);     T = [0:c]*t(end)/c;
imagesc(T,F,20*log10(S));
axis xy
xlabel('Time, s')
ylabel('Frequency, Hz')

% Write Wav
filename = 'pure_1k_sine.wav';
audiowrite(filename,y,Fs);
```

Appendix R – MATLAB Script to Analyse Audio Waveform and Spectrogram

```
% load wav file and show data, spectrogram and fft
% only works for mono data

clear all

[filename, pathname] = uigetfile('*.wav','Open file');

if filename~=0

    wavfile = [pathname filename];
    [y,Fs]=audioread(wavfile);

    yd=length(y)/Fs;                   % yd = y duration
    t=linspace(0,yd,length(y));

    Fres = Fs*8;           %freq resolution 8times 0 padding
    Frange=round(Fres/2);
    Faxis=linspace(0,Fs,Fres+1);

    hanning_window = hann(length(y));
    y_hann=y.*hanning_window;
    y_fft_raw=abs(fft(y_hann,Fres));     % calculate fft
    y_fft=y_fft_raw/max(y_fft_raw); % normalise

    figure % plot waveform
    plot(t,y,'b');
    title(filename);
    ylabel('16-bit data');
    xlabel('Time, s')
    axis([0 yd -1.1 1.1])

    figure %plot spectrogram
    F=[0:10:6000]; % frequencies for which to calculate spectrogram for
    %F=[0:10:22000];
    S = abs(spectrogram(y,8192,512,F,Fs));
    [r,c] = size(S);     T = [0:c]*t(end)/c;
    imagesc(T,F,20*log10(S));
    axis xy
    xlabel('Time, s')
    ylabel('Frequency, Hz')

    figure % plot fft
    plot(Faxis(1:Frange),20*log10(y_fft(1:Frange)),'b')
    xlabel('Frequency, Hz')
    ylabel('Normalised FFT power, dB')
    axis([0 20000 -85 6]);

end
```

Appendix S – MATLAB Script to Calculate Packet Dropouts

```
% load wav file and show data, spectrogram and fft
% only works for mono data

clear all

[filename, pathname] = uigetfile('*.wav','Open file');

if filename~=0

    wavfile = [pathname filename];
    [y,Fs]=audioread(wavfile);

    yd=length(y)/Fs;                    % yd = y duration
    t=linspace(0,yd,length(y));
    sinewave=sin(2*pi*t*1000);
    figure
    plot(t*1000,sinewave,'-*')
    ylabel('Normalised 16-bit data');
    xlabel('Time, ms')

    x=0;
    maxd=0.15*max(y); % although max is 0.1425, use 0.15 to allow for
potential rounding errors
    droptime=[];
    y0=y(1);
    for i=1:length(y)
        if abs(y(i)-y0)>maxd;
            x=x+1;
            droptime(x)=t(i);  % array of dropout times in the data
        end
        y0=y(i);
    end

    drops=length(droptime) % total number of identified dropouts


    figure % plot waveform
    plot(t,y,'b');
    title(filename);
    ylabel('16-bit data');
    xlabel('Time, s')
    axis([0 yd -1.1 1.1])

end
```

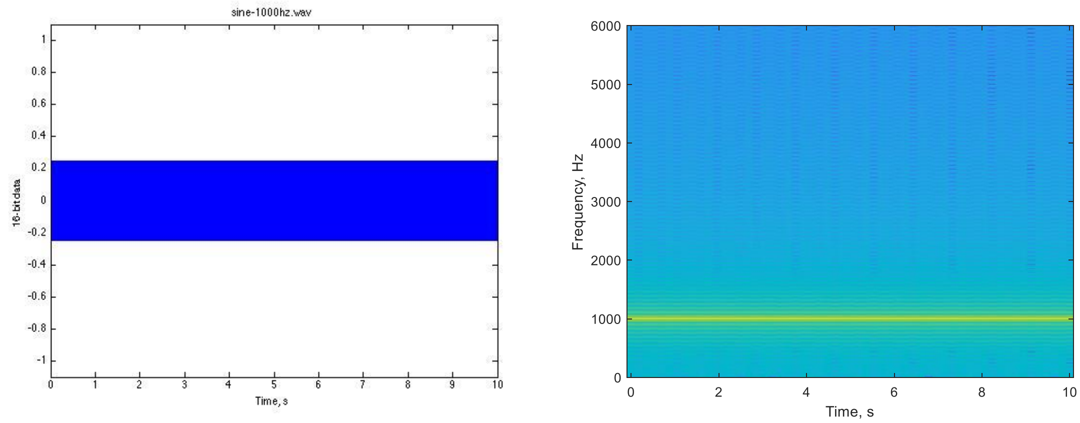
Appendix T – MATLAB Script to Calculate THD+N Measurements

```
% load wav file and calculate THD+N
% only works for mono data

clear all

[filename, pathname] = uigetfile('*.wav','Open file');

if filename~=0


    wavfile = [pathname filename];
    %[y,Fs]=wavread(wavfile);
    [y,Fs]=audioread(wavfile);

    yd=length(y)/Fs;                  % yd = y duration
    t=linspace(0,yd,length(y));

    % test pure simewave to verify (uncomment to implement)
    %ys = sin(2*pi*1000*t);
    %y=ys';

    Fres = Fs*16;           %freq resolution 16 times 0 padding
    Frange=round(Fres/2);
    Faxis=linspace(0,Fs,Fres+1);

    hanning_window = hann(length(y));
    y_hann=y.*hanning_window;
    y_fft_raw=abs(fft(y_hann,Fres)); % calculate fft
    y_fft=y_fft_raw/max(y_fft_raw);  % normalise

    % filter in frequency domain (brick wall)
    y_fft_filtered=y_fft;
    fft_length=round(length(y_fft)/2);
    y_filterprofile(1:length(y_fft_filtered))=1; % profile data for plot

    % remove below 22Hz
    filt_22hz=round(22*Fres/Fs);
    for i=1:filt_22hz
        y_fft_filtered(i)=0;
        y_filterprofile(i)=0.000001;
    end

    % remove above 22kHz
    filt_22khz=round(22000*Fres/Fs);
    for i=filt_22khz:length(y_fft_filtered)
        y_fft_filtered(i)=0;
        y_filterprofile(i)=0.000001;   % non-zero to enable plot
    end

    % remove notch between 900 Hz - 1100 Hz
    filt_notchstart=round(900*Fres/Fs);
```

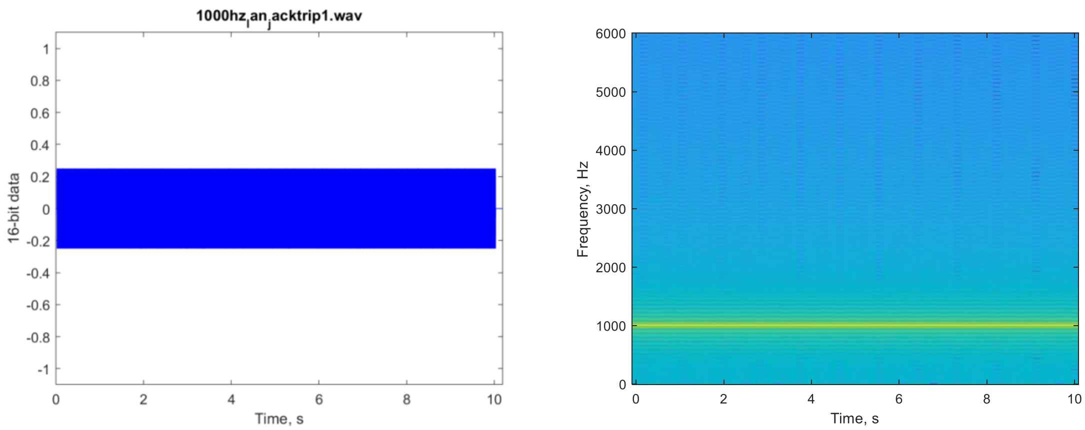
```
    filt_notchend=round(1100*Fres/Fs);

    y_fft_notched=y_fft_filtered;

    for i=filt_notchstart:filt_notchend
       y_fft_notched(i)=0;
       y_filterprofile(i)=0.00001;     % non-zero to enable plot
    end

    %find power of the fft and the filtered fft (sum of squares)
    fftpower=0;
    distpower=0;
    for i=1:fft_length
        fftpower=fftpower+(y_fft_filtered(i)^2);
        distpower=distpower+(y_fft_notched(i)^2);
    end

    fftpowermean=fftpower/fft_length;
    distpowermean=distpower/fft_length;

    %calculate ratio by two methods (both give same answer)
    THDN_fft=100*(distpowermean^0.5)/(fftpowermean^0.5) % fft rms
    yfiltered=ifft(y_fft_filtered); % inverse fft
    ynotched=ifft(y_fft_notched);   % inverse fft
    THDN_t=100*rms(ynotched)/rms(yfiltered)  % time domain rms

    h=figure; % plot fft and filter window;
    axes1 = axes('Parent',h,'XScale','log','XMinorTick','on');
    hold on
    semilogx(Faxis(1:Frange),20*log10(y_fft(1:Frange)),'b')
    semilogx(Faxis(1:Frange),20*log10(y_filterprofile(1:Frange)),'r')
    xlabel('Frequency, Hz')
    ylabel('Normalised FFT power, dB')
    axis([20 23000 -85 6]);

end
```

Appendix U – JackTrip LAN Audio Captures

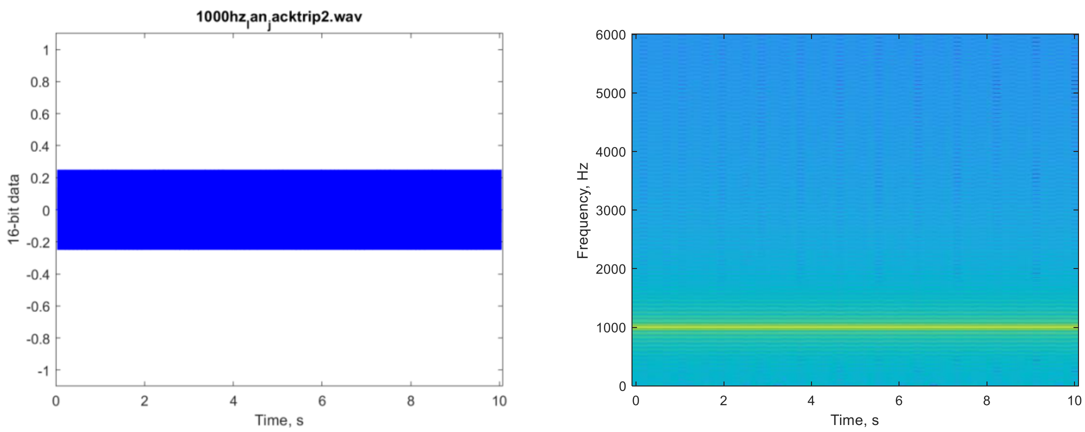
1 kHz Sine Wave



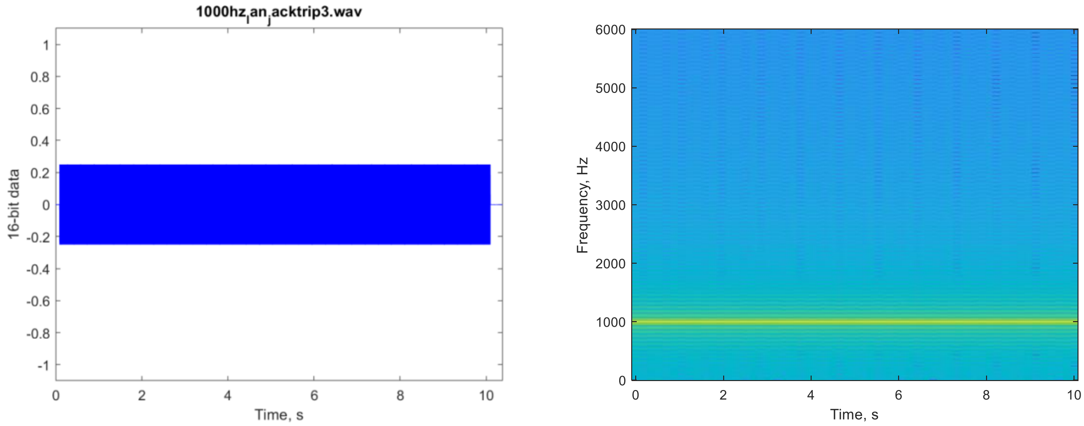
Source Audio



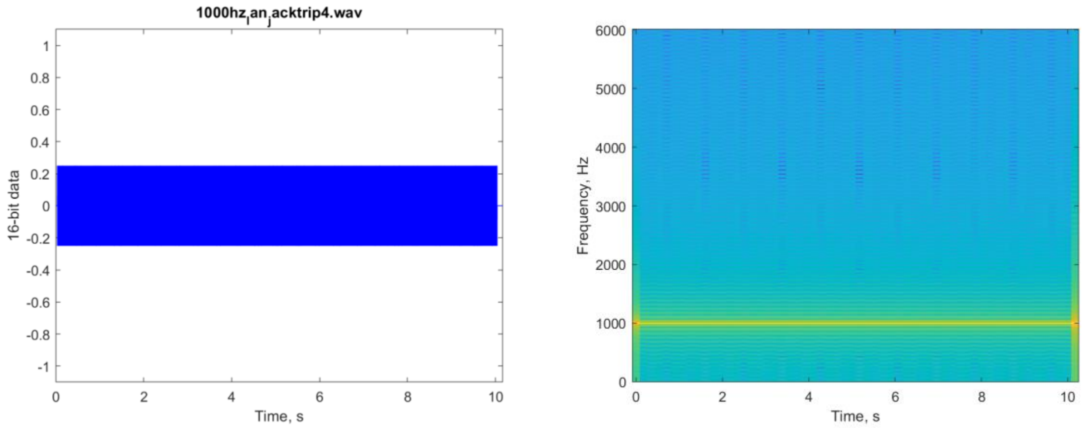
Streaming Capture 1



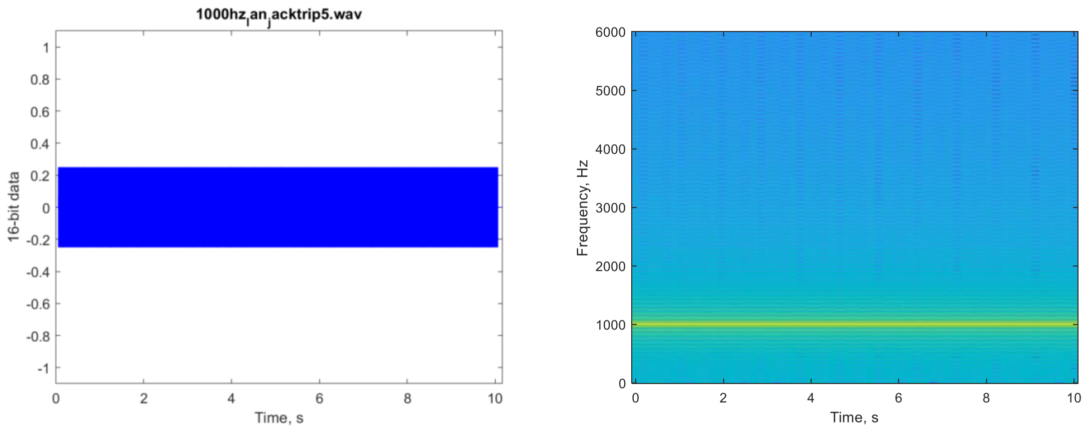
Streaming Capture 2



Streaming Capture 3

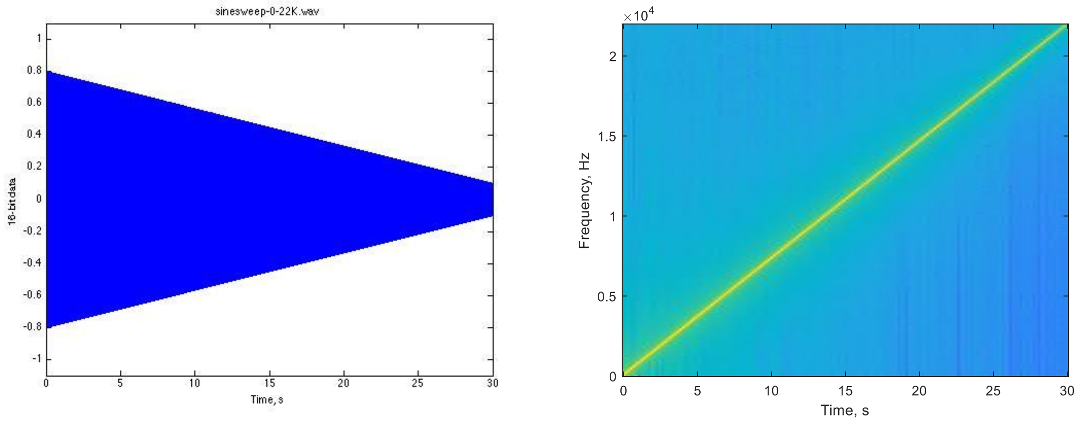


Streaming Capture 4

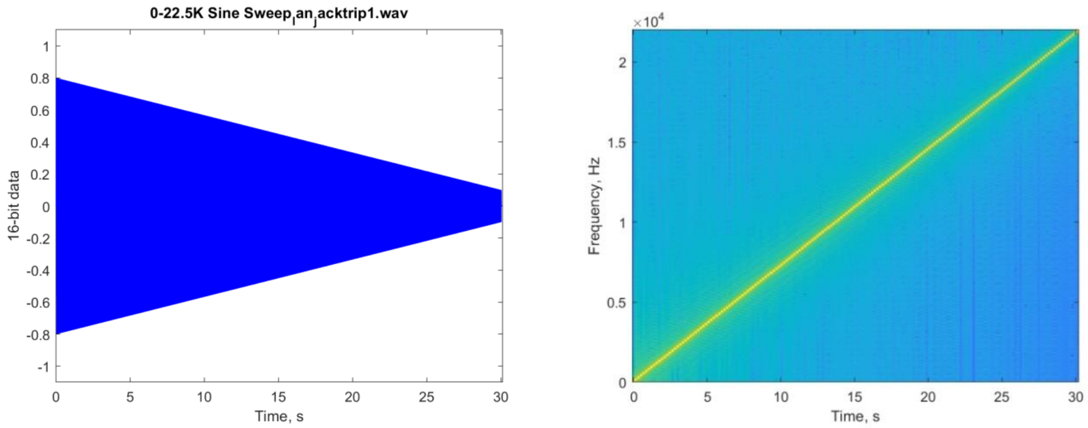


Streaming Capture 5

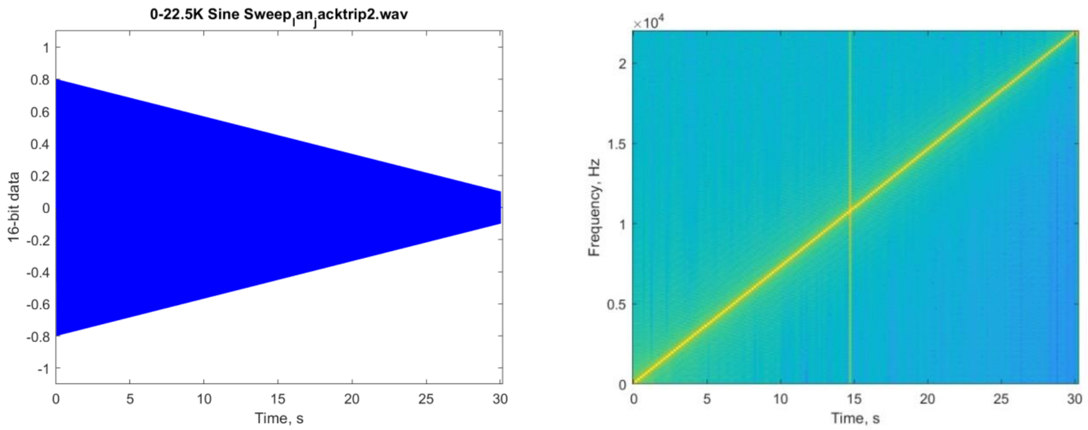
0-22.5 kHz Sine Sweep



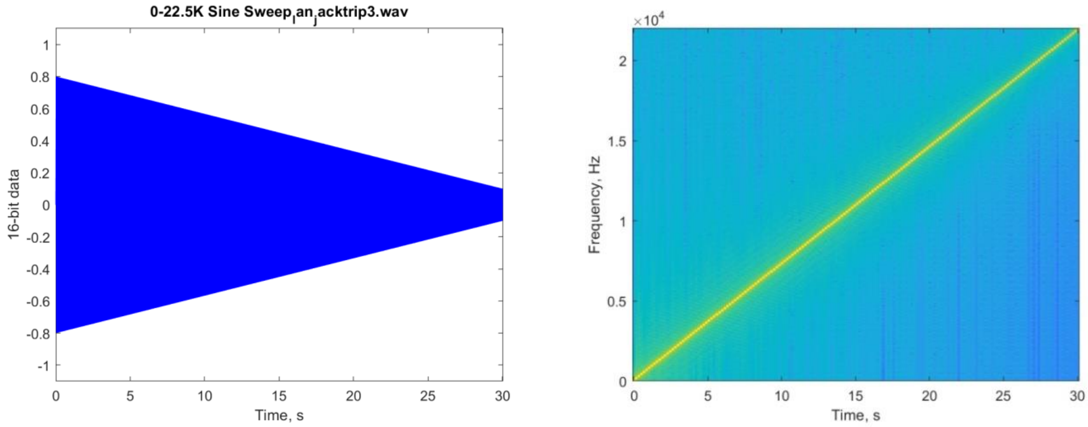
Source Audio



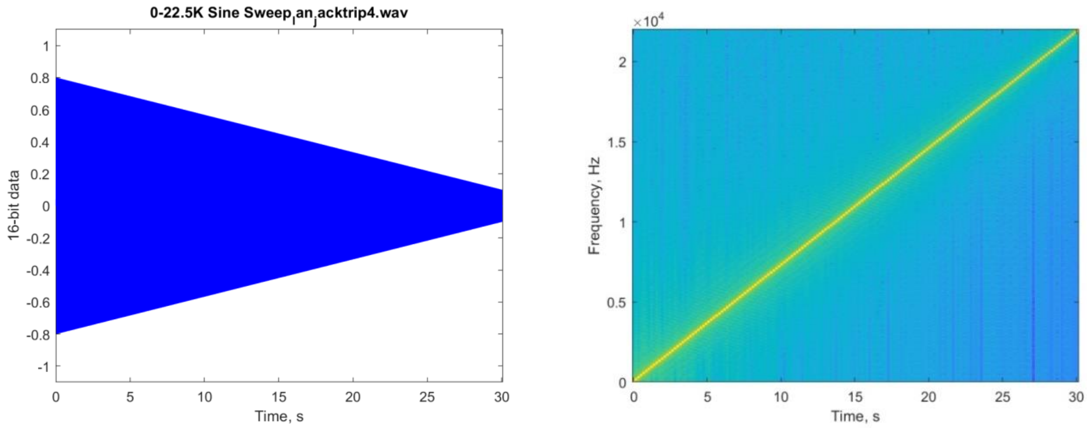
Streaming Capture 1



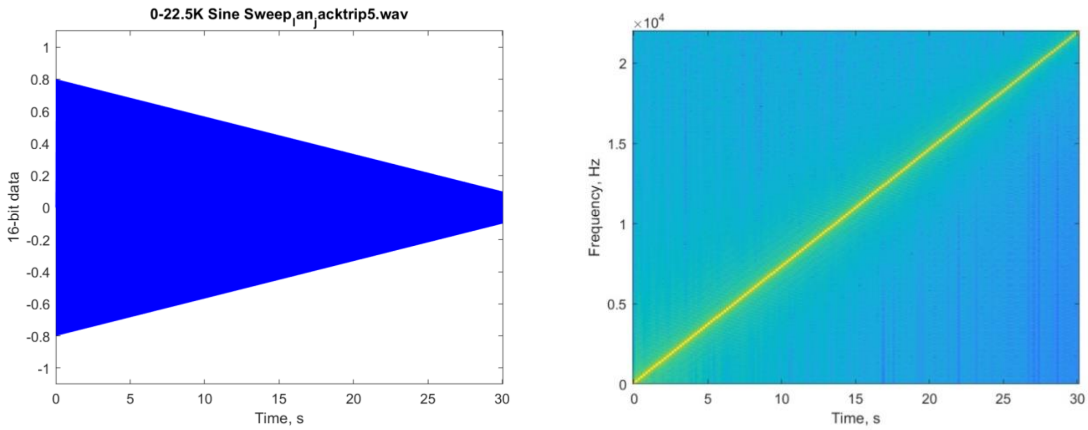
Streaming Capture 2



Streaming Capture 3

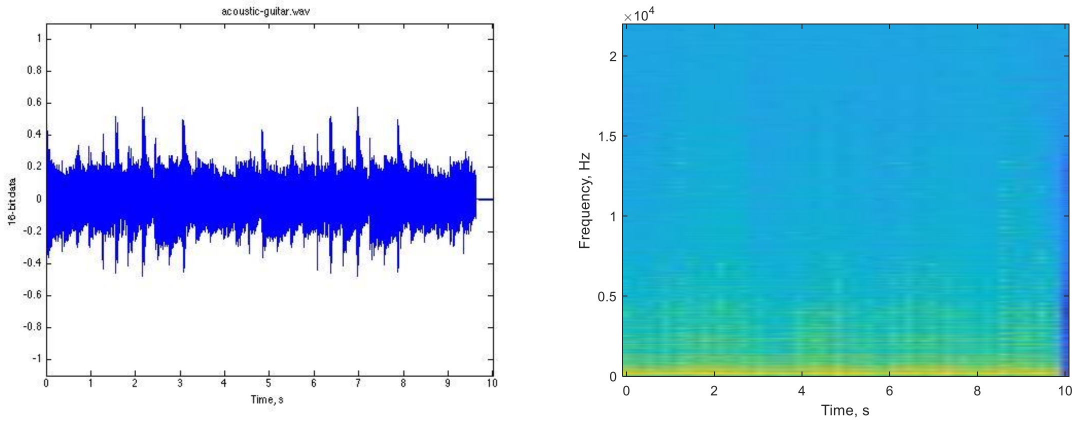


Streaming Capture 4

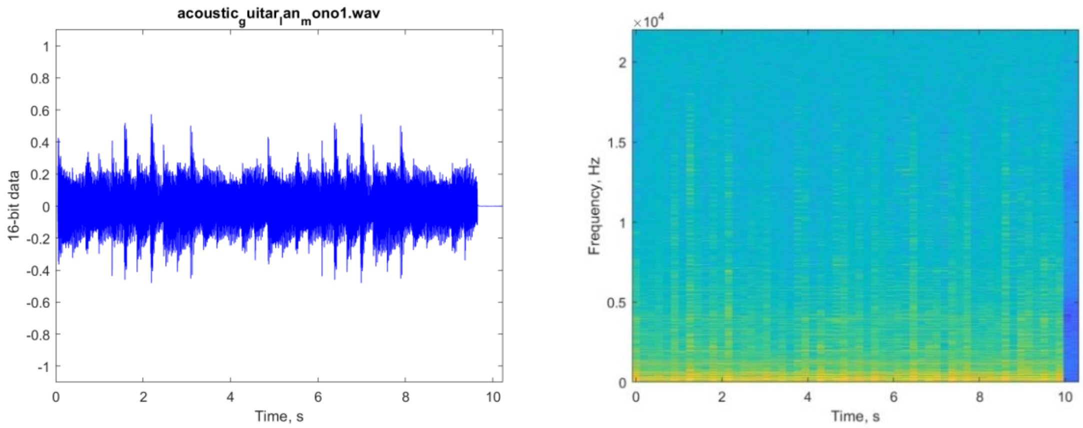


Streaming Capture 5

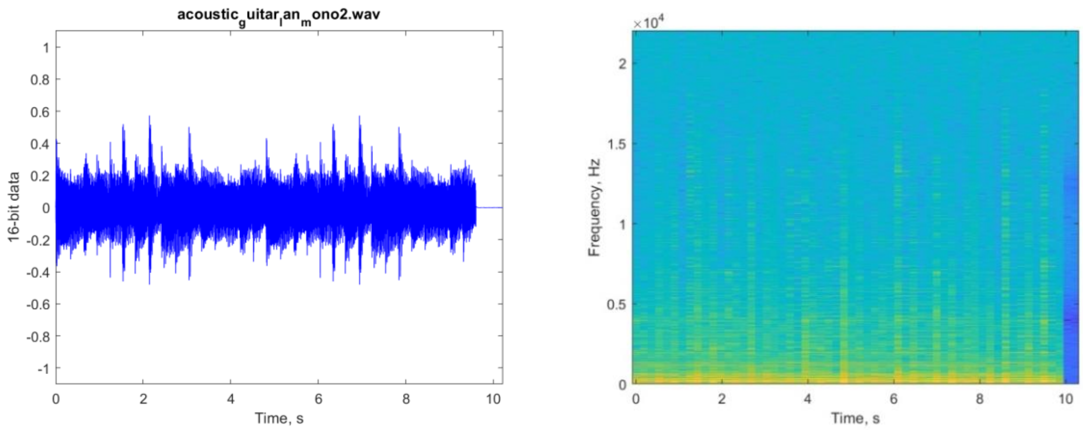
Acoustic Guitar



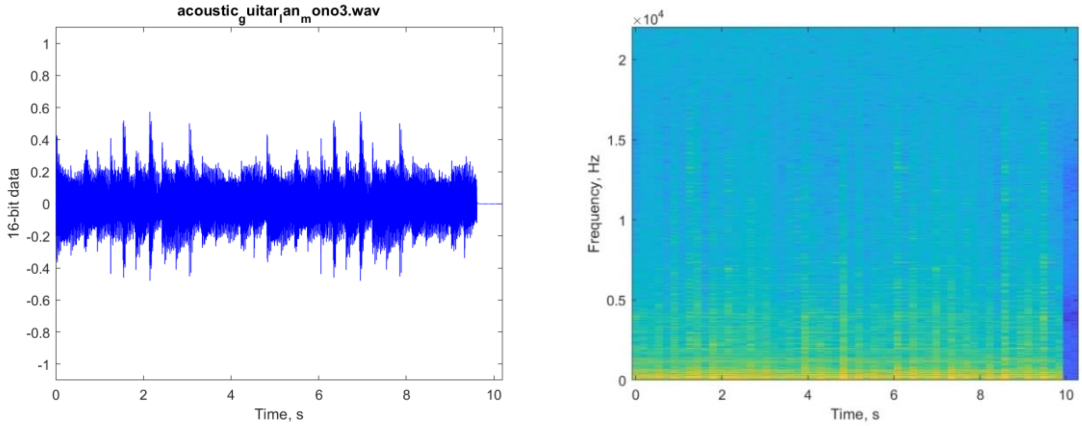
Source Audio



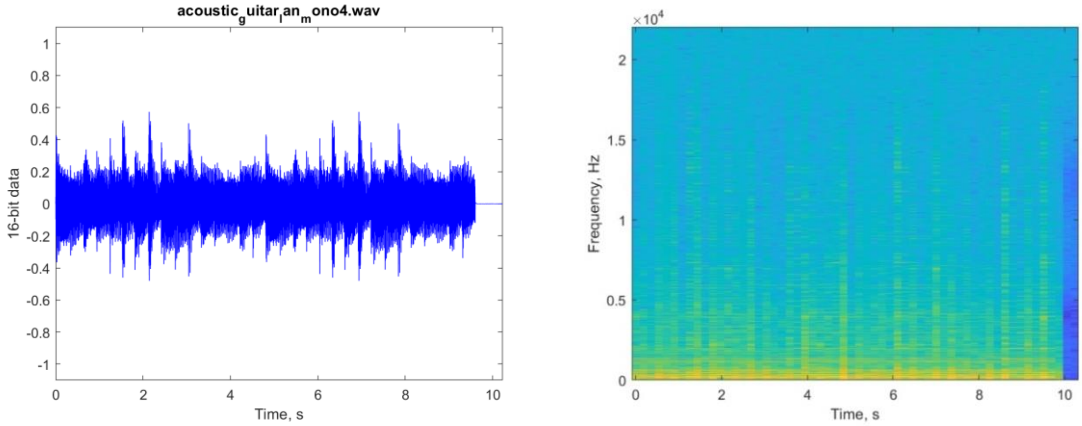
Streaming Capture 1



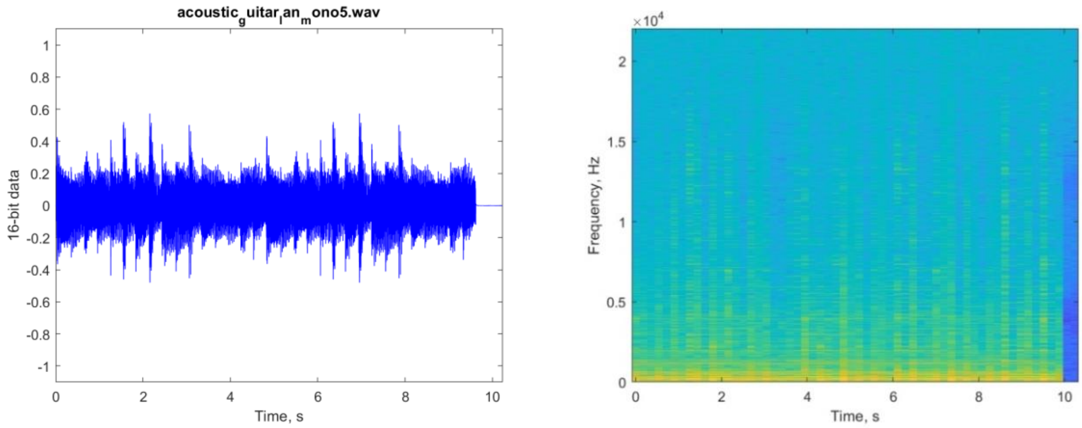
Streaming Capture 2



Streaming Capture 3



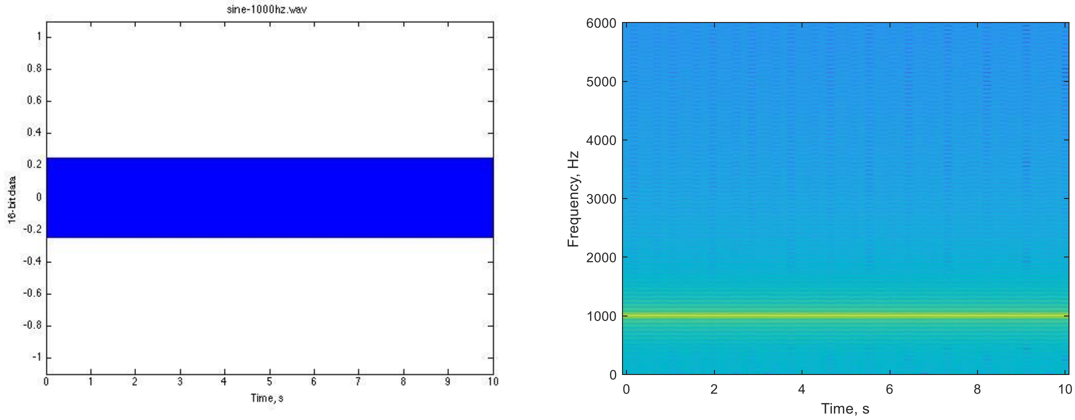
Streaming Capture 4



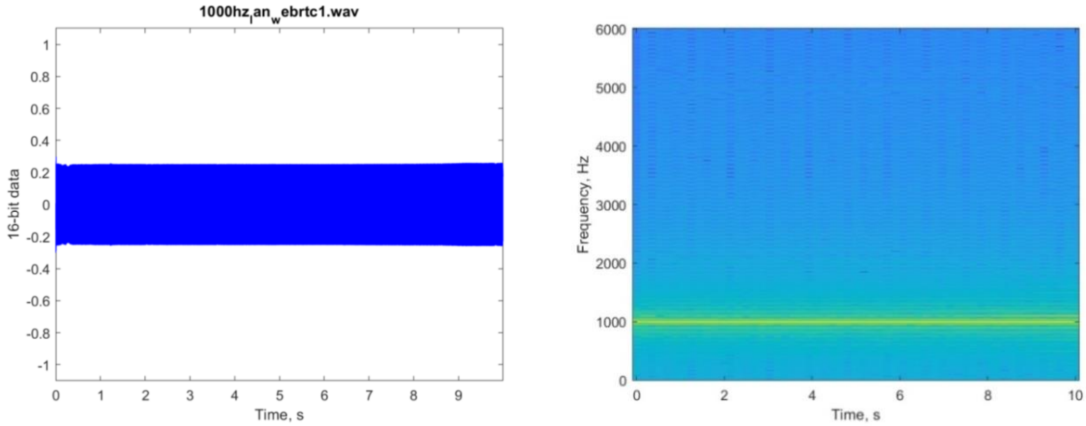
Streaming Capture 5

Appendix V – WebRTC LAN Audio Captures

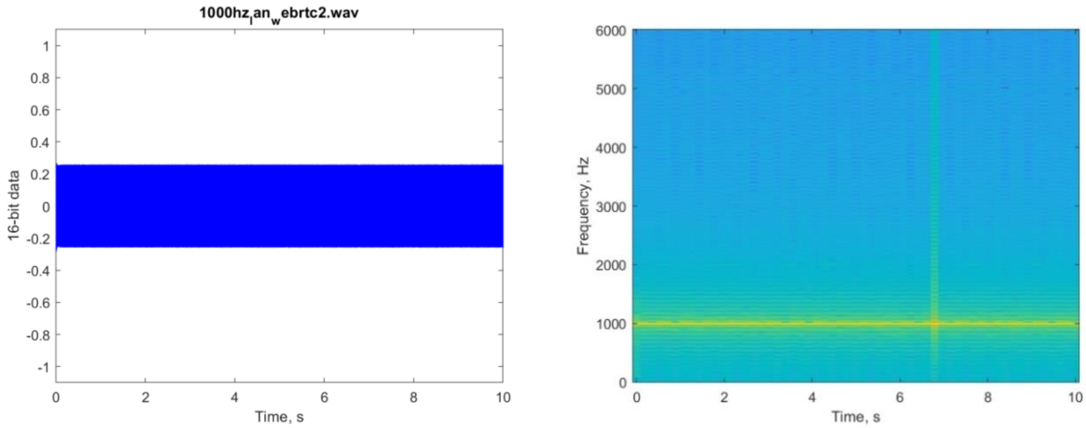
1 kHz Sine Wave



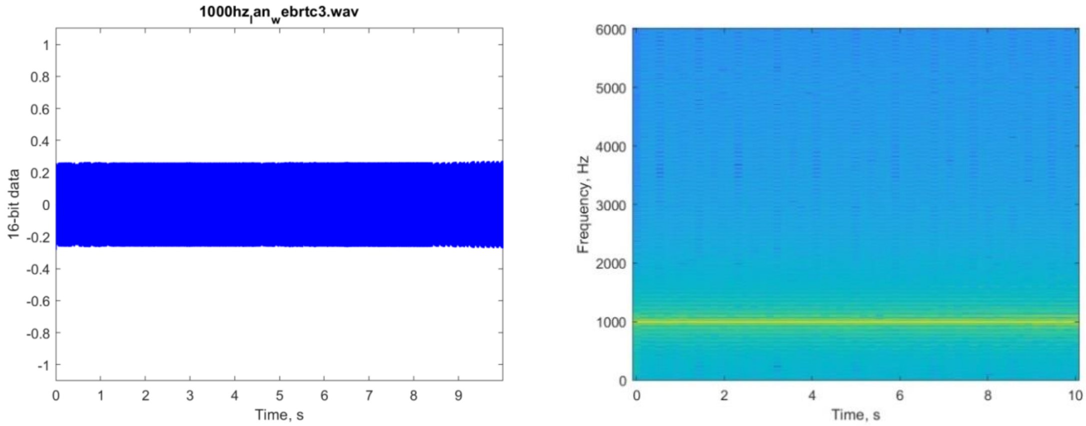
Source Audio



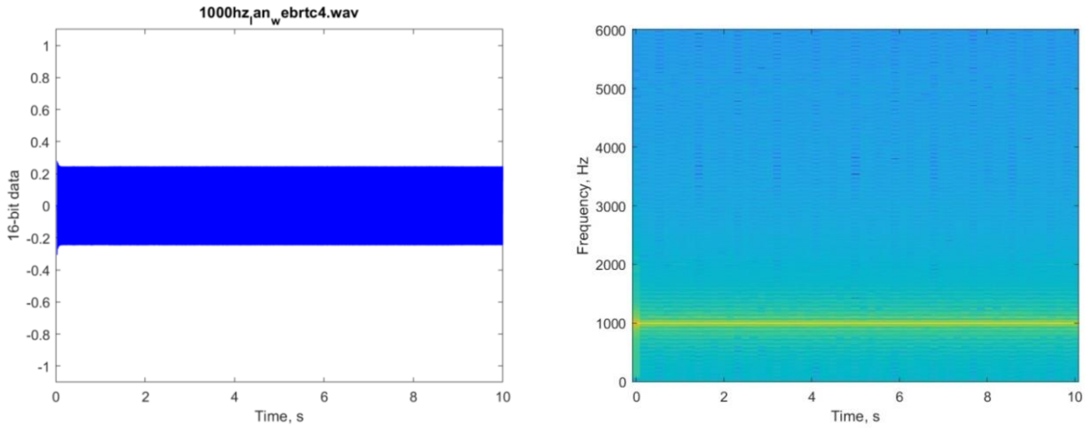
Streaming Capture 1



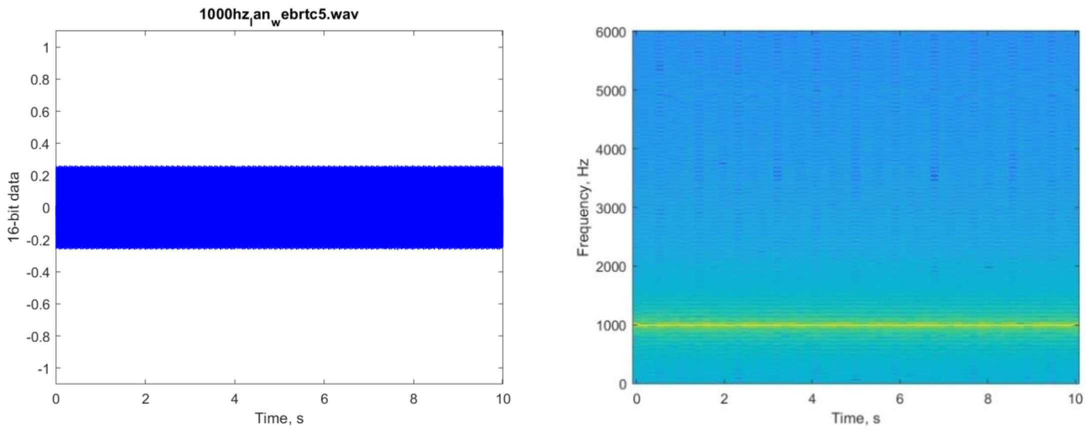
Streaming Capture 2



Streaming Capture 3

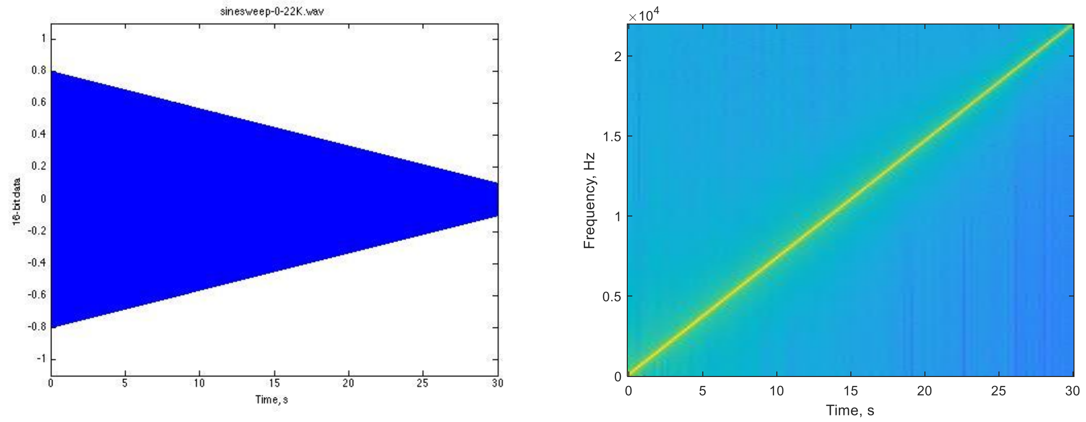


Streaming Capture 4

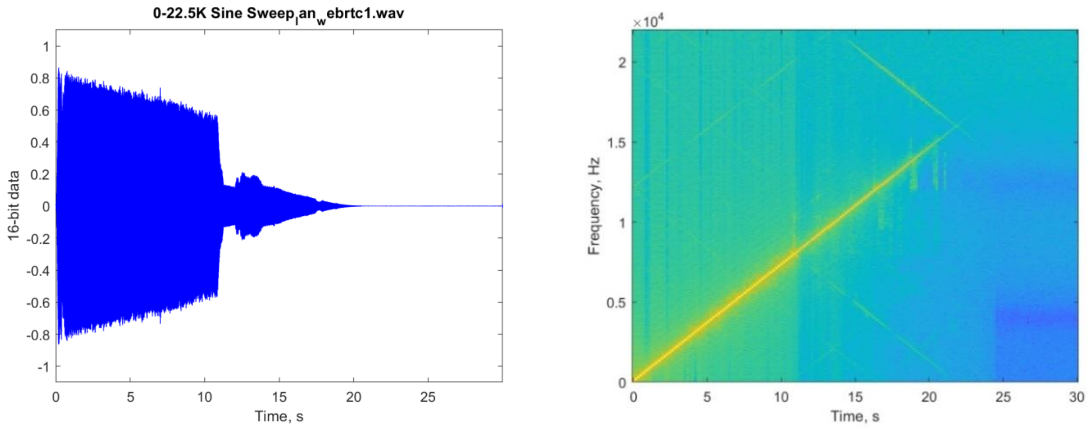


Streaming Capture 5

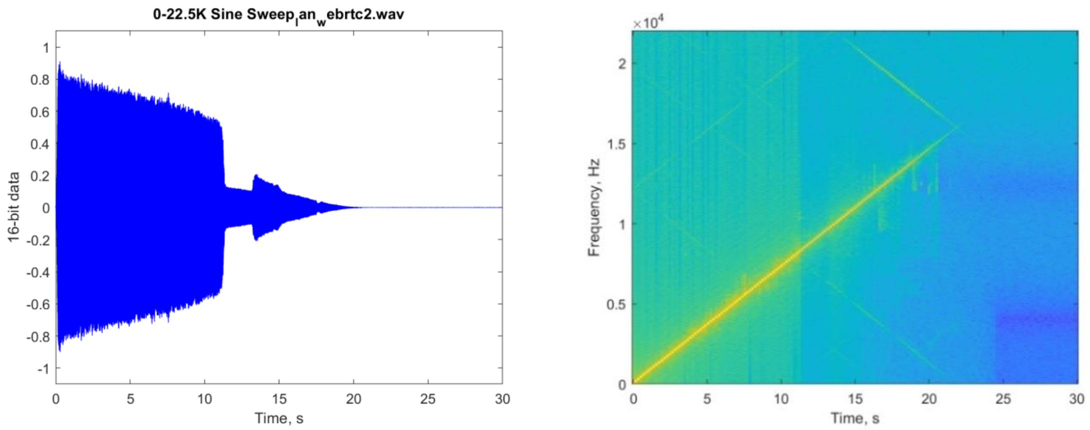
0-22.5 kHz Sine Sweep



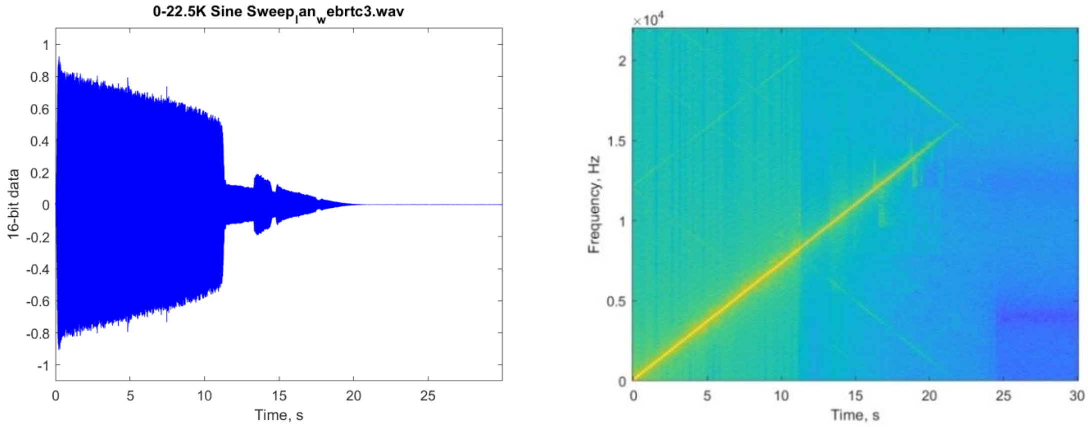
Source Audio



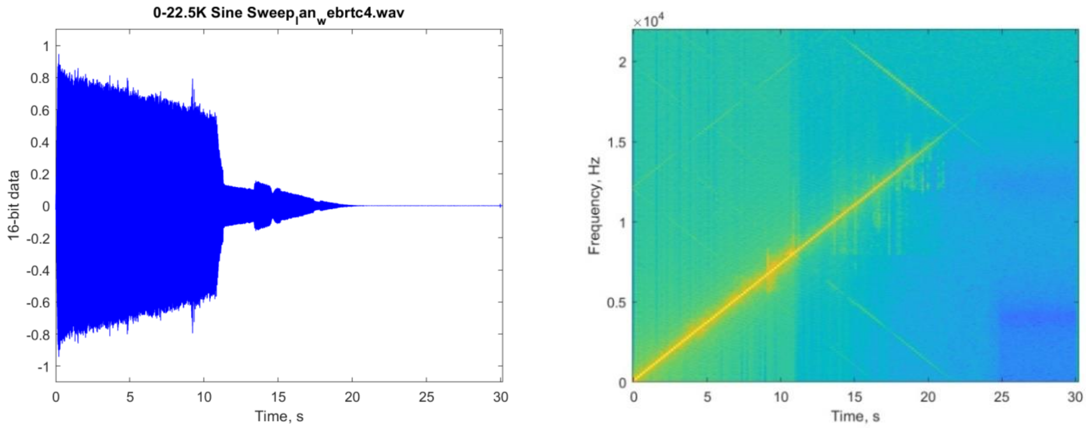
Streaming Capture 1



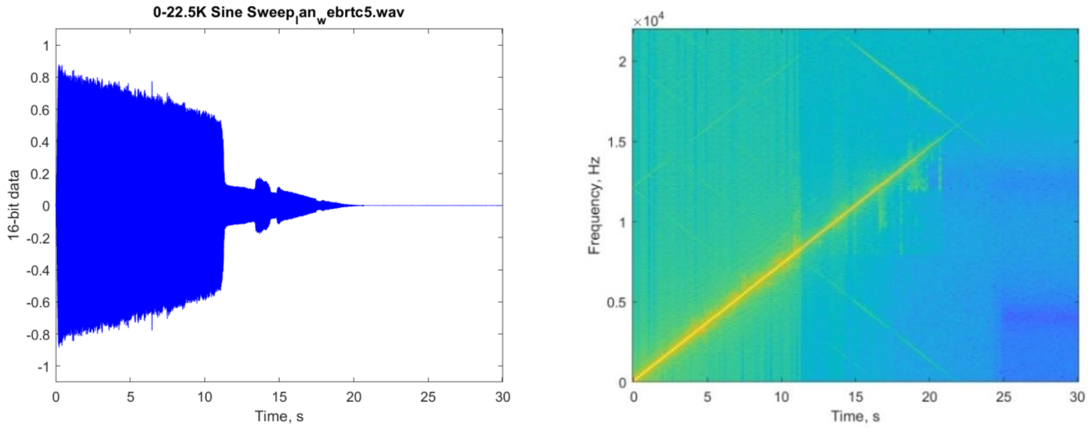
Streaming Capture 2



Streaming Capture 3

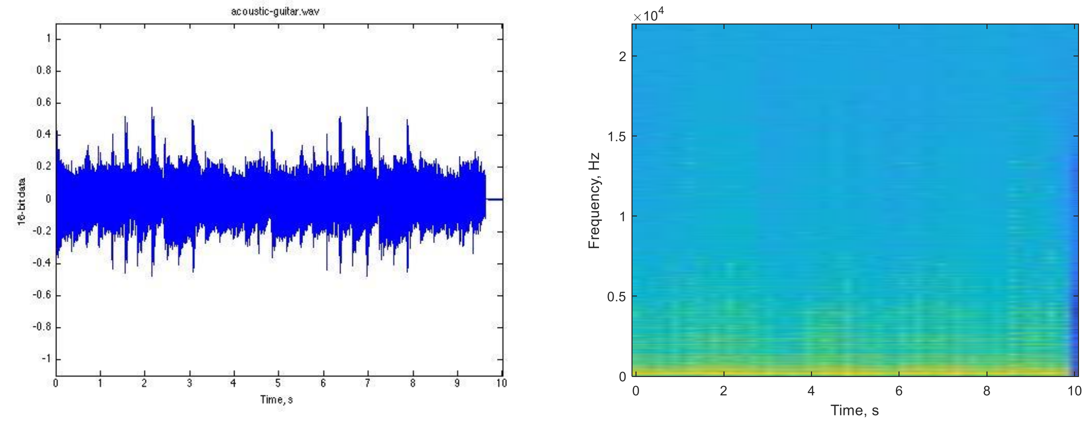


Streaming Capture 4

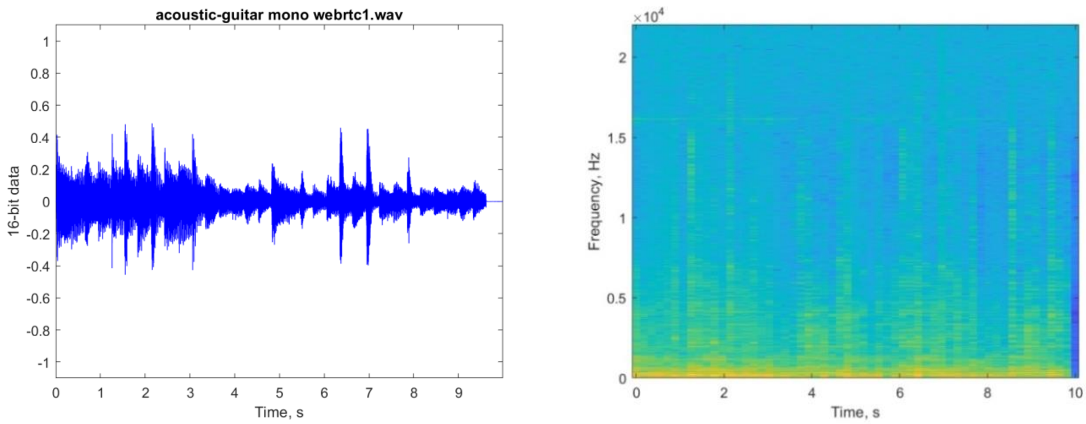


Streaming Capture 5

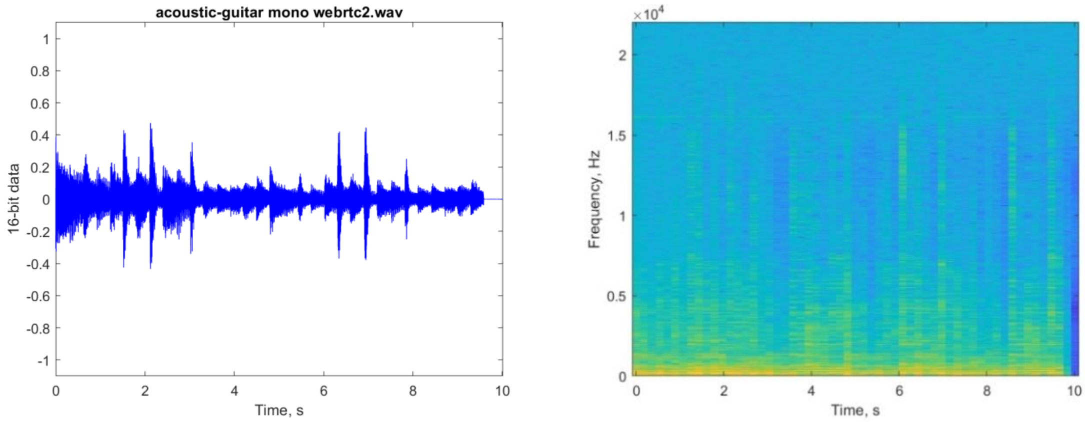
Acoustic Guitar



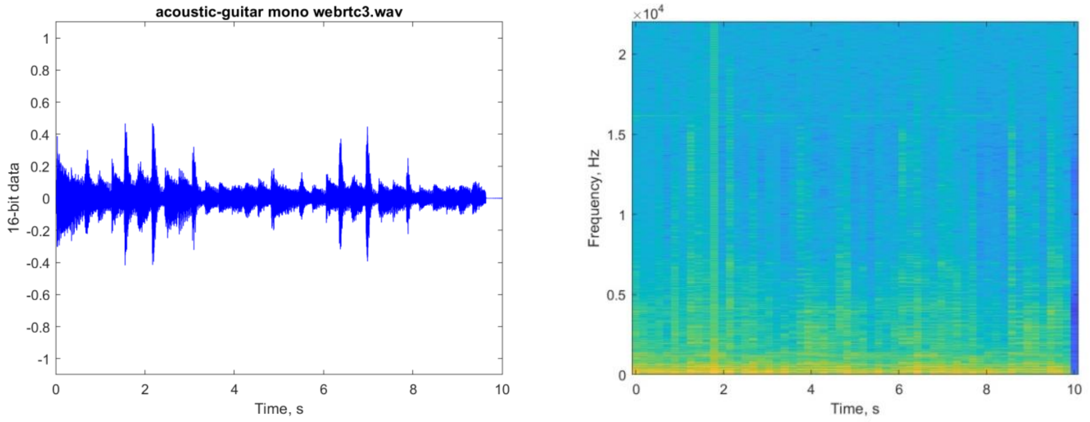
Source Audio



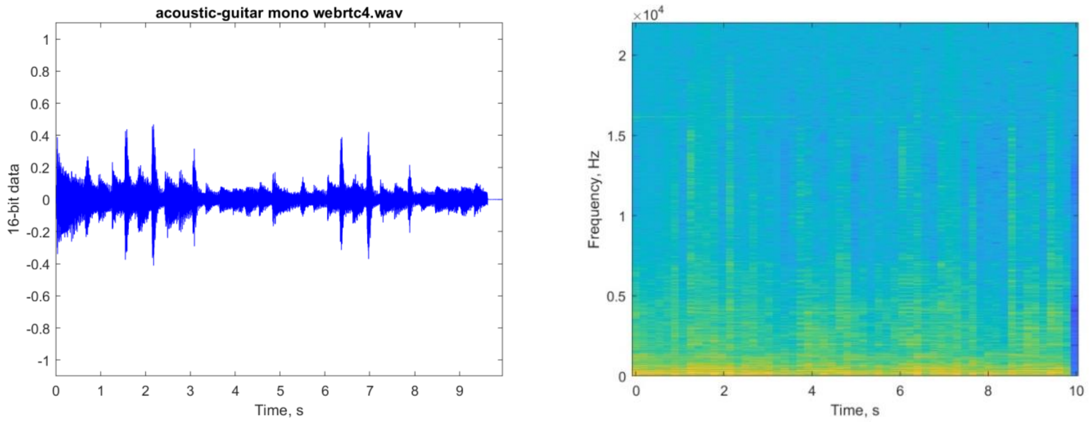
Streaming Capture 1



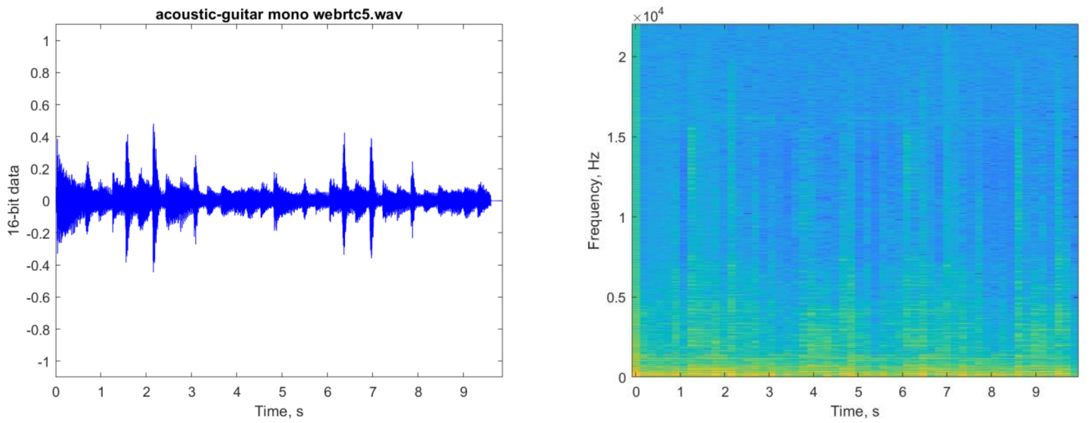
Streaming Capture 2



Streaming Capture 3



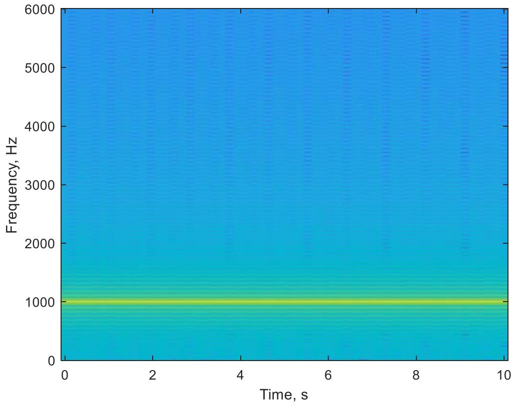
Streaming Capture 4



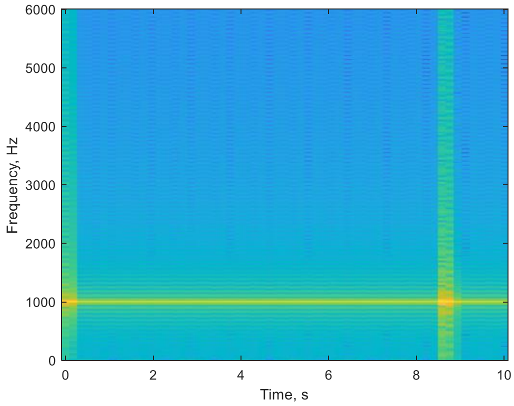
Streaming Capture 5

Appendix W – JackTrip Public Network Audio Captures (128kbps Buffer)

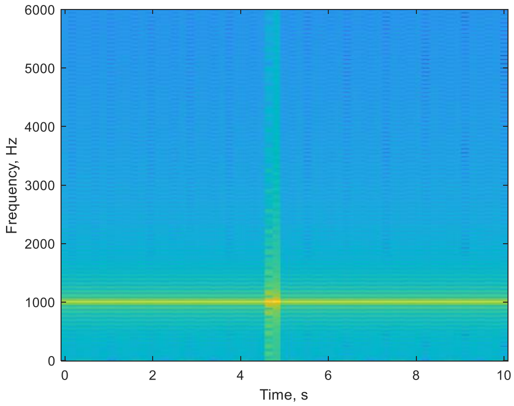
1 kHz Sine Wave



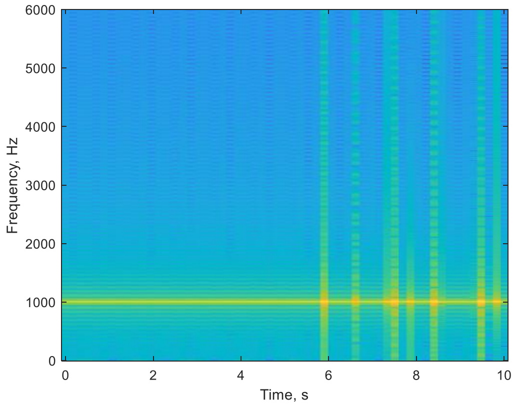
Source Audio



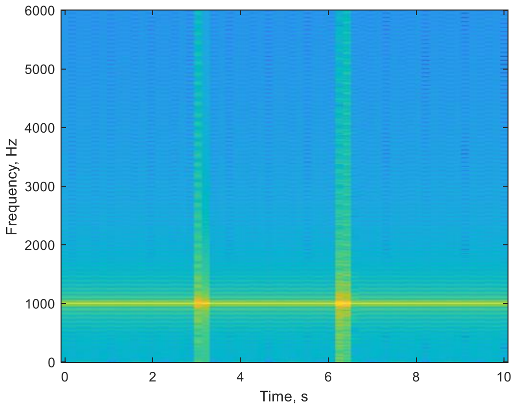
Streaming Capture 1



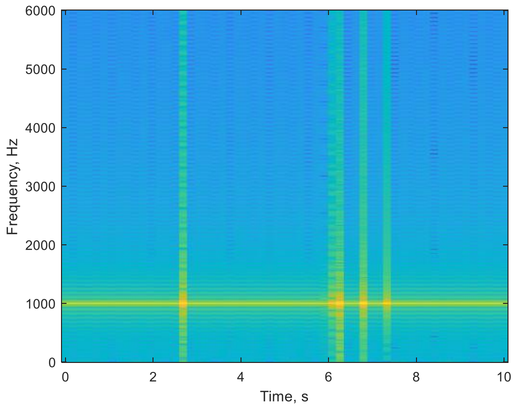
Streaming Capture 2



Streaming Capture 3



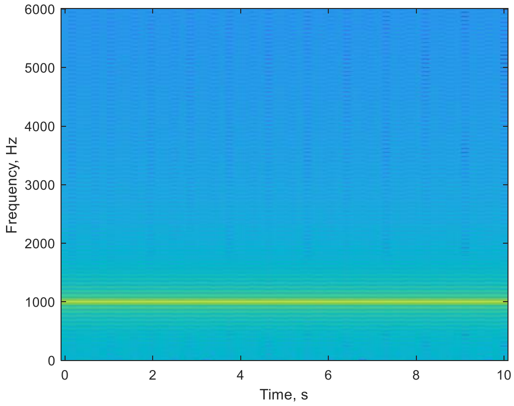
Streaming Capture 4



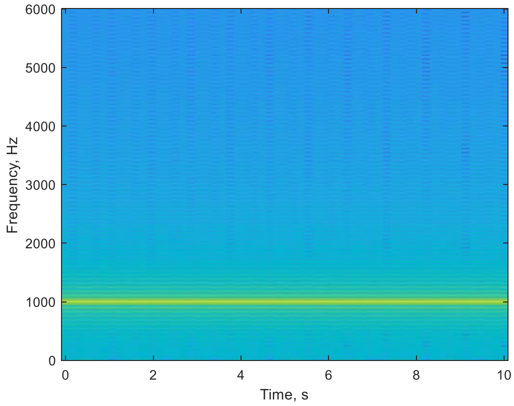
Streaming Capture 5

Appendix X – JackTrip Public Network Audio Captures (256kbps Buffer)

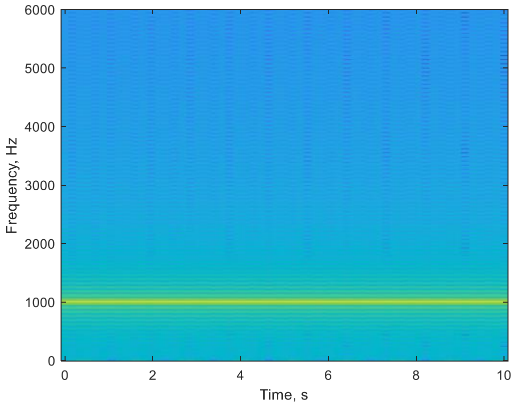
1 kHz Sine Wave



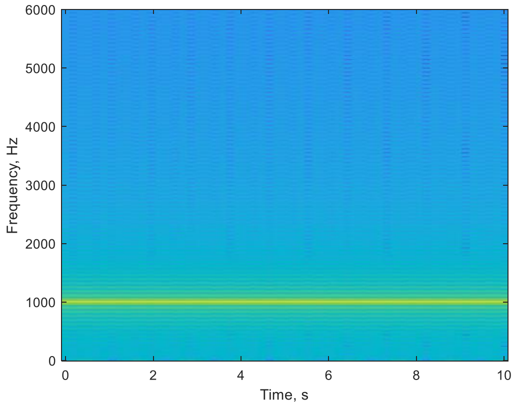
Source Audio



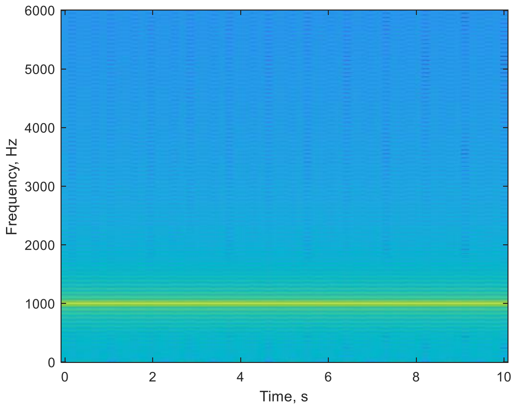
Streaming Capture 1



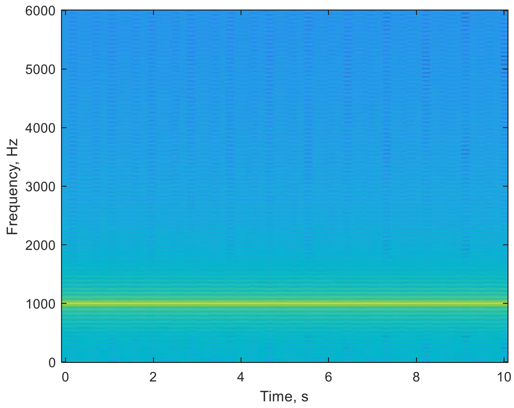
Streaming Capture 2



Streaming Capture 3



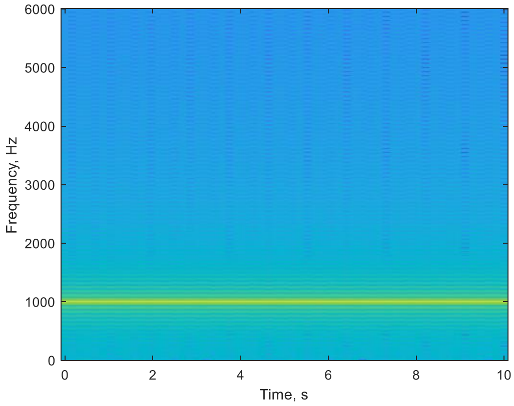
Streaming Capture 4



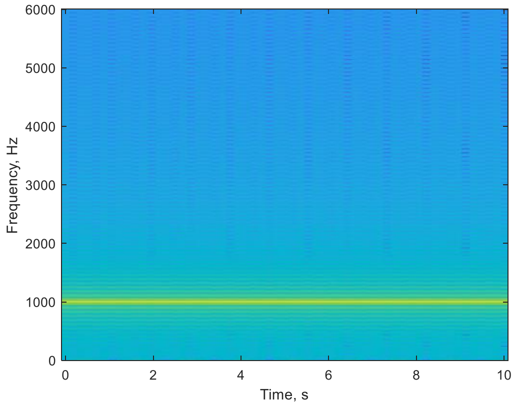
Streaming Capture 5

Appendix Y – JackTrip Public Network Audio Captures (512kbps Buffer)

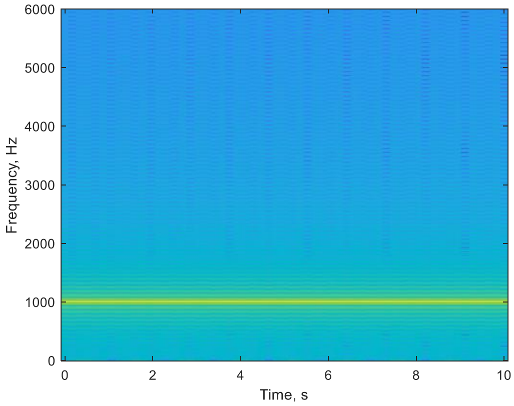
1 kHz Sine Wave



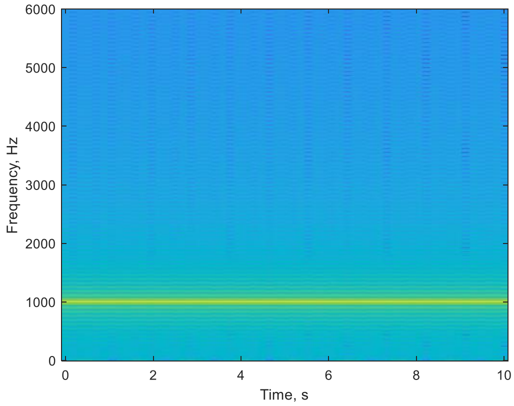
Source Audio



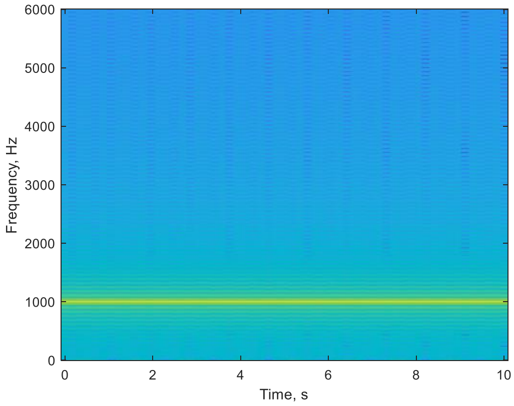
Streaming Capture 1



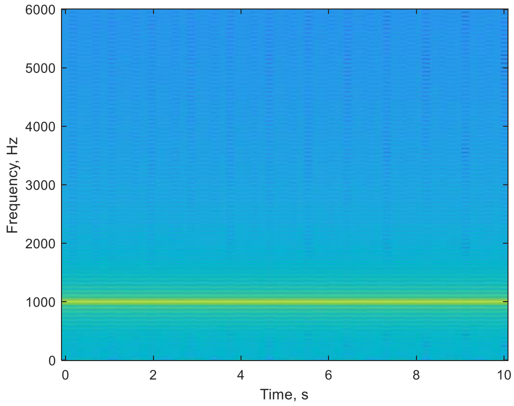
Streaming Capture 2



Streaming Capture 3



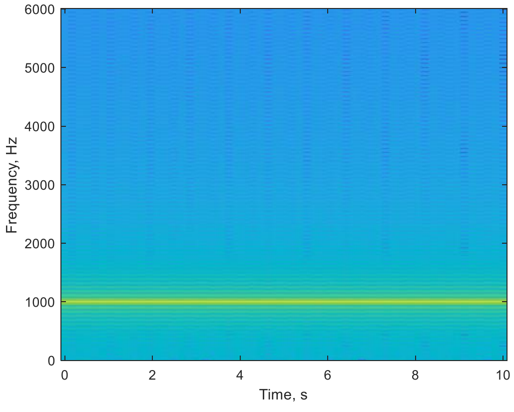
Streaming Capture 4



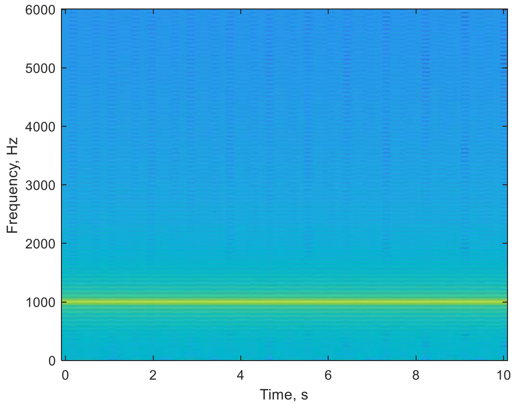
Streaming Capture 5

Appendix Z – JackTrip Public Network Audio Captures (1024kbps Buffer)

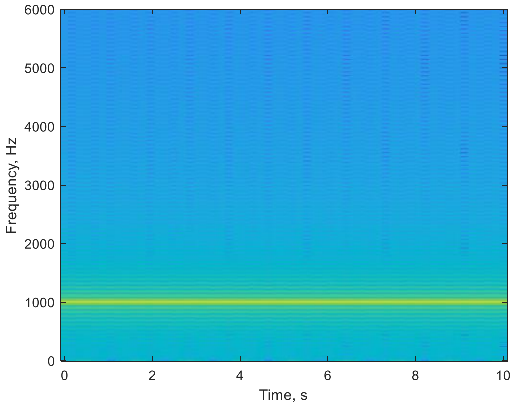
1 kHz Sine Wave



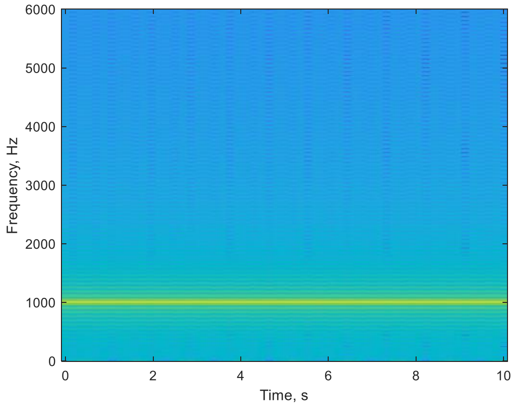
Source Audio



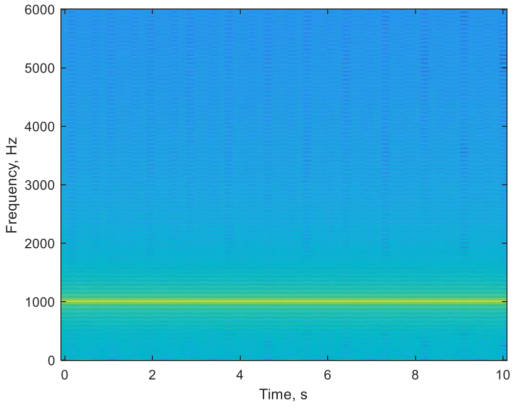
Streaming Capture 1



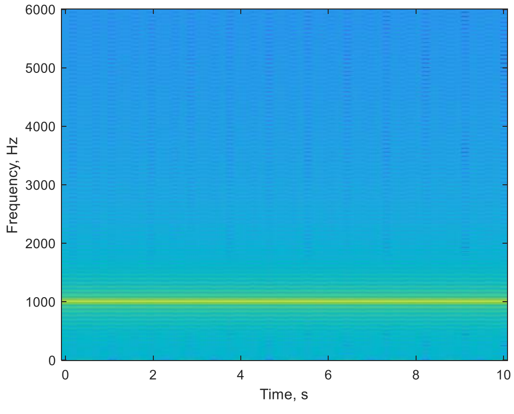
Streaming Capture 2



Streaming Capture 3



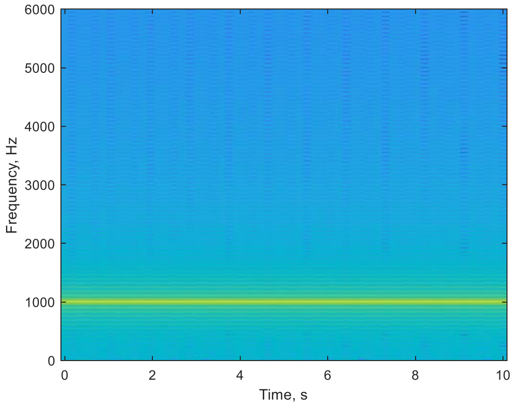
Streaming Capture 4



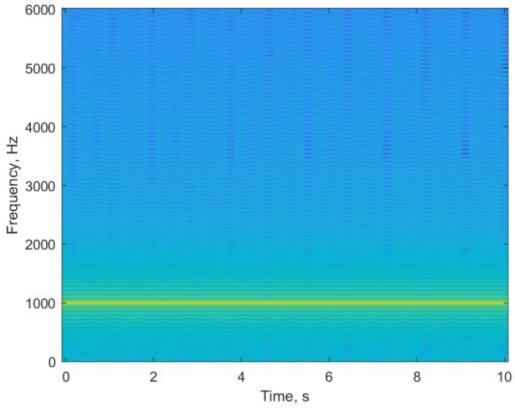
Streaming Capture 5

Appendix AA – JackTrip High Speed Research Network Audio Captures

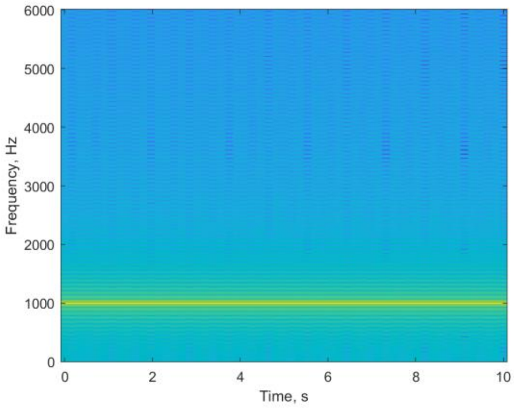
1 kHz Sine Wave



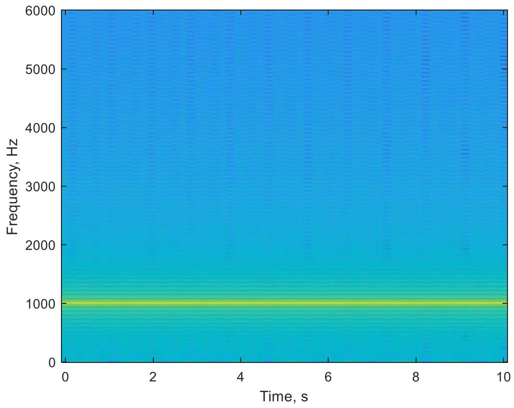
Source Audio



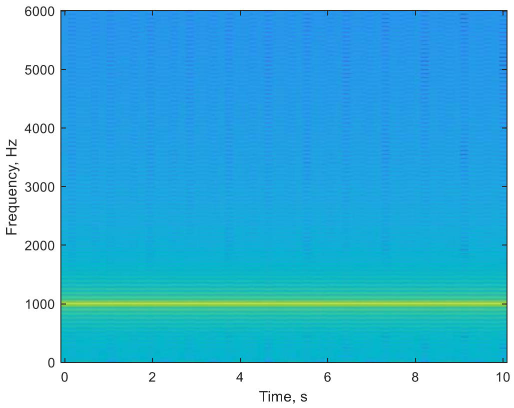
Streaming Capture 1



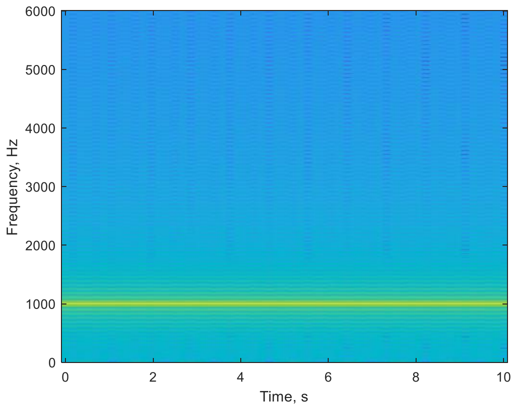
Streaming Capture 2



Streaming Capture 3



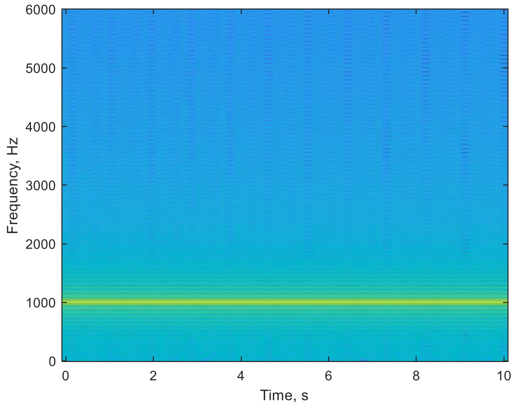
Streaming Capture 4



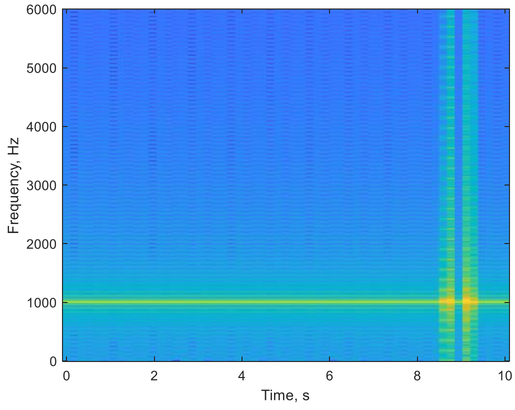
Streaming Capture 5

Appendix BB – JackTrip Wi-Fi Audio Captures

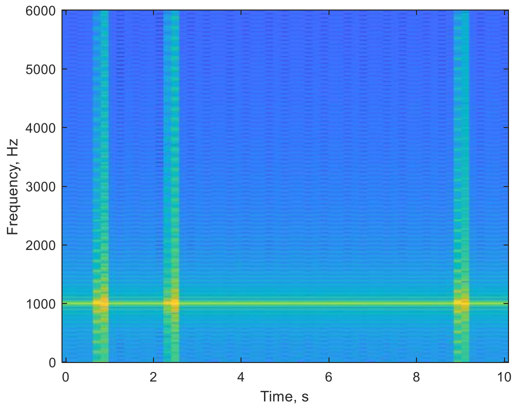
1 kHz Sine Wave



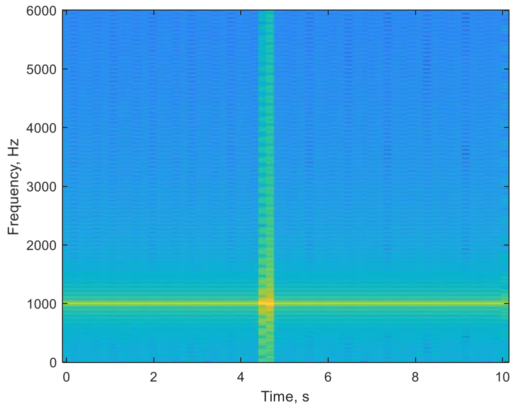
Source Audio



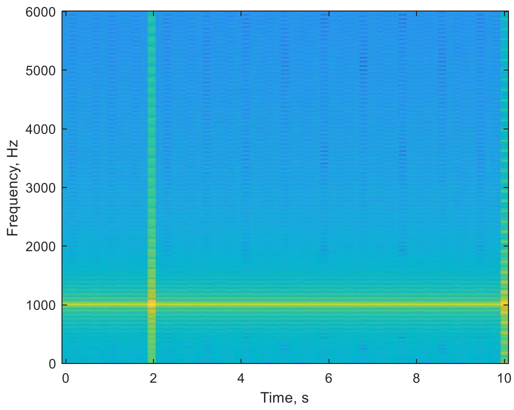
Streaming Capture 1



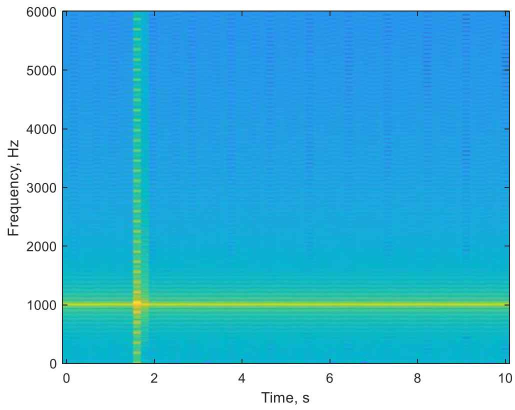
Streaming Capture 2



Streaming Capture 3



Streaming Capture 4



Streaming Capture 5

Appendix CC – Disparities in Distortion Measurements Using Different Audio Editors

THD+N Measurements

Test ‘Sine-1000hz’ Source File: 0.0075

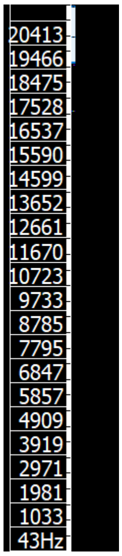
Non-Streaming Measurements – Import Source into Audio editor and Output as .wav

1. Logic Pro 16 bit .wav: 0.0183 (Exported Track as Audio File)
2. Logic Pro (2) 16 bit .wav: 0.0183 (Exported Track as Audio File)
3. Logic Pro (3) 16 bit .wav: 0.0075 (Export as Audio File using Right Click on Audio)
4. Logic Pro 24 bit .wav: 0.0183 (Exported Track as Audio File)
5. Logic Pro 24 bit .wav: 0.0075 (Export as Audio File using Right Click on Audio)
6. Logic Pro 32 bit unsigned: **Can only export as .aiff*
7. Audacity 16 bit signed .wav: 0.0357
8. Audacity 32 bit float .wav: 0.0075
9. Adobe Audition 16 bit .wav: 0.0075
10. Adobe Audition 24 bit .wav: 0.0075
11. Adobe Audition 32 bit .wav: 0.0075

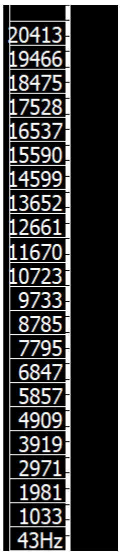
Empty (Zeros) .Wav Spectrogram in Sonic Visualiser

Spectrogram of an empty wave file (MATLAB .wav generated by array of zeros) visualised using *Sonic Visualiser.*

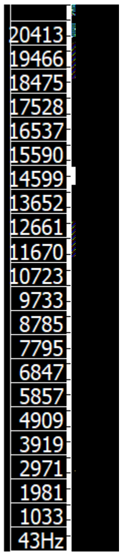
Source



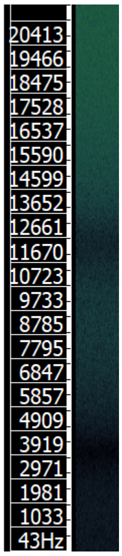
Logic 16 bit



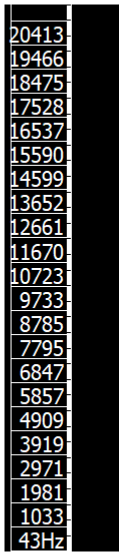
Audition 16 bit



Audacity 16 bit



Audacity 32 bit



Appendix DD – Follow Up WebRTC Audio Streaming Tests Data

	WebRTC LAN Measurements					
Category	**T1**	**T2**	**T3**	**T4**	**T5**	**Average**
Dropout	0	5	0	0	0	1
THD+N	2.9082	1.3686	7.0821	4.6278	5.3677	0.26814
RT Latency (ms)	149	135	134	132	135	137

The WebRTC test page at http://mjhardin.com went down prior to 2nd round of testing due to deprecated RTC functions. Testing was conducted using https://talky.io/, which may implement additional latency over servers

Appendix EE – IoT-Based Music Survey Questionnaire Questions

Terminology

- Physical Processing Systems: Hands-on devices that can be found in a live music studio (i.e. physical EQ, reverb, compressors, etc).
- Digital Software: Software programs like Logic, Ableton, etc, that provide virtual tools and synthesised processors to produce music

PART 1: MUSIC BACKGROUND/CULTURAL QUESTIONS

1. Describe Your Musical Status.

☐ Musician/Music Producer ☐ Casual Music Maker ☐ Other Creative Practitioner

2. If You Mix and Process Your Own Music, Where Does This Normally Occur?

☐ Professional Recording/Music Studio

☐ Home Studio with Musical Processing Hardware

☐ Personal Computer With Digital Music Software (In-The-Box Software Only)

☐ I Don't Mix/Process My Own Music

☐Other: ________________________________

3. If You Mix Music, How Likely are You to Use Physical Audio Processing Systems to Process Music in Comparison to Software Equivalents (e.g Hardware Compression vs. Software Compression)

☐ Never ☐ Sometimes ☐ Mostly ☐ Always ☐ N/A

4. Do You Feel That Physical Processing Hardware Give Better Results Than Their Software Equivalent

☐ Never ☐ Sometimes ☐ Mostly ☐ Always ☐ N/A

Optional (Explain Choice)

5. **With Regards to Physical Hardware, Do You Feel Analogue Components Provide Better Results than Digital Counterparts?**

☐ Never ☐ Sometimes ☐ Mostly ☐ Always ☐ N/A

6. **Would Your Use of Physical Audio Processing Systems Increase If They Were More Accessible?**

☐No ☐ Yes ☐ N/A

PART 2: IOT MUSIC PROCESSOR QUESTIONS

7. **Do You Feel an IoT-Based Music Processing System Adds Greater Accessibility to Analogue or Professional Audio Processing Hardware**

☐ No ☐ Somewhat ☐ Yes

8. **Briefly Describe Your Impression of an IoT-Based Music Processing System if these systems were widely available:**

9. **If Available, How Interested Would You Be In Incorporating IoT-Based Music Processing Systems Into Your Own Music Workflows?**

☐ Not Interested☐ Somewhat Interested ☐ Mostly Interest ☐ Very Interested ☐ N/A

10. **Are There Any Pros and Cons You Can Envision From IoT Extensions to Music and/or Other Creative Fields?**

11. **Please Use this space to give any further comments or feedback regarding IoT-based Audio Applications:**

Appendix FF – IoT-Based Music Survey Interview Questions

Terminology

- Physical Processing Systems: Hands-on devices that can be found in a live music studio (i.e. physical EQ, reverb, compressors, etc).
- Digital Software: Software programs like Logic, Ableton, etc, that provide virtual tools and synthesised processors to produce music

PART 1: MUSIC BACKGROUND/CULTURAL QUESTIONS

Question 1: Please provide your name and a brief description of your professional background in music?

Question 2: Can you discuss the regular space(s) where you ideally mix and/or record music?

Question 3: Do you normally use physical music hardware or digital processing systems to mix music? Why is this the case?

Question 4: Can you think of situations where physical processing hardware give better results than their software equivalents (E.G Hardware Compression vs Software Compression)

Question 5: With regards to **physical devices specifically**, can you speak on your impressions between analogue vs digital hardware, and preferred use cases for either? (Example: analogue vs digital effect pedals, reverb springs/plates vs digital reverb hardware)

Question 6: {***Outside of IoT***} To what extent would your processes for mixing (and recording) music be affected if analogue and professional music processing hardware were more accessible and why?

PART 2: IOT MUSIC PROCESSOR QUESTIONS

Question 7: Do you feel an IoT-Based music processing system adds greater accessibility to analogue and hardware processing systems and why or why not do you think so?

Question 8: Briefly give an account of your overall impression of an IoT-Based music processing system if these systems were widely accessible:

Question 9: If IoT-based music processing systems were available, can you discuss if you would be encouraged or not to incorporate them into your musical processes/workflows?

Question 10: Are there any additional pros or cons you can envision from IoT and the fusion with music technology and/or other creative fields?

Question 11: Please share any other thoughts relating to IoT implemented music systems or other areas that can be influenced by creative uses of IoT that have not been previously discussed.