Interactive Music and the Public Internet; why Milliseconds Matter

David Lake
5GIC & 6GIC, Institute for Communication Systems (ICS),
University of Surrey, UK.

Abstract

Choral singing has been very heavily impacted by the COVID-19 pandemic, causing many groups to only be able to meet "on-line." Experience has shown this to be less-than optimal, leading to a high level of dissatisfaction with the technology. Yet we are constantly told that the Internet gives us the ability to carry-on our normal lives when we can't meet in-person. Many of us suspect this is not true. To understand why the Internet as it is built today cannot deliver a musical ensemble experience, it is important to understand the requirements of good musical interactions and some of the details, both historical and technical of how the Internet has developed. Whilst by nature this is in places complex, technical details and the historical context is important, and this paper seeks to provide an audience-appropriate precis of these areas. To illustrate the issues and explain the impact on singing, an experiment measuring a typical "Zoom" based session as many choirs have been forced to rely on is carried out and documented.

Introduction

The COVID-19 pandemic which started in late 2019 has hit choral groups particularly hard forcing many to suspend in-person activities. One would have expected with the prevalence of high-quality Internet connections and readily available home computing systems that some form of real-time collaborative singing would have been possible; the experience of many groups, including those of the author, is that any form of interactive music-making on-line is difficult, if not impossible. To understand why this is so requires a level of understanding of the method by which media is transported across the Internet which is not readily available. This paper seeks to explain how human sounds are captured by computing devices and transported by a data network between the two locations.

The paper then examines a very typical real-world scenario using two PCs connected via Zoom. The study looks at the end-to-end latency and quality attempting to identify the sources of latency and comparing with the required outcomes for successful music collaboration. In musical terms, latency refers to the time between one player making the sound and another hearing it – in computer systems, it is the time the data takes from travel from one point to another. As will be seen, both have a significant impact on the ability to produce ensemble-based music.

The Internet – A Brief History

When one thinks of "the Internet" the analogy most commonly made is to the traditional telephone network, at least the point at which every house had a fixed telephone in their house, the Public Switched Telephone Network (PSTN).

The telephone could be compared to a PC; the number to the Internet Protocol (IP) address each device has and the concept of "connecting" to another user would seem to be identical to the manner in which we "connect" to a media service on the Internet.

However, there are important differences both in terms of the developmental history of the Internet and in the manner by which we able to make music over it.

Sharing

In all telecommunications networks, the amount of capacity supplied is always many times lower than the number of connected devices – in the days of the PSTN, had every subscriber picked up their phone at the same time to make a call, only a small proportion of them would have been able to due to limited connections between exchanges.

On the PSTN, picking up the handset and calling the other party would cause a circuit to be allocated for those two people's exclusive use for the duration of the call – the service is referred to as "circuit switched." And there is the problem – even if neither person were to be speaking, the circuit would be held between the two users; massively wasteful when one remembers that in a conversation, roughly 30% of the time is silence and that only one person is speaking at a time in a typical conversation.

The problem is even more stark with data services – to send a few bytes of data only takes a few milliseconds and having to connect a circuit for many seconds at-a-time is wasteful and inefficient.

Another form of sharing was therefore required, one able to use the fact that statistically, the connection between the two ends only needs to be long-enough to transmit the data and then could be allocated to another set of users. Additionally, multiple conversations could happen sending data in different directions, to different endpoints.

Early systems used such statistical multiplexing techniques ("statistical" because it is based on a typical amount of data; "multiplex" means simply combine different elements into one entity) to break a connection into multiple parallel paths, allocating a defined timeslot between endpoints asynchronously. However, this limited the number of connections to those pre-defined in the network.

The invention of packet switching whereby each data stream is broken into small bundles, the "packet," and then told where to be sent by using an IP address allows sharing of a different kind. Individual computing devices bundle-up the data into packets, label them to the destination with the IP address and send them to the next point – a router. The success of IP as a protocol has generated the enormous growth of the Internet.

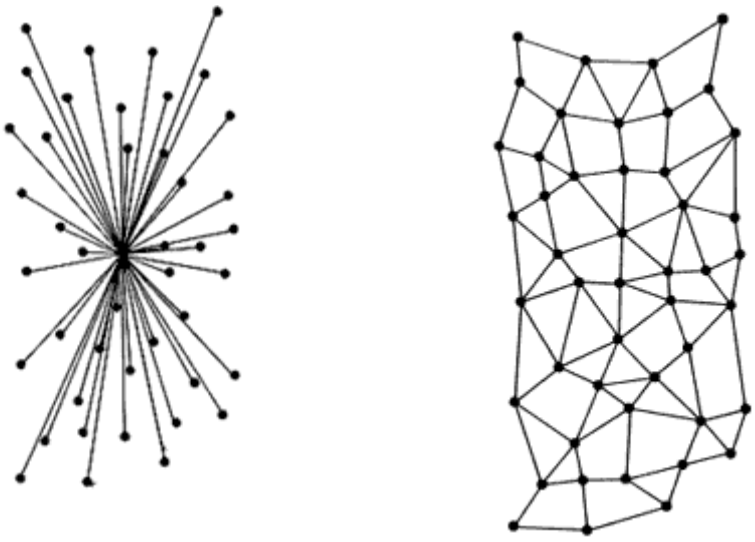
Survivability and Control

The origins of today's Internet can be traced to work at RAND Corporation, a US Department of Defence contractor in 1962 by Paul Barran (1926 – 2011) (Rand Corporation 2021).

A question troubling the US military at that time was the ability of telecommunications system, used as they were for controlling armaments, to survive a nuclear attack. The model of all networks at that point was centralised where the intelligence for the system was located in a single controller – take out the controller, and the entire network would be useless.

Barran suggested that the networks should be distributed (Fig i) with each node having enough information to reach its immediate neighbours, but no more. Also, the links between nodes would be redundant – there would be multiple connections from each node – and algorithms would determine which path to use next. Data would not be switched from end-to-end through a central brain, but instead intelligently "routed" between nodes. Barron referred to this system as "hot-potato routing" (Baran 1964) and the core nodes which switch packets today are known as "routers."

Figure i - Centralised vs Distributed Model



(From Baran, 1964)

Packetisation

Central to the idea of the distributed network was that data would not be sent in one continuous stream between two locations but be broken into chunks of data each able to be routed in a different manner to reach the destination. Barron referred to these as "blocks."

Parallel to the work of Barran, Donald Davies (1924 – 2000) at the National Physical Laboratory in the UK had developed similar ideas in 1965 – he referred to the chunks of data as packets (Davies 1966), and this is the name that has stuck. Davies was unable to secure funding for the project and consequently the first packet-switched network in the

world, the direct forerunner of today's Internet, was the US Advanced Research Projects Agency's ARPANET (Advanced Research Projects Agency Network).

Relevance to the Current Internet

These facets of the original ARPANET are of more than passing interest as they are core elements of today's Internet, relevant to the discussion of music interaction:

- There is no end-to-end "control" of the Internet; nodes will only know how to get to their neighbours, not the entire path, and typically will not have a great degree of information about quality of the link other than in terms of the next hop.
- All data is sent as a series of small packets; an email may consist of many thousands of packets; a video will be a constant stream of packets. These are of finite length and therefore represent an amount of time.
- There are multiple paths between any two locations on the network. This means that packets associated with the same data service may take different length and/or timed paths. Consequently, there is no guarantee that the order and time point at which data emerges is the same as the order and timing that data was sent.
- Individual nodes manage their resources independently. In terms of both the computing power of the node and the link capacity between adjacent nodes, the routers can and will discard sufficient data as needed. Because the nodes have no knowledge of the overall service – they simply see "packets in flight" – they lack the intelligence to know whether discarding ("dropping") packets will negatively impact the service. Likewise, because there is no overall "controller," no entity has a view of service delivery, instead leaving it up to the applications to deal with packet loss.
- There is much talk in the public of broadband "speeds" and latency, and it should be obvious that in a shared network where every hop is responsible for its own treatment of the data, the notion of a fixed "speed" or of a defined transit-time (latency) between any two points is a nonsense. Additionally, any measurement taken will only be of use for the time when it was taken – traffic patterns on the Internet varying greatly and every packet is complete in itself. Unfortunately, this concept is not well understood in the general public leading to disappointment with broadband services.
- The majority of the Internet works on the principle of "best-efforts" – there are no guarantees of a quality of service in any form, including whether the packets will actually arrive at their destination at all.

For the majority of applications, the benefits of the diversity of connectivity and the occasional loss or re-ordering of packets are inconsequential – one can think of an email which may consist of maybe 1000 packets where 1 is lost that could take 2s to download in full – requesting the lost packet and reconstructing the message may add a few tens-of-milliseconds, but for an email, this is irrelevant.

Other systems that require a constant-time playout, e.g., live-streamed video, work around the issue by "buffering" data - packets arrive and are stored in a small amount of

memory for a defined amount of time allowing the receiver to place packets in the right order or request lost packets. A balance needs to be struck between buffering to cope with loss on the Internet, and the additional delay that the process adds between receiving the data and playing the video or audio out. A buffer size of a few seconds is not uncommon. Live-streamed video may appear to be a "killer-app" in terms of high-bandwidth, but most is one-way (watching a film) and techniques such as Content Delivery Networks (CDNs) are used extensively to position data closer to the user. If you are watching an "on-demand" TV programme, chances are that it is actually coming from a server in your nearest large telephone exchange run by a CDN company, not the original provider at-all.

Consumer Broadband

There is one caveat that needs to be mentioned to the distributed nature of the Internet in terms of consumer broadband, both DSL (Digital Subscriber Line – the most common type in Europe which using existing copper telephone wires) and DOCSIS (Data over Cable Service Interface Specification – used on existing Cable TV networks and popular in North America) based.

Consumer broadband is a centralised network within each Internet Service Provider (ISP) which then connects to the Internet at Peering Points. Each broadband operator acts as a terminating node on the general Internet and provides a central "anchor" point that all their home users connect to. To send data to any other point, the consumer will need to transit their own anchor point to reach the general Internet before being onward routed. This is also true for users on the same broadband provider – the only common point between them will be the central anchor point which may be many hundreds of miles away, even for two people living next door to each other. In the case of where two neighbours are connected to different ISPs, data will exit the ISP and cross the general Internet usually at an Internet eXchange Point (IXP).

The Internet for Real-Time Media

Introduction

It should be obvious from the previous section that the Internet was simply not designed to handle the type of interactive real-time media that one associates with video conferencing and Voice over IP services, but during 2020, these have been the major applications. So how is it that, for the most part, we can carry out these kinds of functions, but we struggle to make music on-line? Surely, audio is audio? To understand the issue, another history lesson is required, this time around the emergence of Voice over IP in the late 1990s.

Voice over Internet Protocol – VoIP

Whilst early attempts at voice over the ARPANET in 1974 were partially successful (Gray 2010), it took until 1995 and the availability of a PC program, iPhone by VocalTec (RADVision, 1997, not to be confused with the later product from Apple) for VoIP to become more widespread.

VoIP relies on several elements:

- The ability to represent the voice by a collection of discrete codes;
- The ability to send the codes from one end to the other;
- The ability to maintain a regular stream of data such that the reconstructed voice is continuous.

CODEC

The mechanism by which the voice is coded and then decoded by a CODEC (a portmanteau of coder/decoder), needed as the bandwidth required to send a natural voice sound over the Internet would be much greater than is typically available. CODECS are usually referred to by their specification number given to them by the standardisation body ITU (International Telecommunications Union, a branch of the United Nations).

However, CODECS inherently cause clarity to be lost in the sound and the manner by which they do this is important. The most common codec, ITU G.711 (ITU-T 1988) compresses audio sounds of any source to 64kit/s and can be used for music and voice; more advanced techniques such as G.729 (ITU-T 2012) compress the audio much more aggressively but are closely coupled to voice sounds – trying to play music across a G.729 VoIP session is disastrous.

A typical VoIP session will gather 20 or 30ms worth of audio and produce a "code" for that block which it will send out over the network. VoIP has engineered the data such that each block of code is sent in one packet so there is a concept of a "packetisation" time in VoIP. Therefore, every 20ms, one packet is sent to the other end where it is decoded and played out as analogue audio. The continuity of the output audio depends on the continuity of the packets arriving.

Re-ordering

The first issue that one hits is re-ordering of packets. As each packet contains one element of the final output sound, re-ordering the packets could cause the sound to be edited – this is analogous to tape splicing with a razor blade and tape! Packets are therefore numbered, and the receiving end ensures that the packets are in the correct order.

However, there is a problem – how long should one wait for the correct order of packets? If I send 1,2,3 but receive 3, 2, 1, I will need to wait 3 time periods or 60ms with a packetisation of 20ms before I had the correct sequence.

Packet Loss

A similar issue occurs with packet loss. If I receive packet 1, then packet 3, I can assume that packet 2 is missing but I would only know after receiving packet 3 which means I have to wait 3 time periods, or 60ms in a 20ms packetisation network. Plus, requesting packet 2 may take many milliseconds – how long do I wait as any break in the stream would appear as a period of silence followed by an odd delay?

A quirk of the human ear comes to play here with speech – the brain is very good at ignoring and compensating for small gaps in speech and the loss of a few packets is immaterial in most cases.

Latency and Jitter

Latency is a function of all networks – it is simply the time taken to transit the network due to the speed of electrons/photons in the network components. In circuit-switched networks such as the PSTN, this is fixed and can be calculated. This simply adds a delay to the voice transmission as one is used to witnessing on long-latency satellite links.

More problematic is jitter – the variance in the delay which in a packet network where multiple paths are used, and resources are shared is impossible to calculate. The requirement is that the audio is reconstructed from data received every 20ms but jitter may mean that even though the data is sent every 20ms, it may arrive with gaps both smaller and larger than 20ms. De-jittering the data requires again that the data is buffered for a period of time until there is enough that can be played out at a regular, 20ms pulse.

CODEC Renegotiation

More recently developed are multi bit-rate codecs which adapt to network conditions by changing their coding algorithm in-flight. This is particularly noticeable on radio contributions from interviewees at home – as bandwidth becomes more limited, the codec will adapt to a more aggressive system and to the musically-trained ear, the "timbre" of the sound will change. An unscientific straw-poll amongst musically and non-musically trained friends shows that the non-musically trained don't notice it whereas it is hugely distractive for those with a musical background.

Long-Distance Phone Lines

It was quickly obvious that VoIP had the potential to replace many of the very expensive long-distance (toll) phone lines in the US and by the late 1990s, carriers were using the Internet to provide interconnections across the country rather than relying on trunk circuits. Likewise, as the Internet grew outside North America and Europe, new VoIP carriers emerged to offer cheaper international telephone connections, often in competition with existing telecoms operators.

In the early 2000s, VoIP started to replace traditional telephone systems in private businesses and in the core of public telecom operators – BT's 21CN project (Broersma 2004) aims to replace all circuit-switched trunk circuits with IP-based systems. This is not without issue – broadcasters and musicians have long used the ISDN2 service, a circuit-switched 64kbit/s link, for high quality, low-latency audio connections – this service is being ceased in 2025 and the Internet-based alternatives have proved problematic for many use-cases.

ITU-T Recommendation G.114

Underpinning the use of VoIP is the ITU G.114 (ITU-T 2003) recommendation which states that the one-way end-to-end delay on a voice call should not exceed 150ms. Whilst this sounds like a huge number, there are a number of factors one should remember in voice communication:

- In a telephone call/meeting, typically only one person will be speaking at a time.
- The communication is made up of words of finite length, longer than 150ms.

- The brain is very good at filling in the gaps – a missed letter in a word is irrelevant and will go unnoticed.
- Speech is 30% silence.

Buffering

It should be clear now that within the Internet there does not exist a steady stream of data at a defined rate – instead, the communication is quite "bursty", much like someone talking by saying maybe two-and-half words, waiting, talking, etc. It is very stop/start. We need to be able to turn those burts into a constant, regulated stream and we do that by using buffering to add an artificial delay.

Imagine that you are trying to run a steady stream of water but the supply you have is intermittent. There is a simple solution; point the hose into a bucket with a hole in it. Cover the hole until there is a certain amount in the bucket and uncover it – as long as on-average the water in the bucket and the amount being delivered by the hose are the same or more than being let out through the hole, you will get a steady stream.

But this introduces a delay whilst the bucket fills up enough and if the in-flow stalls, you will cause the stream to stop for a while. This is exactly what happens when you stream audio and video over the Internet. A small delay is inserted to take the "bumpiness" out of the incoming stream. Provided that you don't mind that there is a delay, you won't notice. This is a common trick used in streaming video services – everyone has seen the dreaded "buffering" message or had to wait for the stream to start. This is the bucket filling up with water and/or the hose being trodden on! Video streaming services can get away with this because they are not two-way, interactive. Buffering times can be as high as 30 seconds, an unusable delay for music.

Summary

The main use-case for rich-media is VoIP for telephone circuit replacement in order to reduce costs. Extensive use is made of the ability of the human brain to fill-in missing sounds and cope with delays of up-to 150ms when dealing with person-to-person speech and that the mode of conversation is predominantly one-person-at-a-time.

Music Interaction

Introduction

When it comes to making music together, it is immediately obvious that there are some critical differences in the human interaction involved compared to a speech-based conversation:

- The interaction is collaborative; the musicians will be creating sound but listening to the other person at the same time.
- They will also be listening to the interaction of their sound with the other person – one need only think of how two singers will adjust to blend their voices both in terms of pitching and in response to the acoustics of the environment.
- Loss of any portion of the sound has a major impact on the ability of the performer.

- There is a symbiotic relationship between performers; one slows down, the other will follow!
- The quality and timbre of the sound matters.
- Making music requires intense concentration and any external factors can distract very easily.

Milliseconds Matter

Known to all sound engineers, there are two aspects to the effect of sound in a space that are important when considering how Internet technologies interact with musicians.

Speed of Sound

Table 1 shows the speed of sound in air at different temperatures and the corresponding time per 1m.

Table 1
Speed of sound in air

Temperature of air in °C	Speed of sound *c* in m/s	Time per 1m *Δt* in ms/m
+40	354.9	2.818
+35	352.0	2.840
+30	349.1	2.864
+25	346.2	2.888
+20	343.2	2.912
+15	340.3	2.937
+10	337.3	2.963
+5	334.3	2.990
±0	331.3	3.017
-5	328.2	3.044
-10	325.2	3.073
-15	322.0	3.103
-20	318.8	3.134
-25	315.7	3.165

(from Sengpiel 2021)

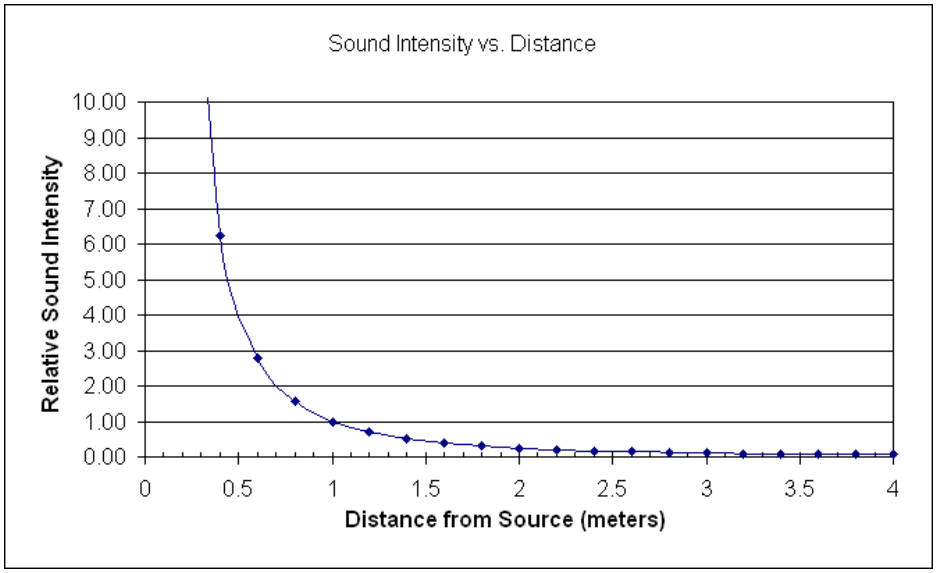
Putting this in a musical context, in the typical choir, each member would be about 1m from their neighbour and probably 10m at most from the most distant. In terms of latency, this is about 3ms between immediate neighbours and about 30ms at the largest distance. In a large choir, there will be a range of latencies, perceived differently by each singer according to their position in the choir. One only has to ask the basses and tenors to swap for one rehearsal to realise just how pronounced and ingrained this is. We each form and adapt to the sound-world we encounter.

Research shows (Chafe et al. 2004) that the optimal delay for natural performance between musicians is between 10 and 25ms with rapid deterioration above 66ms. There is also an interesting acceleration effect below 8ms most pronounced at 3ms. The style and genre of music is of importance – this should be of no surprise to choral singers used to long resonance times where slower anthems work much better than an up-tempo pop-tune!

Inverse Square Law

Sound decays according to an inverse-square law as shown in Figure ii. Our brains are used to coping with this – we expect someone further away to be correspondingly quieter and our experience will equate the delay between two people with the reduction in sound pressure.

Figure ii
Sound Intensity vs. Distance



(from Ricksci, 2021)

Application to IP Technologies

Putting these together in a world where we are separated by the Internet, it is obvious that music making using the available technology will not work and for good reasons:

- The latency inherent in packetisation, even if every packet were delivered perfectly in 20ms is already at the upper bound of what is acceptable for musicians. Even in a perfect 20ms world, the processes in the computer typically add 5-10ms of delay meaning that the minimum mouth-to-ear delay would be 30ms.
- For a choral singer or ensemble player used to their position in the group, the loss of spatial awareness due to the delay variances of the other musicians is problematic.
- Any variance in the delay is problematic – as each millisecond equates to one foot of distance, adding 2ms is akin to picking the musician up mid-flow and moving them 2ft away!
- A CODEC change would cause a change in the timbre of the sound which would be very off-putting.

- The sound pressure from VoIP is not varied in accordance with the latency – there is an unnatural feel to the sound which sounds at both times far away due to the delay but close-at-hand due to the sound pressure. Additionally, unless properly built through a DAW, every musician will appear at the same pressure.
- If wearing headphones, the musician will not experience any spatial changes to the sound-world, for example, if they turn their head. Likewise, unless a substantial surround-sound system is built, even with speakers, a very flat stereophonic sound world exists.

Real World Experiments

Introduction

In order to determine how possible it is to use the current readily-available technology to make live, interactive music, an experiment was conducted to measure the effect of transmission between two PCs running standard, business-grade video-conferencing.

Laboratory Setup

The most commonly used interactive software, now a byword of our times, is Zoom (Zoom 2021a).

The goal of the experiment was to quantify three elements:

- The latency due to the soundcard within the PC.
- The latency between the two PCs at a number of given time points including any variation over those measurements.
- The impact on the quality of the sound received.

The laboratory setup consisted of two PCs, PC A and PC B. PC A was connected via 1Gigabit/s wired Ethernet, PC B via WiFi to the same DSL service (76Mbit/s down, 16Mbit/s up) using a Zoom video conferencing session.

On PC A, a test "click" track was generated using Audacity (Audacity 2021) with the output connected via a VB-Audio Virtual cable to the input ("Mic") and the output ("Speaker") of Zoom connected to the input of Audacity via a second Virtual Audio cable. On PC B, the Zoom session was joined, and a VB-Audio cable connected such that the "Mic" and "Speaker" connections were looped together. The Zoom meeting was configured to use "Original Sound" as per Zoom's instructions (Zoom 2021b).

Zoom uses a cloud-located conferencing bridge in common with all other similar conferencing applications. The location of this bridge is important as it could add significantly to the latency. Unfortunately, the user has very little control over the location of the bridge – Zoom simply state that they choose a data centre based on the region set by the user profile (Zoom 2020). Further, the location is very general stating simply "Europe." However, the user can see which data centre location is chosen by clicking the shield in the top left of the screen even though they are unable to influence the choice in any way. The test runs were carried out with Zoom showing that it was connected to a data centre in Ireland.

A click track was constructed around a standard 4-beat rhythm track, with one pulse every 0.5 seconds. The first pulse of each group of 4 is at a slightly different frequency and a higher amplitude allowing to spot if the first pulse is lost and therefore retain synchronization. Audacity was configured to allow simultaneous playback and recording such that the looped-back waveform would be displayed below the played waveform. Each loopback test was repeated 5 times and the distances between the sent and received pulse measured. The full results are detailed in Appendix I.

Laboratory Results

The two VB-Audio cables are software-emulated and therefore will introduce some latency. Therefore, to quantify the amount of latency added to the overall budget, the click experiment was run using the input and output of a single cable to measure the total latency on both PCs. Results are shown in Appendix II. The result for this test was consistent and shows that on PC A the loopback cable accounts for 192ms whilst on PC B the loopback accounts for 160ms.

This scenario with loopback virtual cables is obviously not what one would have in a real environment – instead, headphones and microphone would be connected to either an internal audio card or a USB module. It was therefore felt to be useful to measure the loopback performance of the Sabrent USB audio card, a very typical, cheap USB audio module. The same Audacity click track was used this time to the local audio card with a loop connected between speaker and microphone sockets. The result, as shown in Appendix III, was a consistent 140ms of latency.

In order to understand other sources of latency, it was decided to analyse the traffic to/from the Zoom data centre, ostensibly in Ireland. Wirehsark was used to try to find the real location of the data centre. Analysis shows that the IP address of the chosen data centre was at IP address 134.224.116.28. Checking the geolocation of this address shows that it is allocated to "Zoom Video Communications Inc" and is in San Jose, California. However, further research revealed that the address is that of an Amazon Web Services (AWS) tenant and therefore could be located anywhere in the world as AWS use their own internal network.

A simple "ping" to the address shows that the Round-Trip Time from West Sussex averages 25ms so it is definitely NOT located in California. The minimum time is 24ms, maximum 28ms giving a jitter of 4ms. A "traceroute" reveals that the last hop that responds is 52.93.36.155, located in Washington, DC and owned by Amazon.

Without understanding the placement of the AWS instance within the cloud, it is very difficult to be certain where the real service is located. In order to compare the impact of the Zoom audio codec, a spectral plot of both the sent and received audio signals was taken using iZotope 8 Audio Editor. Appendix IV shows the two plots.

Commentary on Results

Absolute Latency

Putting aside Run 3, the most obvious outcome is the very high latency in the communication. One must first remember that this is a 2-way latency and therefore to

compare with the ITU-T G.411 recommendation of 150ms one-way delay, the number must be halved. The virtual audio cables certainly introduce some delay, but the USB audio card also introduces considerable delay.

In order to arrive at a "real-world" figure for Mouth-to-Ear latency where two USB audio cards are used across a Zoom link, the following formula was adopted:

(One way delay due to Zoom) = [(Measured Latency) – (160ms + 192ms)]/2

As each USB card introduces 140ms of latency, it can be shown that a good approximation at "real-world" latency would be the above one-way delay plus 140ms.

Working on the average delay across Runs 1,2,4 and 5 of 532.1ms:

One way delay due to Zoom = (532.1 – (160 +192))/2 = 90.05ms

However, the "real-world" delay is heavily impacted by the sound card latency of 140ms:

Real-world delay = 230.05ms

There is another aspect in the latency relevant to music-making which needs to be considered, the jitter. Putting aside Run 3, the smallest latency which was observed was 519ms and the largest 544ms. Using the above formulae, this shows real-world latencies of **223.5ms** and **236ms** respectively, a variance of **12.5ms**. We know that 4ms of this jitter can be attributed to jitter to the Zoom server leaving a further 8.5ms as associated with the local conditions.

Running a "ping" test from PC A to the local router, 192.168.1.1 shows a consistently low result of 1ms (upper bound 1.5ms, lower 0.75ms). This PC is directly connected to the router over 1Gbit/s Ethernet. Running a "ping" test from PC B to the local router shows more variable results from 4ms to 8ms. This PC is connected over a 5GHz WiFi network, albeit less than 2 metres from the access point. It is fair therefore to conclude that a considerable portion of the delay and jitter is due to WiFi.

Run 3

The results from Run 3 are worth considering both from the absolute latency they display but also from the fact that this is the middle set of tests in a continuous single Zoom session. The latency here is obviously unusable in any scenario, but the question has to be why the sudden jump between Run 2 and then back to the expected result in Run 3. The only conclusion one can draw is that Zoom have implemented some dynamic allocation of conferencing resources such that if a period of silence is detected, they are deallocated and take some finite time to reallocate leading to an extended delay. What is most surprising is that the delay is consistent – there is no attempt to "catch-up" after the first lengthy pulse. Whilst this may be acceptable for speech as the parties would simply adjust to the delay, for music, this is disastrous.

Audio Quality

With reference to the spectral plots of sent and received audio, one observes that the fidelity is quite good until about 4kHz when the waveform tails away quite rapidly. Whilst this is quite high in terms of musical pitch – 4kHz is around C_8 – harmonics add

considerable colour and depth and consequently any loss of "top" will make the audio sound dull. A high-quality music system will maintain the audio response sometimes as high as 22kHz – FM broadcast radio in the UK is specified to 15kHz.

Applicability to Music Making

At this point one must remember that in terms of making music interactively, the upper-bound on mouth-to-ear latency is 25ms. Even with perfect, zero-latency sound cards, it can be seen that this is not achievable. In terms of the earlier discussion around delay and its relation to distance, an average delay of 230.05ms is the same as trying to interact with a musician approximately 230ft away.

However, this isn't the only problem. With the use of good-quality audio interfaces, the sound-card latency can probably be managed down to a few milliseconds but the variation in latency due to both the server link and the Wi-Fi connection could add a variance of 12.5ms, the same as moving the players apart up to approximately 12ft! And this is variable – it is more like a marching band where everyone is marching in different, and random, directions…

Moreover, it appears that even when connected to a known datacentre, there is no guarantee that the latency will be kept – run 3 suggests that the latency could jump to very high levels at any time. Certainly, for interactive, collaborative music-making, it is very clear from these results why business-grade conferencing systems cannot provide the quality required.

Conclusions

This paper has sought to both to explain why the current Internet model will struggle to deliver live, interactive music-making and further show why the tools which would appear to be able to offer such a service cannot. Whilst it appears that the goals of business videoconferencing and music-making would be similar, the physics of sound transmission and its relationship to distance plus the need for two-way communication mean that in many respects, on-line music-making forms a very different and quite demanding use-case.

The real-world experiments pinpoint the sources of latency in a typical Zoom session allowing further study into these effects. Known issues such as Wi-Fi connections and audio-card latency can be managed – however, the "black-box" nature of the business conferencing systems will preclude the level of tuning required and new solutions need to be found.

The author is encouraged by results showing less than 1ms jitter over a wired connection and 4ms to the Zoom data-centre – whilst this still represents a movement of 5ft between players (coupled with an average 25ms latency to the server), this is only just outside of the bounds of acceptability and careful positioning and sizing of the server coupled with more reliable and lower-latency conferencing software offers a glimmer of light and one can look to the hardware-based solutions such as the JackTrip Virtual Studio (JackTrip Foundation 2021) devices as potential solutions although they come with the

requirement to use wired Ethernet and a standalone box that requires a deeper level of technical knowledge than most musical groups would have including having to procure and configure a suitably-placed server. A delay of 25ms+ may be acceptable for certain types of music or activities – slow "note-bashing" for example – and even post-pandemic there could be good use-cases for distance learning that would leverage such technologies.

References

Audacity. 2021. "Home". *Audacity* ®. https://www.audacityteam.org/.

Baran, P. 1964. "On Distributed Communications Networks". *IEEE Transactions on Communications* 12 (1): 1-9. doi:10.1109/tcom.1964.1088883.

Broersma, Matthew. 2004. "BT's 21St Century Network Get Underway". *Computer Weekly*.

Chafe, Chris, Michael Gurevich, Grace Leslie, and Sean Tyan. 2004. "Effect of Time Delay on Ensemble Accuracy". *Proceedings of the International Symposium on Musical Acoustics*.

Davies, Donald. 1966. "Proposal for A Digital communication Network". National Physical Laboratory

Gray, Robert M. 2010. "A Survey Of Linear Predictive Coding: Part I Of Linear Predictive Coding And The Internet Protocol". *Foundations And Trends® In Signal Processing* 3 (3): 153-202. doi:10.1561/2000000029.

ITU-T. 2003. "G.114: One-Way Transmission Time". *Itu.Int*. https://www.itu.int/rec/T-REC-G.114.

ITU-T. 1988. "ITU-T G.711 (11/1988)". *ITU*. http://handle.itu.int/11.1002/1000/911.

ITU-T. 2012. "ITU-T G.729 (06/2012)". *ITU*. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11675.

JackTrip Foundation. 2021. "Jacktrip Foundation". *Jacktrip.Org*. https://www.jacktrip.org/studio.html.

RADVision. 1997. "Radvision And Intel Target Compatibility Between Radvision's H.323/320 Videoconferencing Gateway And Intel's Business Video Conferencing And Teamstation Products". Voip Developer Solutions. Tel Aviv, Israel.

RAND Corporation. 2021. "Paul Baran And The Origins Of The Internet". *Rand.Org*. https://www.rand.org/about/history/baran.html.

Ricksci. 2021. "Using A Graph Of Sound Strength". *Ricksci.Com*. http://www.ricksci.com/phy/phya_sound_strength_graph.htm.

Sengpiel, Eberhard. 2021. "Time Difference Per Sound Path Distance Ms Per Metre or Length Millimetres Time Of Arrival Milliseconds Calculation Calculate Delay Line Noise Sound Wave In Air Calculator Variance ITD Haas Effect Duration - Sengpielaudio Sengpiel Berlin". *Sengpielaudio.Com*. http://www.sengpielaudio.com/calculator-soundpath.htm.

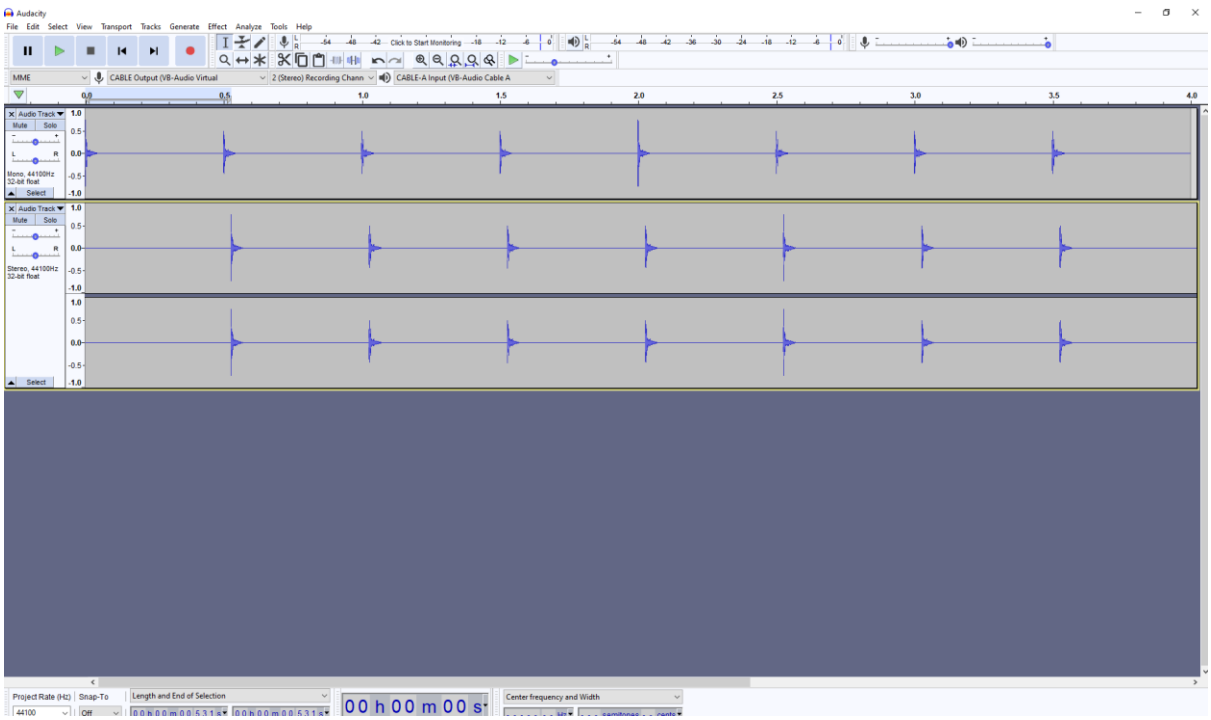
Zoom. 2020. "Coming April 18: Control Your Zoom Data Routing - Zoom Blog". *Zoom Blog*. https://blog.zoom.us/coming-april-18-control-your-zoom-data-routing/.

Zoom. 2021a. "Video Conferencing, Web Conferencing, Webinars, Screen Sharing". *Zoom Video*. https://www.zoom.us/.

Zoom. 2021b. "Enabling Option To Preserve Original Sound". *Zoom Help Center*. https://support.zoom.us/hc/en-us/articles/115003279466-Enabling-option-to-preserve-original-sound.

Appendix I – End-to-End Loopback Tests

Run 1



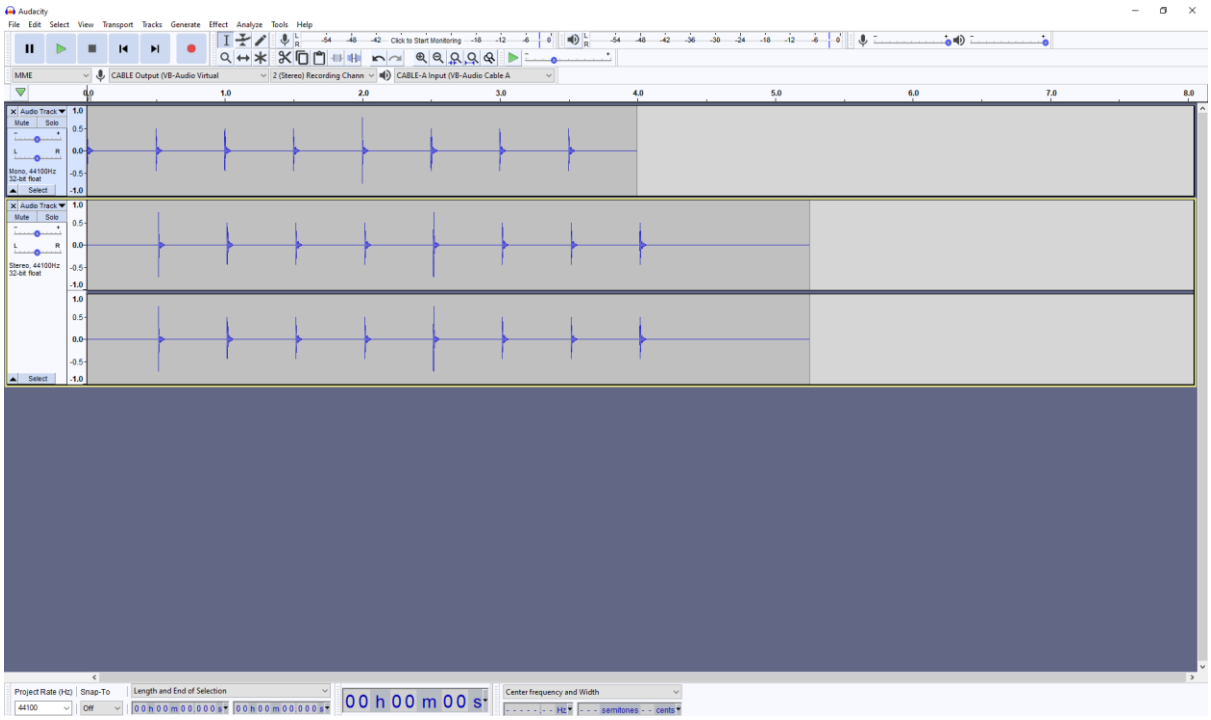
(Upper track, sent audio pulse; lower track, received audio pulse)

Latencies

Pulse 1	527ms
Pulse 2	529ms
Pulse 3	530ms
Pulse 4	526ms
Pulse 5	528ms
Pulse 6	527ms
Pulse 7	527ms
Pulse 8	528ms

Average, 527.75ms. Jitter 4ms

Run 2

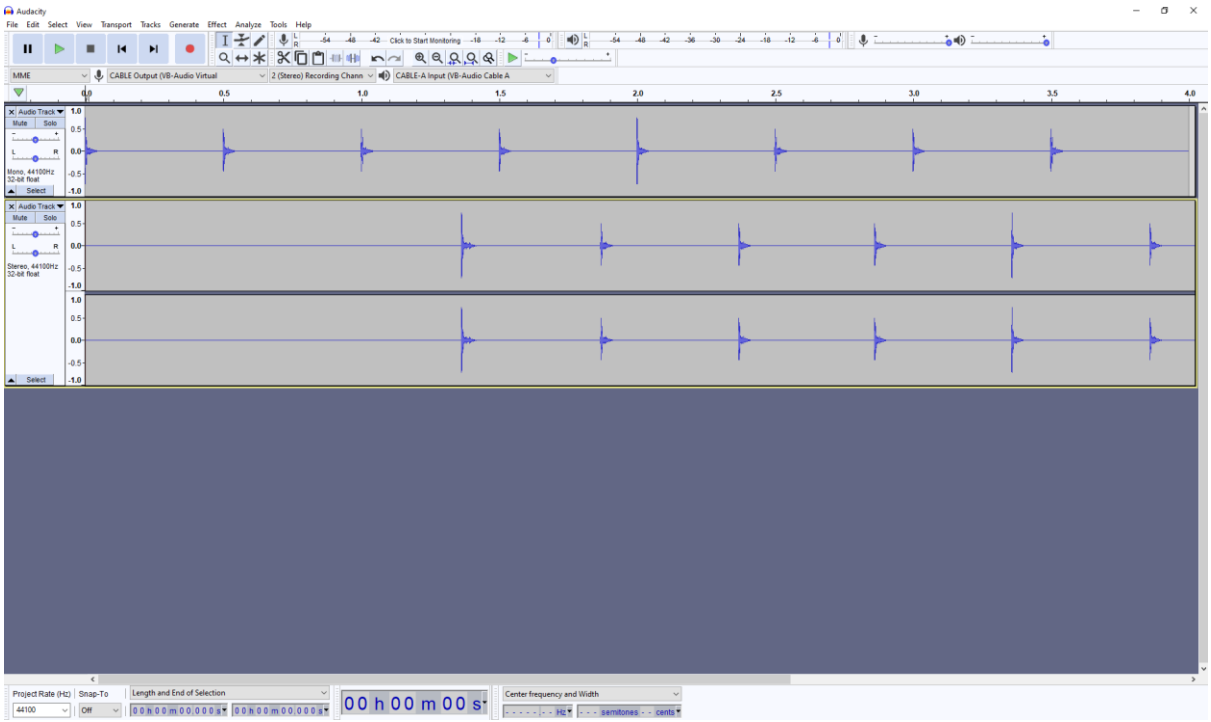


(Upper track, sent audio pulse; lower track, received audio pulse)

Pulse 1	519ms
Pulse 2	520ms
Pulse 3	520ms
Pulse 4	520ms
Pulse 5	519ms
Pulse 6	520ms
Pulse 7	520ms
Pulse 8	520ms

Average, 519.75ms. Jitter 1ms

Run 3

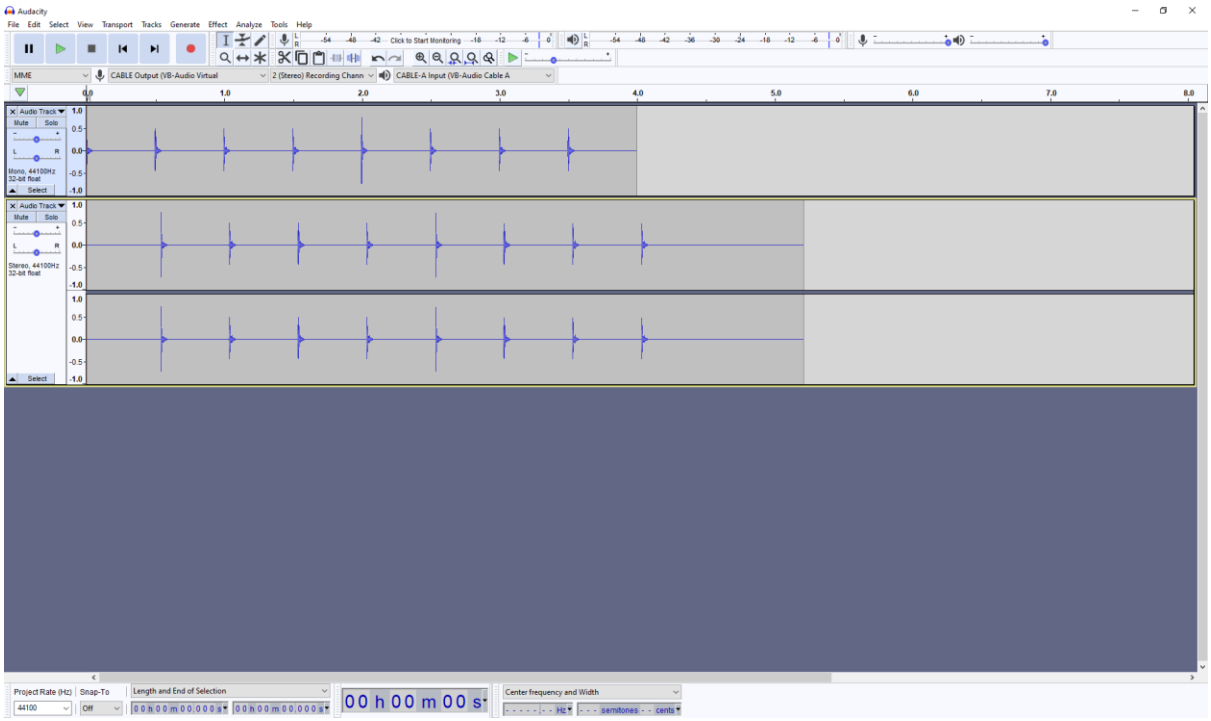


(Upper track, sent audio pulse; lower track, received audio pulse)

Pulse 1	1363ms
Pulse 2	1369ms
Pulse 3	1369ms
Pulse 4	1363ms
Pulse 5	1358ms
Pulse 6	1358ms
Pulse 7	1338ms
Pulse 8	1339ms

Average, 1357.125s. Jitter 31ms

Run 4

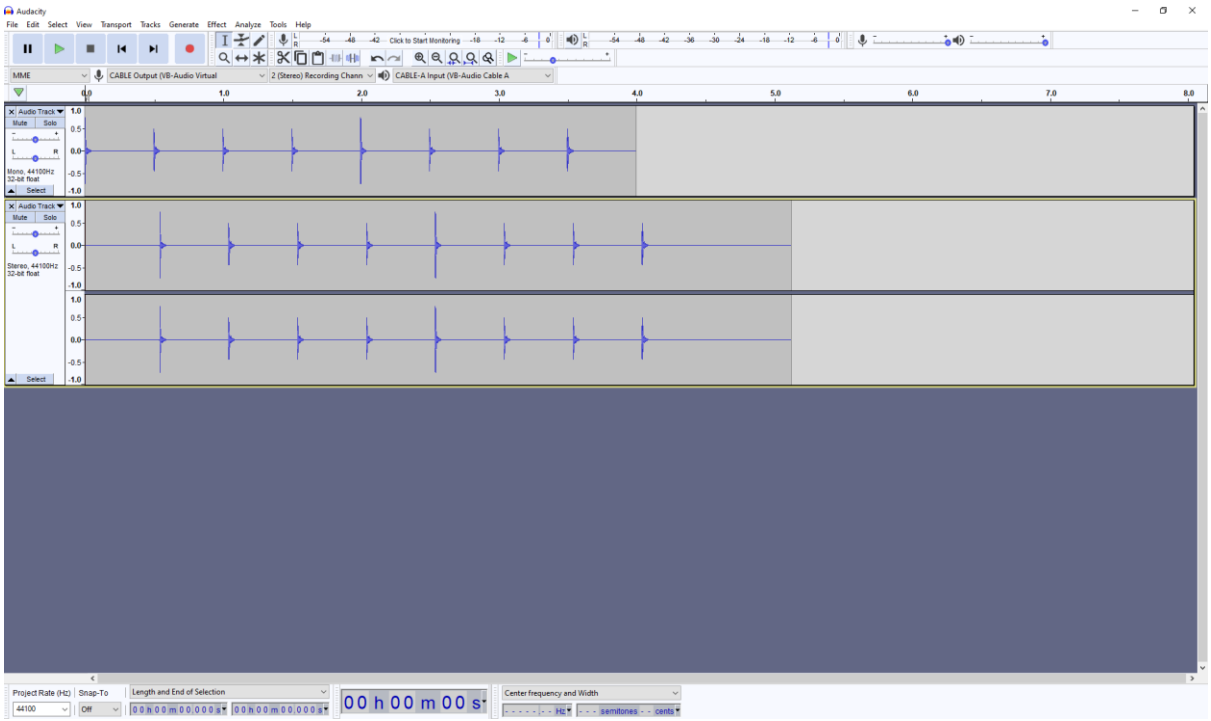


(Upper track, sent audio pulse; lower track, received audio pulse)

Pulse 1	540ms
Pulse 2	542ms
Pulse 3	540ms
Pulse 4	540ms
Pulse 5	541ms
Pulse 6	537ms
Pulse 7	532ms
Pulse 8	534ms

Average, 538.25ms. Jitter 10ms

Run 5



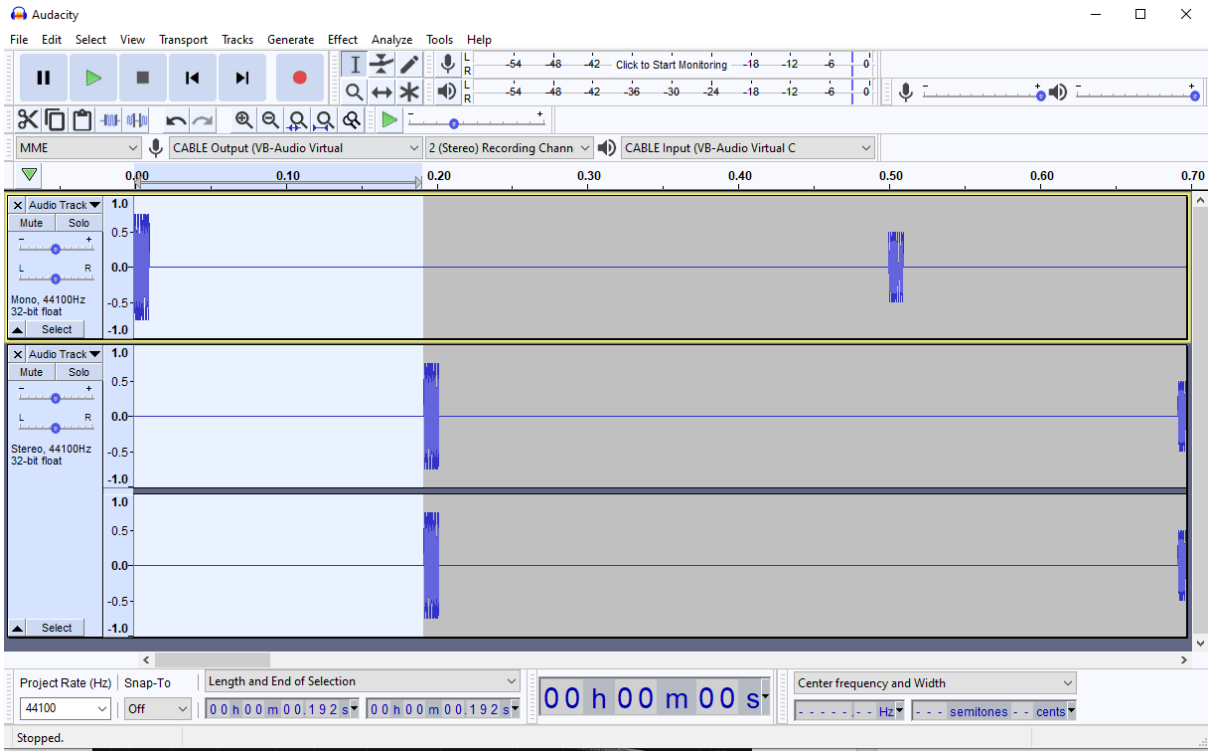
(Upper track, sent audio pulse; lower track, received audio pulse)

Pulse 1	544ms
Pulse 2	542ms
Pulse 3	542ms
Pulse 4	544ms
Pulse 5	543ms
Pulse 6	542ms
Pulse 7	542ms
Pulse 8	542ms

Average, 542.625ms. Jitter 4ms

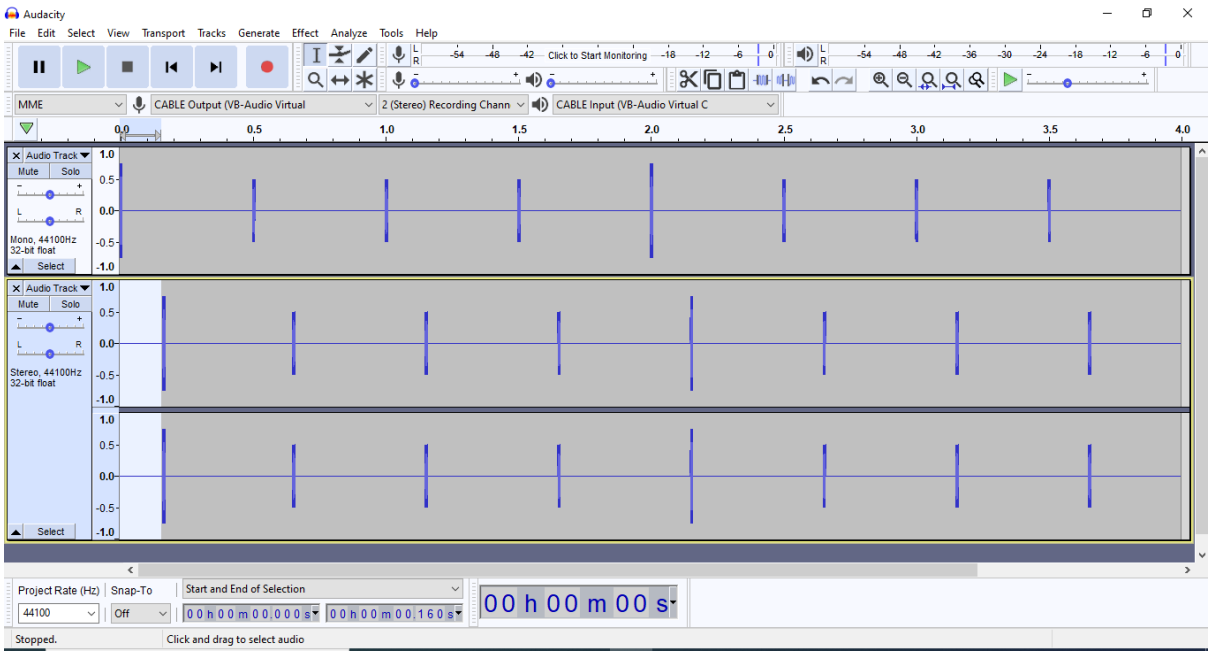
Appendix II – Cable Loopback Tests

PC A



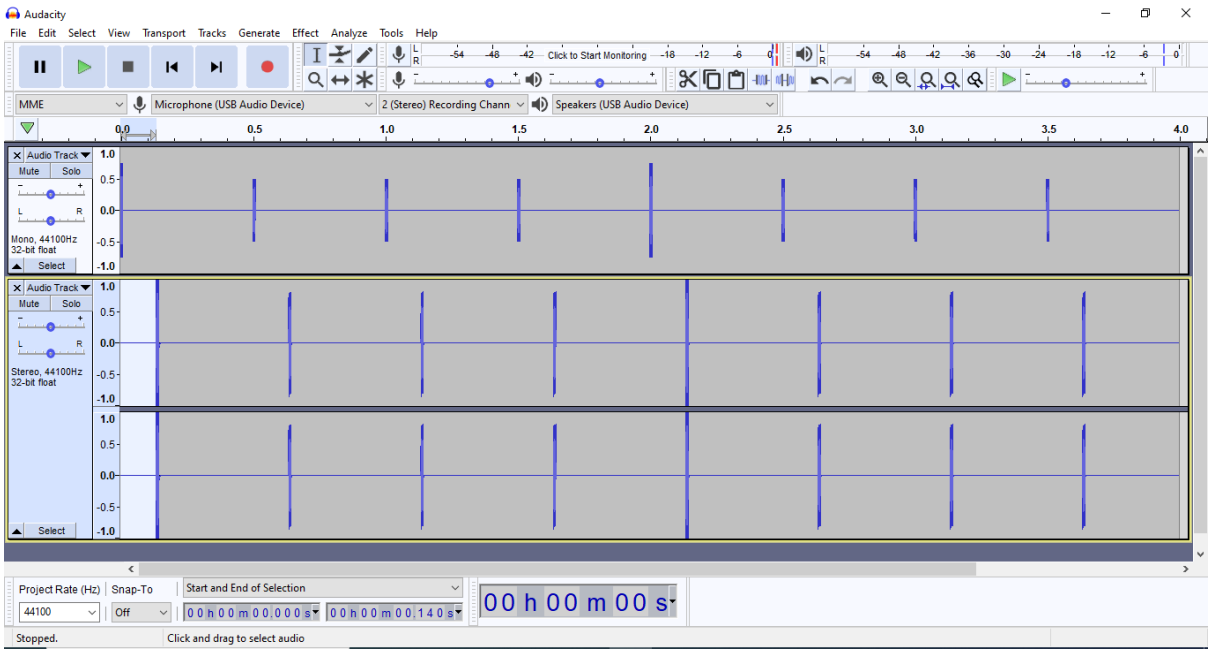
Constant Latency, 192ms *(Upper track, sent audio pulse; lower track, received audio pulse)*

PC B



Constant Latency, 160ms *(Upper track, sent audio pulse; lower track, received audio pulse)*

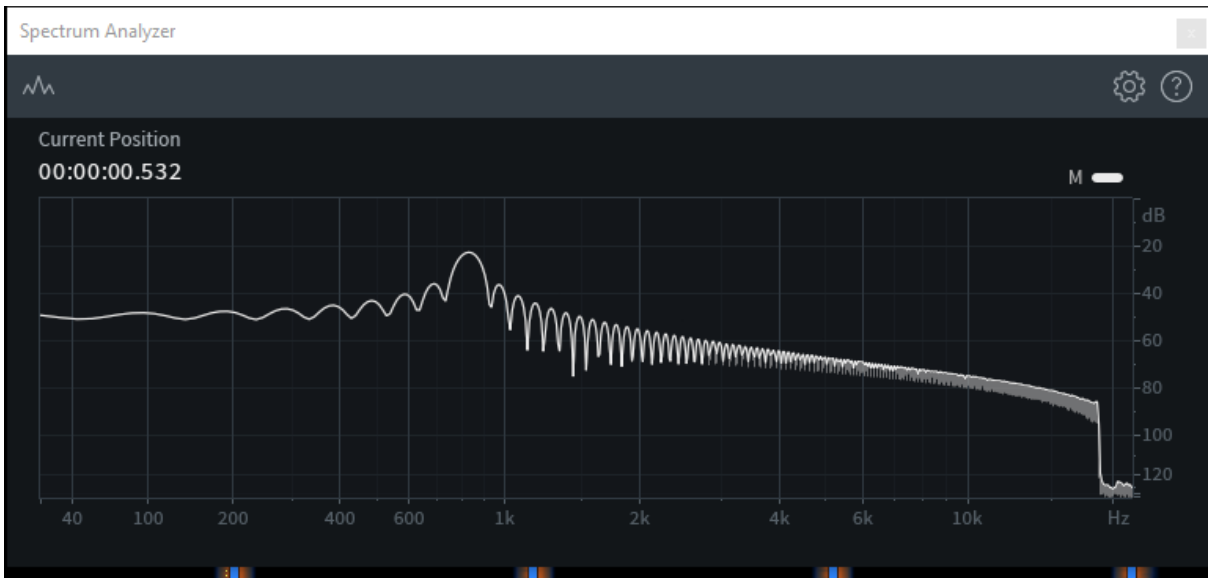
Appendix III – USB Audio Adapter Loopback Tests



Constant Latency, 140ms *(Upper track, sent audio pulse; lower track, received audio pulse)*

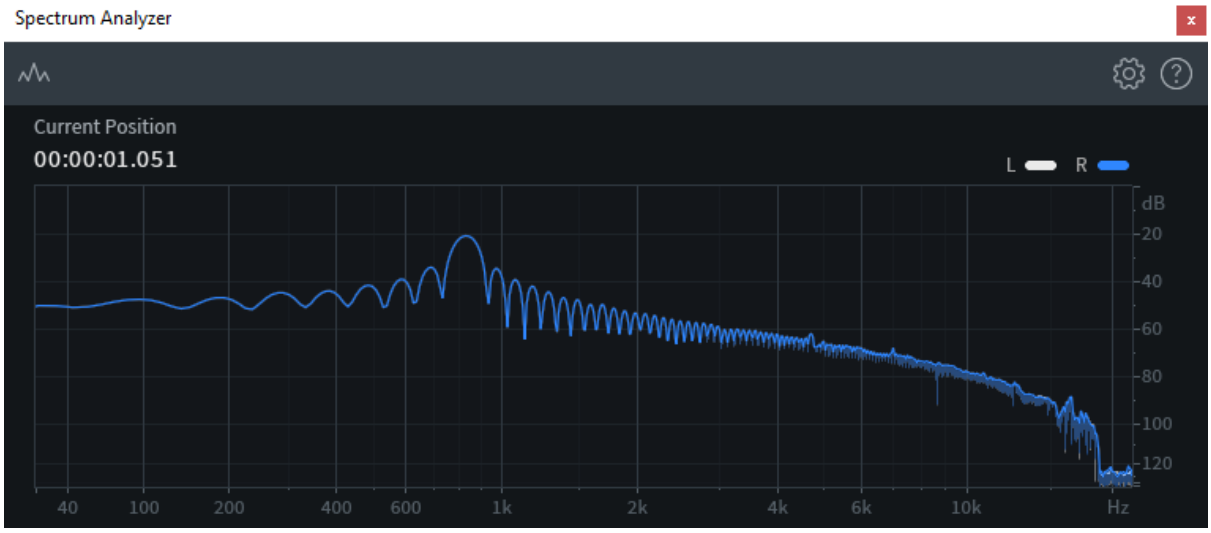
Appendix IV – Audio Spectral Plots

Sent



(Frequency response plot of sent audio)

Received



(Frequency response plot of received audio)