Telematic performance and the challenge of latency

Michael Rofe | Federico Reuben

Abstract

Any attempt to perform music over a network requires engagement with the issue of latency. Either latency needs to be reduced to the point where it is no longer noticeable or creative alternatives to working with latency need to be developed. Given that Online Orchestra aimed to enable performance in community contexts, where significant bandwidth and specialist equipment were not available, it would not be possible to reduce latency below the 20–30ms cut-off at which it becomes noticeable. Instead, Online Orchestra developed new software that controls network latency, locking it to musical tempo. This in turn enabled musicians to perform precise rhythmic music in a latency-rich environment.

Keywords

Online Orchestra; telematic performance; latency; tempo synchronisation; perceiving simultaneity ; latency measurement

Introduction

The key challenge for any telematic performance is the issue of latency. Broadly defined, latency is the time delay between the input and output of a system. In the context of telematic music-making, this can be understood as the delay between the moment a musician in one location makes a sound and the moment a second musician in a different location hears that sound.

All data transfer involves some degree of latency, and in the world of telecommunications, its effect becomes perceptible in the short delays, and occasional overlaps, that we all experience in mobile telephone or Skype conversations. Telematic conversations are at times not as fluid as their real-world counterparts, but the latency is sufficiently low that communication is perfectly possible. However, when attempting to make music over a network, latency becomes a significant problem. Music can be conceived of as a series of temporally coordinated sound events: musicians might be asked to 'play together', for instance, or to perform materials that have specific rhythmic relationships to other parts of a musical texture. The more complex these rhythmic relationships, the more crucial that every aspect of a musical performance is realized precisely *in time*. But in any telematic performance, there will be some degree of latency, as data are processed and sent between locations. If that latency is noticeable, rhythmic relationships articulated in a score by a composer will be deformed by the very nature of the environment: the music will appear *out of time* despite the best efforts of the performers.

Perceiving simultaneity in musical performance

Any traditional musical performance involves small degrees of timing delay. A trumpet, for instance, produces a sound infinitesimally after its performer starts to vibrate their lips within the mouthpiece. If that performer is following a conductor and score, he or she will play their note on cue, meaning an additional decision-making process takes place in the mind of the performer about when to initiate their note: the more experienced the performer, the more able they are to judge that delay, as they anticipate the conductor's beat such that their sound is produced at precisely the right moment; less-experienced musicians might not judge this so well, giving rise to a note that is slightly late. Conductors do their best to train and coordinate musicians in this regard, but listeners are also rather forgiving: how 'together' two notes actually need to be realized in order to be considered 'simultaneous' is a matter of some complexity (see Repp and Su 2005). And where large venues are involved, there is an additional latency as sound travels around the room, causing small phase differences (see Gade 1990).

Up to a point, therefore, musicians are used to working with small time delays in music, and listeners are used to hearing and absorbing/ignoring small amounts of delay within a performance. The key issue is to establish the point at which a delay becomes so large that coordinated performance is no longer possible, and the perception of simultaneity in the minds of the listener and performer breaks down. Studies have come to varying conclusions. Bartlette et al. established, in a test involving two Mozart duets, that a latency of anything over 100ms was enough to render the experience either unmusical or non-interactive (Bartlette et al. 2006: 49). Others have observed a much

lower cut-off: Chafe et al., for instance, observe disruption at <11.5ms in the form of overall tempo deceleration: each performer waits slightly for the other, creating a gradual deceleration in tempo overall (Chafe et al. 2004: 1). More extensive studies tend to agree that latencies of between 20 and 30ms are where notable disruptions to performers start to occur (see Schuett 2002); this corroborates seminal psychoacoustic research by Haas, which suggests that the threshold for hearing two events as simultaneous is in the region of 25–30ms (Haas [1949] 1972: 145).

As part of working groups during the design phase of Online Orchestra, a simple experiment was undertaken to test latency perception. Four undergraduate musicians were split into two rooms connected with analogue audio feed. A controllable artificial delay was added to the audio signal path being sent between rooms. Participants were asked to clap together at 120bpm and then report on the difficulty of that task when different levels of artificial delay were added to the signal. Participants found this task increasingly difficult as delay was added, and, in line with Schett's findings, latency over 30ms rendered the task impossible. However, at points where the delay intersected with the 120bpm tempo of the clapping (or intersected as an integer multiple or divisor of it), then the task became achievable – participants could in effect clap 'in time' with the other node. For instance, at a regular pace of 120bpm, clapping on the beat, a delay of 500ms caused the feed of the second node to sound on the half beat with respect to the live clap of the first node, and vice versa. As such, participants were able to settle into this rhythmic relationship and perceive it as a musical pattern rather than a latency: Chafe and Cáceres describe this technique as 'feedback locking' (Chafe and Cáceres 2010: 183). Subsequent discussions with the participants highlighted this important finding: a

very small amount of latency was enough to disrupt musical performance, but much higher latencies could be managed as long as they intersected with the tempo of the musical materials. Online Orchestra's software solution, described below, was rooted in this underlying principle.

Contributors to latency in telematic performance

Given 30ms as a point at which performers seem to be significantly affected by latency, an obvious solution to the issue of latency in telematic performance is to design a system architecture in which latency consistently falls below this threshold. However, this approach is complicated by the wide-ranging factors that give rise to latency.

Distance and the speed of light

Data are sent over the Internet largely as pulses of light through a fibre optic network; light travels at a maximum speed of 299,792,458m/s. The greater the distance light has to travel, the longer it takes. For instance, the distance from London to New York is 5567km, meaning a single pulse of light takes an absolute minimum time of 18.57ms to make its journey. In reality, the fact this beam of light is being propagated through a narrow cable adds significant resistance, reducing the speed as it travels.

Network infrastructure

Rare is a network connection that takes place entirely over fibre optic cable. Often, the first and last stretches of the journey take place over copper cable, wifi or mobile 3G/4G networks, all of which have significantly slower transmission speeds. And even over faster, wired connections, routers, firewalls, switches and exchanges each themselves add latency, increasing the transmission time of data – meaning that data never actually travel at the speed of light through the Internet.[1]

Bandwidth

Bandwidth – and hence speed – available at different locations varies widely. Sometimes this results from the types of service for which a user is willing and able to pay, and at times, this is more structural (e.g. fibre-to-premises might not be available). In the United Kingdom, high-speed services such as Janet run on a 100Gbps backbone but require specialist cabling, of which there is only currently about 5000km in the United Kingdom, thereby only delivering this type of bandwidth to specialist institutions. Commercial internet service providers (ISPs) in the United Kingdom (at the time of writing) claim to be able to deliver up to 300Mbps download, though the reality is often much slower in the majority of locations, as indeed was the case in Online Orchestra's chosen locations, as discussed below.

Network traffic

The number of users online at any given time, and the amount of bandwidth they are using within the overall network, affects latency. For instance, the UK network can be up to 30 per cent slower during peak usage times of 7–11pm, as discussed below.

Data size

The size of data being sent over a network affects the time taken to make the journey. Large data are broken into packets, and these packets are sent separately and recombined at destination. This all takes time. Receiving a text-only e-mail requires a single download of no more than 0.1Mb; watching HD video requires downloading at least 4Mb of data per second.

System latency

A full communications system involves more than the propagation latency involved in sending and receiving data over the network. For instance, in the case of audio-only transmission, there are many steps between the moment one individual says something and the moment someone somewhere else hears it. Sound is captured by a microphone, is converted from an analogue to a digital signal, is compressed, is split into packets and these are then sent over the Internet. The packets pass through routers, firewalls, wires and exchanges, perhaps also over wifi and/or 3G/4G. Then, at destination, packets are reassembled, converted back to analogue, sent down more wires to speakers and ultimately across the room to the listener’s ears. Latency is introduced at every step.

Computer processing power

Significant computing power is needed to achieve all of the above. Higher speed computers process data more quickly, and so enable lower latencies.

Latency in community contexts

So the aim of optimizing a multinodal system such that the overall latency remains consistently below 30ms is a challenging, and indeed expensive, enterprise. It is possible, however: the LOLA research group, for instance, outlines in detail in their user manual the various hardware, software and network requirements needed to enable a sub-30ms performance, and this has resulted in a number of highly successful telematic performances when these conditions are implemented (see LOLA 2015).

As detailed in the first article in this special issue (see Rofe et al. 2017b), the key aim for Online Orchestra was to deliver telematic performance opportunities into community venues and to do so at as low a cost to the user as possible. The key limitation of LOLA is its bandwidth requirements. The LOLA manual makes clear the need for at least a 1Gbps connection (LOLA 2015: 8), but, as described above, such speeds are only currently available in the United Kingdom on specialist networks such as Janet, which in turn are only available at specialist research institutions. By comparison, the four locations used for the Online Orchestra pilot performance had bandwidths as detailed in Figure 1: significantly lower than that required by LOLA.

Figure 1: Network bandwidths of the four Online Orchestra nodes.

Falmouth University (Janet)	Truro Cathedral (BT Super Fast Broadband – FTP)	Mullion School (BT Super Fast Broadband – copper on final stretch)	Five Islands' School (BT Super Fast Broadband – FTP)
10Gbps	73Mbps down	27Mbps down	41Mbps down
	18Mbps up	7Mbps up	8Mbps up

Likewise, Online Orchestra's aim to limit cost to users meant a need to design a system architecture that did not rely on specialist equipment that users might not already own. Although distances between chosen locations in the pilot performance were quite modest (maximum 100km), the longer term ambition was to find a solution that would enable telematic performance between locations anywhere in the world, meaning speed-of-light limitations might become a factor in future performances; LOLA only enables sub-30ms connections within a radius of roughly 3000km.

Online Orchestra's approach to latency

Given the technical limitations inherent to working in community contexts – in particular the available bandwidth – it would not be possible to design a system that consistently had a latency below 30ms. As such, Online Orchestra instead took the approach of investigating ways of working *with* latency, rather than trying to eliminate it; to embrace it as a creative opportunity within the online environment, rather than an impediment to telematic performance. And as the findings of the latency-perception experiment (described above) suggested, performers seemed able to make music more easily in latency-rich environments when latency and musical tempo are linked: where the musical

materials are performed at a tempo that is equal to, or an integer multiple or divisor of, the duration of the latency.

Making music in which the tempo synchronizes to the latency has a number of precedents in the history of telematic performance. Examples include the *Net vs. Net* collective (see Cáceres and Renaud 2008), *Ping* (see Traub 2005: 464), *Ninjam* (see Driessen et al. 2011) and a telematic performance of Terry Riley's *In C* (see Cáceres et al. 2008; see also Farner et al. 2009). However, these and similar performances tend to adopt the approach of performing musical materials 'in time' with the latency inherent in the system during a given performance. Online Orchestra aimed to enable the opposite – to specify the latency in line with the desired tempo of a given piece of music. This latency control would need to overcome three factors:

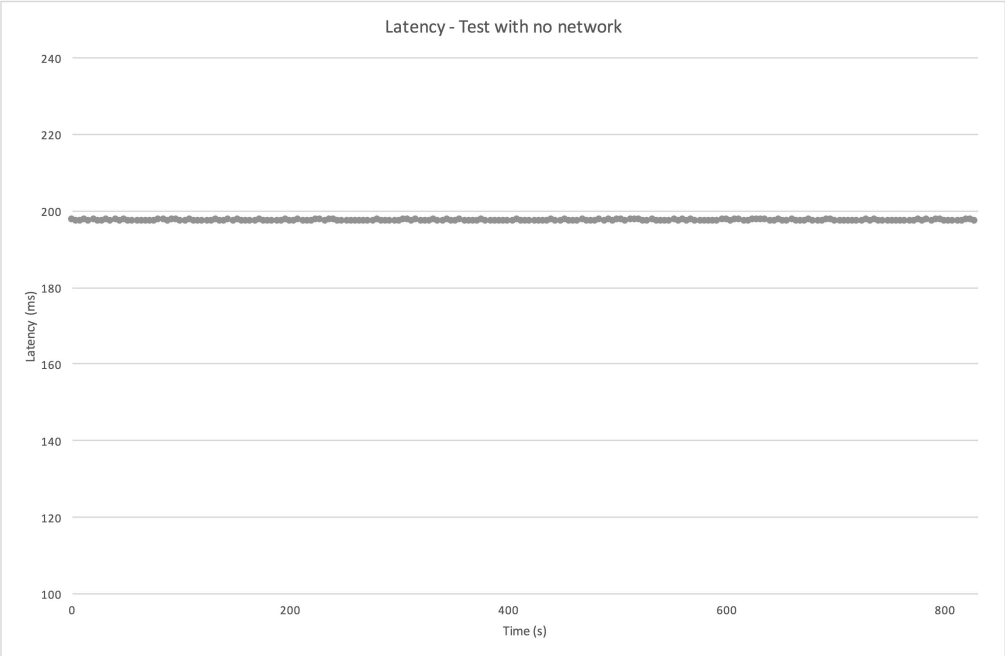
1. In a multi-node performance (i.e. when more than two nodes are connected), each connection will likely have a different latency, as a consequence of differing bandwidths and differing user equipment. Any multinodal performance would require that these differences be eliminated.
2. Latency is constantly changing, both at the macro-level (e.g. times of day) and at the micro-level (short-term jitter). To perform 'in time' to the latency requires stabilization of the latency at the macro- and micro-levels.
3. In order to perform music at tempi that are different to the inherent latency, it would be necessary to be able to determine an exact duration of that latency, rather than work with whatever latency happens to be present within a given system at a given time.

Audio latency measurement tool

Given that it would be difficult to calculate the specific latency at each individual stage within the system (and indeed in future performances precise equipment and network conditions might not be known), a simple yet effective measurement tool was developed to determine the total latency in the audio signal. A programme was developed to measure the round-trip trajectory of an impulse between server and client. The system produces a percussive sound synthetically through an impulse and sends it through the network. The sound is detected when it gets back using an onset detection algorithm.[2] The time is measured between the generation of sound and its detection as it arrives back from the round-trip trajectory. The measured time is then divided by two, to arrive at the total latency of the audio between server and client.

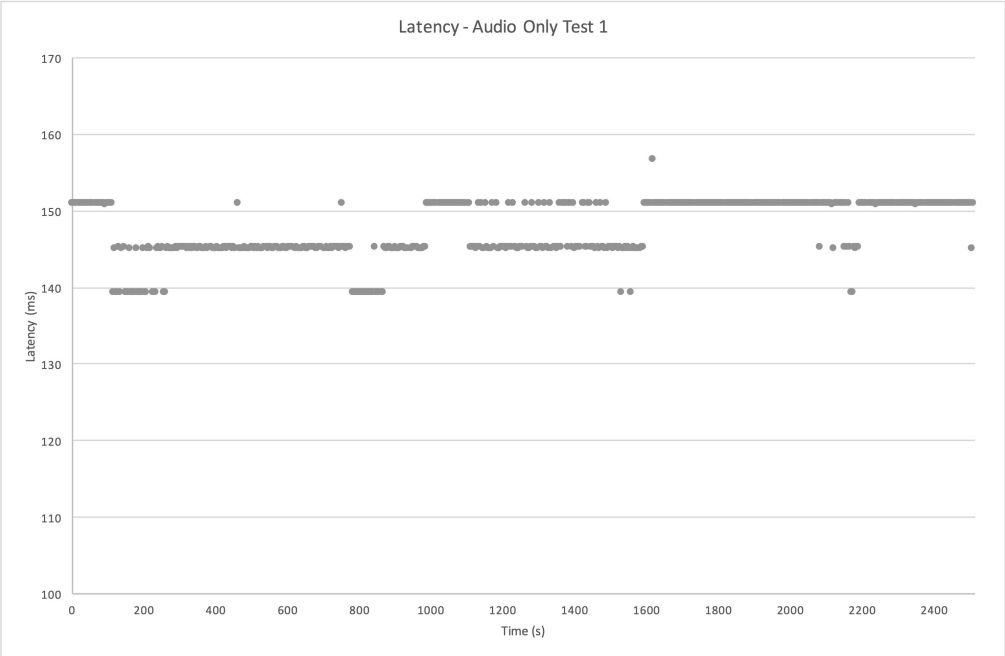
Figure 2 shows data produced by this measurement tool with no network conditions. An artificial delay line was used between the generation of the sound and its detection without sending it over the Internet. This test was conducted to evaluate the precision of the onset detection algorithm and timer. The results show a highly constant delay time and consistent onset detection with a range (maximum to minimum values) of only 0.411ms and a standard deviation of 0.066ms.

Figure 2: Test of audio latency measurement tool with artificial delay.



Latency tests were then conducted with the measurement tool to estimate the amount of latency between server and client and how that latency changed over time. Different network conditions were tested to see how bandwidth use might affect audio latency. The tests were conducted using high-quality broadband connections within Falmouth University. Initially, the latency produced solely by the transmission of audio data was determined, without the added bandwidth from video streaming. This was achieved by transmitting audio through JackTrip[3] and measuring the latency every four seconds for approximately 40 minutes. Figure 3 shows the results, which are fairly stable and seem to lock at three main values: 139ms, 145ms and 151ms. In this test, the total latency average is 147.534ms, the range is 17.477ms and the standard deviation is 3.652ms. The data also show changes of approximately 6ms between values.

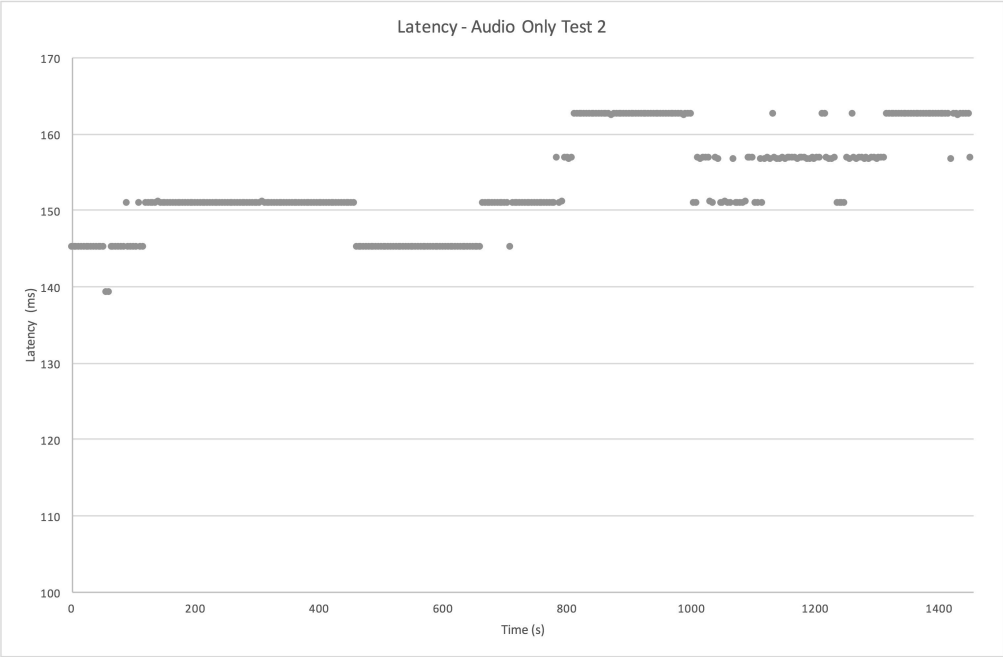
Figure 3: Audio transmission test 1.



Because of varying conditions across the network, it was expected that the amount of available bandwidth would change during the day depending on network traffic. Therefore, the same test (audio transmission only) was run at a moment during the day when an increase in bandwidth use was expected. Figure 4 shows the results of this test, demonstrating a gradual increase in audio latency after approximately thirteen minutes. In this test, the latency was measured every four seconds for approximately 24 minutes. Even though the time sample was considerably smaller than the previous test, the audio latency average (153.30ms), range (23.314ms) and standard deviation (6.295ms) show a considerable increase. The data also show that while the latency tends to lock at higher values (151ms, 157ms and 163ms) in the second test, the interval of approximately 6ms between the points of stability remains. It can also be seen that while

in the previous test, the value of approximately 151ms is the stable point at the higher range, it changes in the second test after thirteen minutes to approximately 163ms – a difference of 12ms between the two. These two tests demonstrate that, even in a high-quality broadband connection, network traffic can have a considerable effect on the total system latency of an audio transmission between server and client.

Figure 4: Audio transmission test 2.



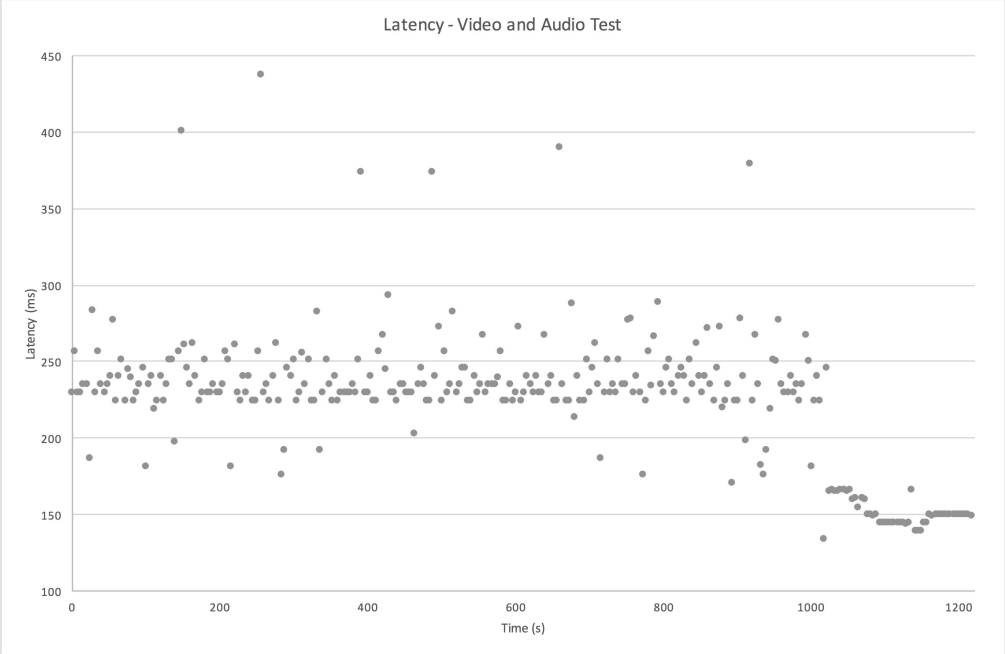
After observing the latency patterns and behaviour of audio streaming with varying network traffic, video data were added to the transmission: it was crucial to test how adding video streaming to the same connection would affect audio latency and its rate of change over time. In addition to a JackTrip connection, peer-to-peer video streaming was enabled using a browser-based application.[4] Figure 5 shows the results of this third test, where a video connection is established for the first seventeen minutes and

then disabled for approximately the last three minutes, while the JackTrip transmission is constantly running. The video connection was stopped before the end of the test to show the contrast in audio latency from when video and audio data are being transmitted simultaneously, to when only audio data are being transferred. From the results, it is evident that video streaming not only adds significant audio latency but also has a considerable impact on how latency changes over time. In the period of time that both audio and video data are transmitted, the average audio latency is 238.72ms, the range is 303.996ms and the standard deviation is 31.753ms. In contrast, the last set of data collected, which only shows data from the audio transmission, has an average latency of 151.102ms, a range of 26.854ms and a standard deviation of 8.308ms. The video data therefore added an average of 87.618ms to the audio latency, 277.142ms to the latency range and 4.899ms to the standard deviation.

It is also apparent that the latency patterns and behaviour distinctively change when video data are added to the transmission. When the video is running, the 6ms interval clearly observed in the previous (audio only) tests disappears. Once the video streaming is disabled, the 6ms interval re-emerges, as well as the patterns of stability observed in the previous tests. While the patterns in the audio-only tests can be characterized by their discernible points of stability, the added video results show a more unstable and unpredictable behaviour. When video and audio data are transmitted simultaneously, some patterns can still be identified roughly, but they are more arbitrary and complex. It can be observed that the highest values differ considerably from the average latency – these values can be from 135ms to 199ms apart from the average latency value. Even if these exceptionally high values are excluded, the rest of the values

are within a range that exceeds 100ms. The results from this test show how the additional video data transmission not only adds a considerable amount of audio latency but also produces a more substantial and unpredictable change in latency over time.

Figure 5: Video and audio transmission test (audio transmission only after 1024 seconds).



Audio system design considerations

After the results of the latency measurement tests were evaluated, it was determined that audio latency change over time had to be considered in the design of the audio system for Online Orchestra. Because of the star topology of the audio-streaming system (see Prior et al. 2017a), the total audio latency between two nodes would be the addition of the latency of two peer-to-peer connections: ‘total node-to-node latency (node 1 to node 2) =

latency of peer-to-peer connection 1 (node 1 to server) + latency of peer-to-peer connection 2 (server to node 2)'. It was estimated that the latency between two nodes would be at least twice as much as the latency measurements from the above tests (tests were conducted in a very good connection so it was anticipated that there would be more latency in the other locations). Moreover, because each connection would have different network conditions, it could be expected that the latency between nodes would vary. In other words, the total latency between node 1 and node 2 would be different than that from node 1 to node 3 (and node 2 to node 3) because of the different connection conditions in each location. The change in latency over time could also be significantly different for each node-to-node stream.

Small differences in latency between connections can also alter the characteristics of a sound as it is heard back through the loudspeakers in the other locations. Small phase differences can result in various transformations of the signal that affect the timbral, spatial and rhythmic perception of a sound. Phase differences of 1–20ms can cause constructive and destructive interference in the signal affecting timbral qualities of a sound. Short delays can also affect sound localization and spatial perception. Delays of 20–50ms can be associated with medium delay effects such as doubling and chorus, and longer delays (50+ms) generate rhythmic effects such as slap and echo. The differences in latency would not only make the synchronicity of sound events difficult in a musical performance but may also compromise the integrity of audio signals, particularly in the transformations of the echo due to effects associated with phase shifting.

In order to maintain the integrity of audio signals and to avoid difficulties synchronizing musical events, it was decided that a new programme was needed that

would enable equal latency across all locations and that would stabilize that latency such that fluctuations observed in tests would be eliminated. Given (1) that musicians in tests were able to perform more effectively when the latency was equivalent to the musical tempo and (2) that it would be preferable to be able to perform music at any given tempo, a solution was needed that did not simply stabilize the latency: it would be necessary to determine its total value. If the latency of the system could be matched to a user-defined tempo, a predictable and musically meaningful time difference between locations could be established that would enable easier synchronicity with the conductor as well as with musicians in the other nodes and the performance of music in any tempo. Even though the audio signal would be heard later in the other locations, this delay would be predictable and therefore easier to manage musically.

Audio system implementation

Online Orchestra's audio system was implemented using Max 7 (Cycling '74 2014), a visual programming environment for multimedia and interactive systems integrating a data-flow system (Max) with audio (MSP) and video (Jitter) packages. Max was chosen for its user-friendly interface that allows for easy use and modification of programmes (Max patches) by technicians who do not have a programming background. Max also provides an intuitive way of interfacing with JACK router[5] and a quick approach to prototyping functioning graphical user interfaces. In addition, Max has a large variety of third-party externals that can be installed for further features and functionality. These extensions include implementations of onset detection algorithms, real-time composition

libraries and video frame sharing solutions that were necessary for developing Online Orchestra's telematic performance systems.

Two patches were designed for the audio system: a more complex patch for the server computer at Falmouth University and a simpler patch for the node computers in the other three locations. JackPilot[6] was used to route the audio signals from different JackTrip connections into Max. The node audio system patch consists of simple audio connections: from the analogue audio inputs capturing the local performance to JackTrip's output for broadcasting to other locations and from JackTrip's input receiving the audio from the other locations to the analogue outputs of the speakers. The server patch manages more complex connections – it routes the incoming audio signals from all nodes and sends a different mix to each location. The user operating the patch can make separate mixes for every node giving more control over each audio stream, including individual settings for number of channels and volume levels. The patch also manages the audio of the local conductor node and allows talkback between conductor and remote locations. In addition to complex audio routing, the server patch handles latency management: the user can set any tempo, as long as it is slower than the maximum audio latency between nodes, and the patch manages the latency across the system, ensuring a stable and consistent latency that remains securely tied to the desired musical tempo.

The server patch was built to have additional features extending the functionalities of the system. It has an in-built metronome that synchronizes to the selected tempo and can be used by a conductor to keep time and match the music to the latency. The patch also sends user datagram protocol (UDP) messages to all of the locations with relevant information like the tempo to synchronize the video and a chat

system for the technician operating the patches to communicate. In addition, the patch automates certain processes like audio settings and application management through shell scripts. It is also possible to save presets for each composition with important information such as tempo changes and beat subdivisions. Finally, a set of calibration tools were developed to make up for subsidiary system latencies added by hardware peripherals, software buffering and other factors that contributed to the system latency.

The audio system for Online Orchestra performed well when put into practice. It successfully and consistently locked the latency to the tempo of the music, enabling the pilot performance, an extract of which can be seen at www.onlineorchestra.com. The system did require careful consideration and calibration. Before being used, it needed testing for timing accuracy as well as audio quality. These calibration tests involved using various test signals in the system to evaluate the integrity of the audio signal in different conditions. Once the tests were conducted, the system was stable during musical performance and the overall audio quality remained very high, with only minimal glitching. Stabilizing the latency also proved to be intuitive for the musicians of Online Orchestra, who did not experience difficulties synchronizing musical rhythms, and felt as if they were playing together across different nodes (see Rofe et al. 2017a).

Audio and video integration

The principle of stabilizing the latency in the audio was also applied to the video stream: the same idea of synchronizing the latency to the tempo was applied to video frames. After careful consideration, it was decided that the best-performing video-streaming

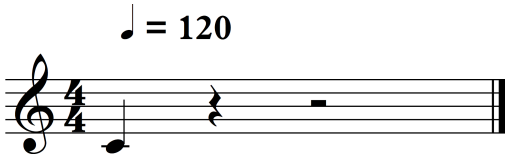
solution for Online Orchestra was a video chat service for Telemedicine called VSee (see Prior et al. 2017a; see also Chen 2008). VSee provided the most suitable features for the video-streaming system as required by Online Orchestra, including multiwindow peer-to-peer teleconferencing and flexible user settings. VSee also allows the user to choose between different video inputs, which makes it compatible with frame-sharing applications such as Syphon (see Butterworth and Marini 2010) or Spout (see Jarvis and Jarvis 2014). In order to synchronize the latency in the video signal, Max 7 was again used, with Jitter's video processing capabilities deployed to match the latency in the video signal to the desired musical tempo. Once the video latency matching was implemented, the video and audio systems were synchronized. The Jitter patch in each location receives the tempo information from the audio server patch through open sound control messages using UDP. When the latency of the video is matched, the Jitter patch recalibrates when the tempo is changed in the audio server patch.

Online Orchestra's prototype system therefore relied on a composite of multiple software platforms: JackTrip and VSee used to stream audio and video, respectively, and Max used to integrate these systems and provide overall control of latency (with separate, but connected, patches across locations used to match latency to tempo in the audio and video domains).[7] This solution proved highly effective as a prototype, though ultimately quite complex, particularly in the set-up and operational demands placed upon the user. Members of the Online Orchestra team are currently developing software that integrates all of this functionality into a single, easy-to-use format, in order that future users will be able to employ our system simply and effectively.

Latency in musical terms

The locking of latency to musical tempo enables a musical language to be used to describe latency relationships, which in turn proved to be a highly effective way of enabling musicians to work in a latency-rich environment. The Online Orchestra programme matches the latency to the desired musical tempo such that the length of one tactus beat is precisely the same as the latency between any two nodes. In the simple example in Figure 6, the programme stabilizes the latency between all nodes to 500ms (one crotchet at 120bpm lasts 500ms). As such, the latency can be conceptualized in terms of musical units rather than milliseconds – the latency is *one crotchet long*. In a slower piece, say crotchet = 100, the system is set to a latency of 600ms; conceptually, the latency is still *one crotchet long*, but that crotchet, and its associated latency, has a longer duration.

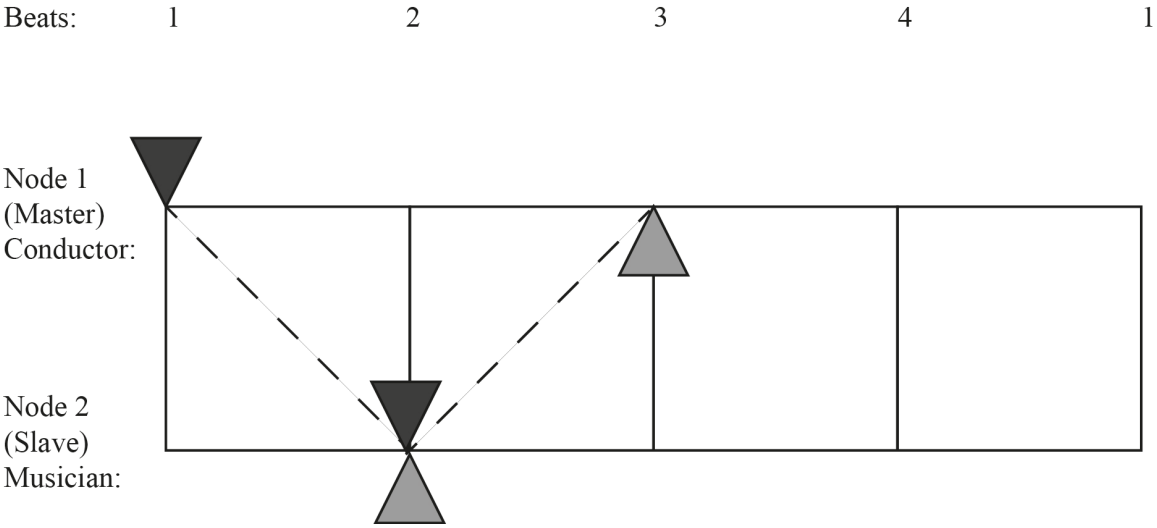
Figure 6: Example score.



In a two-node scenario, this brings about what can be termed a *master-slave relationship*, as visualized in Figure 7. In this case, node 1 contains a conductor (master), and node 2 has a musician (slave). Taking the simple score of Figure 6 again as an example, the system locks the latency to 500ms and outputs a metronome to the conductor in node 1 that enables him or her to beat exactly in time with the latency, which in turn is also the tempo of the music. Given that the system is constantly

monitoring and stabilizing the latency, the conductor is able to conduct constantly at 120bpm, knowing that the signal received by the musician remains stable. The conductor cues the musician to play, and the musician follows. Given the latency, it takes a crotchet beat for the signal of the conductor in node 1 to reach the musician in node 2. However, the musician is unaware of this delay, and so from their perspective is simply playing their note 'in time' with the conductor.

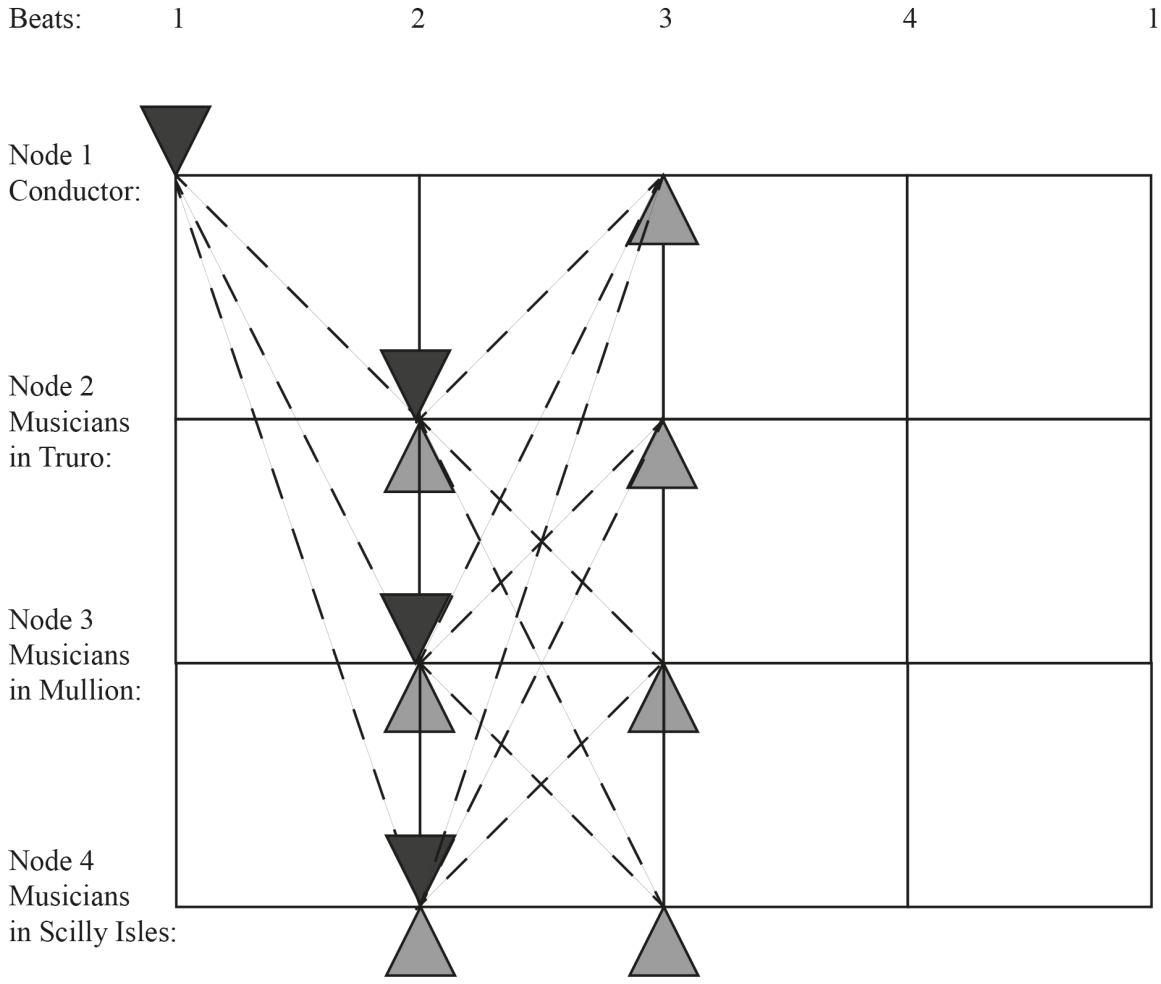
Figure 7: Latency relationships in a two-node scenario.



But from the conductor's perspective, the musician's note sounds back on beat three: it takes one beat for the visual signal to reach the musician and another beat for the note to return. So, in a two-node scenario, the musician (slave) need never knows there is latency – they are simply playing their part in time with the conductor. The conductor (master), though, must familiarize themselves with the fact that they hear the musician two beats later than written. Despite the apparent difficulty of this challenge, the conductor of Online Orchestra reports that it was surprisingly straightforward to become accustomed to this relationship (see Hargreaves 2017). Due to the latency control, the

return signal always and precisely falls on the conductor's third beat, making it easier to anticipate musically; without the latency control, it would fall indiscriminately according to the natural latency in the network, rather than being in time with the beat.

Figure 8: Latency relationships in a four-node scenario.



The situation is made more complex as more nodes are introduced. Figure 8 presents a schematic of the four-node scenario implemented in the Online Orchestra pilot performance, with the (master) conductor in node 1 and three (slave) musicians in nodes 2, 3 and 4. Given the latency control, the conductor's signal arrives simultaneously at all other nodes. Assuming the musicians in nodes 2, 3 and 4 all have the score of Figure 6, a

notional ‘bird’s eye view’ would confirm that all musicians do indeed play that downbeat simultaneously. This is confirmed by the fact that all musicians sound back to the conductor simultaneously on beat three. However, there is also a latency in the signal being sent between nodes 2, 3 and 4. From node 2’s perspective, for instance, they play their downbeat in time with the conductor, but hear nodes 3 and 4 a beat later (beat two). But from node 3’s perspective, it is them who are in time with the conductor, and nodes 2 and 4 who sound on beat two. And from node 4’s perspective, nodes 2 and 3 are on beat two. In other words, there are as many different ‘versions’ of the music as there are nodes – each node hears a slightly different realization of the score.

From the perspective of the musicians, this need not matter. Due to the stability of the latency control, other musicians are always exactly one beat late in this example. Without access to the full score, therefore, they need never know that latency is involved – from the perspective of node 2, for instance, whenever the score in Figure 6 is performed, nodes 3 and 4 have their notes sounding a beat later; this is simply the nature of the music, from their perspective.

So the challenge is less one of performance – each group of musicians simply plays their parts in time to the conductor – and more one of composition: the composer must write music that makes musical sense simultaneously in each different version. It is for this reason that Online Orchestra commissioned composers to write music specifically for this environment, with that rule system in mind (see Rofe and Geelhoed 2017); performing music that has not been composed for this environment would be deformed by the aforementioned latency relationships. In order to assist composers with this challenge, an additional piece of software was developed that takes a score and realizes it

as it would be heard in each location, taking into account the aforementioned latency relationships. This enables composers to hear how a piece would sound in its multiple manifestations.

Summary

Online Orchestra's solution to the challenge of latency is two-fold. First, a programme was developed that stabilizes latency, matching that latency to the musical tempo of the piece being performed. Second, music was commissioned designed explicitly for this latency-rich (or, rather, latency-controlled) environment, such that the latency is absorbed into the musical content. This solution is challenging to the composer and requires familiarization by the conductor. But the musicians are entirely unaffected – they simply play their part, as notated in a score, in time to the conductor. Whilst this approach may seem complex, that complexity enables a solution that (1) works over the types of bandwidths available in community contexts; (2) can work over any geographical distance; (3) does not require specialist equipment; and (4) enables a standard process of performance on the part of the musicians, in terms of conductor–score–performer relationships. It is a solution that works with the latency, rather than trying to eliminate it.

References

Bartlette, C., Headlam, D., Bocko, M. and Velikic, G. (2006), 'Effect of network latency on interactive musical performance', *Music Perception: An Interdisciplinary Journal*, 24:1, pp. 49–62.

Brossier, P. (2003), aubio: A library for audio labelling, http://aubio.org. Accessed 1 March 2016.

Butterworth, T. and Marini, A. (2010), Syphon, http://syphon.v002.info/. Accessed 1 March 2016.

Cáceres, J. and Renaud, A. (2008), 'Playing the network: The use of time delays as musical devices', *Proceedings of the International Computer Music Conference*, Belfast, Northern Ireland, 11-13 May, Michigan: University of Michigan. pp. 244–50.

Cáceres, J., Hamilton, R., Iyer, D., Chafe, C. and Wang, G. (2008), 'To the edge with China: Explorations in network performance', *ARTECH 2008: Proceedings of the 4th International Conference on Digital Arts*, Porto, Portugal, 7-8 November, Porto: Repositório digital da Universidade Aberta, pp. 61–66.

Chafe, C. and Cáceres, J. (2010), 'JackTrip: Under the hood of an engine for network audio', *Journal of New Music Research*, 39:3, pp. 183–87.

Chafe, C., Gurevich, M., Leslie, G. and Tyan, S. (2004), 'Effect of time delay on ensemble accuracy', *Proceedings of the International Symposium on Musical Acoustics* (ISMA2004), Nara, Japan, 31st March-3rd April.

Chen, M. (2008), VSee: World's Largest Video Telecommunication Platform, www.vsee.com. Accessed 1 March 2016.

Chown, T. (2016), End-To-End Performance Initiative, https://www.jisc.ac.uk/rd/projects/janet-end-to-end-performance-initiative. Accessed 1 December 2016.

Cycling '74 (2014), Max 7, https://cycling74.com/max7/. Accessed 1 March 2016.

Driessen, P., Darcie, T. and Pillay, B. (2011), 'The effects of network delay on tempo in musical performance', *Computer Music Journal*, 35:1, pp. 76–89.

Farner, S., Solvang, A., Sæbø, A. and Svensson, U. P. (2009), 'Ensemble hand-clapping experiments under the influence of delay and various acoustic environments', *Journal of the Audio Engineering Society*, 57:12, pp. 1028–41.

Gade, A. (1990), 'The influence of architectural design on the acoustics of concert halls', *Applied Acoustics*, 31:1–3, pp. 207–14.

Grame, A., Davis, P. and Petratoni, J. (2004), Jack OS X: A jack audio connection kit implementation for Mac OS X, http://www.jackosx.com. Accessed 1 March 2016.

Haas, H. ([1949] 1972), 'The influence of a single echo on the audibility of speech', *Journal of the Audio Engineering Society*, 20:2, pp. 145–59.

Hargreaves, J. J. (2017), 'Notes from the podium of an Online Orchestra', *Journal of Music, Technology and Education*, 10: 2-3, pp. 277-87.

Internet Society (2013), Workshop on Reducing Latency, http://www.internetsociety.org/latency2013. Accessed 14 October 2014.

Jarvis, L. and Jarvis, R. (2014), Spot: Realtime Video Sharing Framework for Windows, http://spout.zeal.co/. Accessed 1 March 2016.

LOLA (2015), LOLA: Low Latency Audio Visual Streaming System Installation & User's Manual, http://www.conts.it/art/lola-project/documents/Lola_Manual_1.4.2_rev_001.pdf. Accessed 1 November 2015.

Meier, F. (2013), 'Low-latency audio over IP on embedded systems', master's thesis, Hamburg: Technische Universitat.

Prior, D., Reuben, F., Biscoe, I. and Rofe, M. (2017a), 'Designing a system for Online Orchestra: Computer hardware and software', *Journal of Music, Technology and Education*, 10: 2-3, pp. 185-96.

Prior, D., Reeder, P., Rofe, M., Biscoe, I. and Murray, S. (2017b), 'Designing a system for Online Orchestra: Peripheral equipment', *Journal of Music, Technology and Education*, 10: 2-3, pp. 197-212.

Repp, B. and Su, Y. (2005), 'Sensorimotor synchronization: A review of the tapping literature', *Psychonomic Bulletin and Review*, 12:6, pp. 969–92.

Rofe, M. and Geelhoed, E. (2017), 'Composing for a latency-rich environment', *Journal of Music, Technology and Education*, 10: 2-3, pp. 231-56.

Rofe, M., Geelhoed, E. and Hodsdon, L. (2017a), 'Experiencing Online Orchestra: Communities, connections and music-making through telematic performance', *Journal of Music, Technology and Education*, 10: 2-3, pp. 257-76.

Rofe, M., Murray, S. and Parker, W. (2017b), 'Online Orchestra: Connecting remote communities through music', *Journal of Music, Technology and Education*, 10: 2-3, pp. 147-66.

Santana, I. (2014), 'Networked dance performance: A new temporality', *Liminalities: A Journal of Performance Studies*, 10:1, pp. 2–19.

Schuett, N. (2002), 'The effects of latency on ensemble performance', doctoral thesis, Stanford: Stanford University.

Traub, P. (2005), 'Sounding the net: Recent sonic works for the internet and computer networks', *Contemporary Music Review*, 24:6, pp. 459–81.

Willassen, S., Ødegaard, I., Bruun, T., Aas, D.-I., Hughes, B. T. and Hansen, S.-E. (2012), appear.in, https://appear.in. Accessed 15 February 2016.

Wright, M. (2005), 'Open sound control: An enabling technology for musical networking', *Organised Sound*, 10:3, pp. 193–200.

Notes

1. Jisc's *End-To-End Performance Initiative*, led by Tim Chown, is currently mapping different component latencies in an attempt to optimize network performance. Results are due for publication in July 2017. See https://www.jisc.ac.uk/rd/projects/janet-end-to-end-performance-initiative.

2. The onset algorithm we used for the audio latency measurement tool is part of the *aubio* set of tools developed by Paul Brossier; see Brossier (2003).

3. JackTrip, developed by the Centre for Computer Research in Music and Acoustics at Stanford University, was the primary audio engine used in the pilot performance; see Prior et al. (2017a), in the present special issue; see also https://ccrma.stanford.edu/groups/soundwire/software/jacktrip/.

4. Appear.in was used for this test: a WebRTC-based application for global communications; see Willassen et al. (2012).

5. This is the routing system used by JackTrip.

6. Jack OS X was used, which includes the JACK server, router and plugins integrated into the JackPilot application for Macintosh computers; see Grame et al. (2004).

7. See Prior et al. (2017a) and Prior et al. (2017b), both in this special issue, for more detail on the overall system design.