Scaling Up Live Internet Performance with The Global Net Orchestra

Roger B Dannenberg
Carnegie Mellon University

Tom Neuendorffer

Scaling Up Live Internet Performance with The Global Net Orchestra

Roger B. Dannenberg
Carnegie Mellon University
rbd@cs.cmu.edu

Tom Neuendorffer
Carnegie Speech Corp.
tneuendorffer@yahoo.com

ABSTRACT

Networked or "telematic" music performances take many forms, ranging from small laptop ensembles using local area networks to long-distance musical collaborations using audio and video links. Two important concerns for any networked performance are: (1) what is the role of communication in the music performance? In particular, what are the esthetic and pragmatic justifications for performing music at a distance, and (2) how are the effects of communication latency ameliorated or incorporated into the performance? A recent project, the Global Net Orchestra, is described. In addition to addressing these two concerns, the technical aspects of the project, which achieved a coordinated performance involving 68 computer musicians, each with their own connection to the network, are described.

1. INTRODUCTION

The Global Net Orchestra set out to connect many musicians performing live in a worldwide concert over the Internet. While Internet or "telematic" music performances have become almost routine for many performers, nearly all performances involve at most two or three locations. The first author's Federation of Laptop Orchestras tied 6 ensembles and approximately 50 musicians together with audio, video, and control links in 2012. The Global Net Orchestra builds on this experience and expanded the orchestra from 6 sites to over 60.

In the following sections, we consider first the motivation for the Global Net Orchestra and related work. Then, we discuss concerns about latency: how much is there and what does it imply for performances? Next, the implementation of the Global Net Orchestra is presented. Finally, we present some conclusions.

2. WHY NETWORKED MUSIC PERFORMANCE?

Music often involves constraints. When composers write for string quartet, use tone rows, or write for children, they take on a set of constraints and limitations on what is possible, but often, solving the implied problems brings structure and interest to the work. Networks have a significant impact on performance. Communication is usually hampered by the lack of proximity, limited visual cues, audio latency, feedback, small screens, and cumbersome technology. Why would anyone want these problems? As with many other musical forms, constraints serve to organize music, inspire new musical directions, and let the audience appreciate the overcoming of obstacles.

In addition, networks are a reflection of the new world that we live in. If we can speak of cyberspace as a place, why should it not have music? One fascinating attraction of network-based music is the simple question: If there were music in cyberspace, what would it sound like? Many musicians have developed music performances and compositions in order to find out.

Another rationale for network-based music is purely pragmatic. Travel costs time and money, but network transmission is very inexpensive. In some sense, broadcasting enabled the first network-based music. Broadcast media project the sights and sounds of performances to distant audiences. High-definition audio and video broadcasts of opera capitalize on the advantages of high-speed digital networks over older broadcast technologies. Network communication offers the possibility of more interactive performances, including music instruction. Many music teachers use Skype and other Internet applications to communicate with students, and there have been many high profile demonstrations of master classes and even rehearsals conducted using network communication.

The objective of the Global Net Orchestra is more philosophical than pragmatic. Music is a way of bringing people together. Music requires collaboration and sharing. Music making is often a social experience where one makes friends and enjoys their company. Every musician has had the positive experience of making music within a group. What would it be like to be part of an orchestra that spanned the entire globe, where a sense of "we are here" and "they are there" (so "they" must be different) was replaced by a sense of "we are all together" and "we unite across all boundaries" (so "we" are all one)? This is the main goal of the Global Net Orchestra: to answer the question by creating that experience for scores of players around the globe.

3. RELATED WORK

Music performances using network technology have a long history. Bishoff, Gold, and Horton (1978) describe music created through the live interaction of their network of microcomputers. Although their first network was about the size of a tabletop, they worked later at a distance and their work inspired many to consider the implications of networks for music performance [7], [19]. Artists explored applications of telecommunications in the 1980's using available technologies such as telephones and slow-scan video [11]. Sawchuck, et al. describe a number of early Internet music experiments and performances [16].

While researchers explored the limits of latency and bandwidth using a variety of technologies, commercial recording studios, especially those with large budgets, were quick to adopt digital networks to allow studio musicians to record tracks at a distance. For example, Robjohns describes remote recording supported by ISDN as an already well-established practice in 1999 [15].

Laptop orchestras [17] frequently use networking to coordinate players, although usually in the confines of a single stage. The first author's Federation of Laptop Orchestras [4] linked multiple laptop orchestras and acoustic instrumentalists for a live performance. Oliveros et al. [14] and Mills and Beilharz [13] describe a number of Internet music performances.

Networked applications based on constructing shared loops include TransJam [2] and Daisyphone [1], which allow people on the Internet to edit and perform a shared loop of music in quasi-real time, and JamSpace [8], which supports many users on a local area network (LAN). Weinberg, et al.'s *Beatbug Network* was not spatially distributed but explored synchronized networked music interaction [18]. Miletto, et al. describe a system for more asynchronous networked interaction and compare a number of networked music environments [12].

4. THE LATENCY QUESTION

Latency is inherent in communication. While we often imagine a conventional acoustic performance to be free of any latency issues, the speed of sound is a limiting factor for coordination and synchronization in larger ensembles such as orchestras. As it takes sound roughly 1 ms to travel 1 foot (about 3 ms/m), musicians commonly deal with delays of 50 ms or more. The speed of light in fiber is about 200,000 km/s, so a round-trip across the United States is about 50 ms. Routers, repeaters, packet-switching, and the actual length of cable add to this figure, so for example, an actual round-trip from Carnegie Mellon University (eastern United States) to Stanford University (western United States), measured using the Unix *ping* command is 86 ms.

The Global Network Orchestra extends worldwide, so latencies are even greater. The website startping.com (no longer in operation) conveniently posts round trip times from many locations. Some interesting minimum round-trip times (in ms) from Pittsburgh, measured in February, 2014, include: 8 (Detroit), 81 (London), 142 (Helsinki), 176 (Santiago), 259 (Melbourne), 274 (Malaysia), 280 (New Delhi), 286 (Hanoi), and 530 (Hangzhou, China, with a 30% packet loss rate). To these times, one might need to add up to 100ms for local routing and cable or DSL modem delays to get to a home. For example, Verizon's lowest cost DSL service plan in Pittsburgh has a 90 ms round trip time to Carnegie Mellon University, also in Pittsburgh.

These times are between two points, but it may not be practical to have a complete peer-to-peer organization where every performer connects to every other performer. The simplest configuration, and the one adopted by the Global Net Orchestra, is a star or hub-and-spoke arrangement (see Figure 1). Every performer sends to a server, and the server distributes information back to every performer. In that configuration, the best-case latency between two points is the mean of the sum of round trip times from those points to the server. For example, the latency from London to Detroit through a server at Carnegie Mellon University should be half the round-trip times from London to Pittsburgh and from Pittsburgh to Detroit, or (81 + 8) / 2 = 44.5 ms.

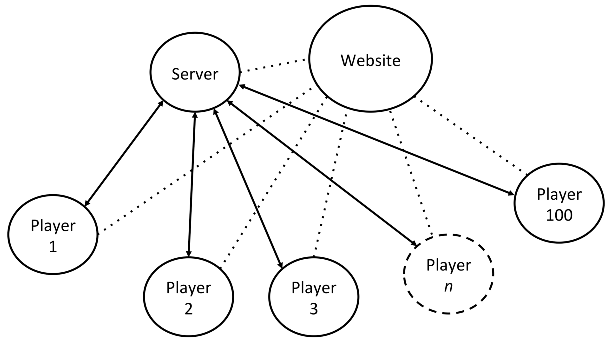

Figure 1. Network configuration of the Global Net Orchestra. Players locate the Server by contacting the GlobalNetOrchestra.org website. During the performance, players communicate only with the Server.

Given these potentially long latencies, one can expect a significant impact on music making. Good analogies are a crowd singing or chanting in a large hall or a marching band spread across an entire football field. There, the speed-of-sound issues impose delays in the same range of 200 to 400 ms.

Our ideas borrow from the experience of other projects that have found ways of dealing with latency. For the Global Net Orchestra, we implemented and used 3 different techniques.

4.1 Perform with Latency

The first approach is to simply do the best one can. For example, if even conventional music is played slowly enough, the asynchrony of different parts and voices can be tolerable. In the Global Net Orchestra, we can "conduct" performances with scrolling scores that at least avoid the tendency to slow down when one hears other parts with significant delays.

4.2 Emphasize Texture over Rhythm

A second approach is to emphasize music based on texture and gestures, where rhythm and synchronization are not so important. In the Global Net Orchestra, following experience with the Federation of Laptop Orchestras, we use a "guided improvisation" approach inspired by Anthony Braxton [10]. Players are given graphical images depicting "Musical Languages" that specify styles or sonic textures. A conductor selects these images for different subgroups of the orchestra and the images are displayed on the players' laptop screen. In addition, the conductor can give cues for other attributes such as intensity, time points (for certain textures), and pitch (high, medium, low). Some images are shown in Figure 2.

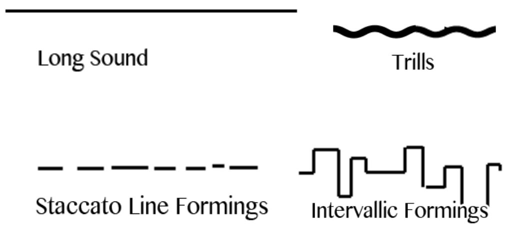

Figure 2. Selected images describing "Musical Languages" that convey playing styles or textures to players.

4.3 Cyclical Music with Accurate, One-Period Delays

A third possibility is to delay everything so that performance information can be delivered to all sites and played synchronously. That allows every performer to hear the same timings and synchronization as every other performer, but the drawback is that every performer experiences at least the worst-case latency applied to everything. Delaying everything by, say, 1 second would make performing *very* difficult, but an interesting possibility is to delay by an entire cycle in a cyclical structure [6]. For example, if everyone plays 12-bar blues, and everyone's performance is delayed exactly 12 bars, the performances will line up perfectly on beats, measures, and chord structures, even though the performed notes will not be heard until 12 measures after they are played.

Inspired by drum circles, we implemented this idea with shorter cycles of 4-measure riffs. To be precise, the delay was 16 beats at 100 beats per minute, or 9.6 s. Beats were displayed visually, scrolling scores were used to tell players what to play when not improvising, and a drummer played a beat (also delayed precisely 9.6 s) to help with synchronization. Software allowed performers to request to play a 16-beat solo, and when cued to do so, the soloist's sound was boosted in volume to feature the solo above the background riffs.

4.4 Latency Compensation with Time-Advanced Scores

A fourth possibility (not implemented yet) is to estimate the network latency from the performer to the server and advance the scrolling score by that amount of time. If every performer follows the locally time-advanced scores, then all performance data will reach the server synchronously and the merged stream of performance data will be synchronized. There will be additional varying delays sending this data back to the performers, so the "full performance" will occur at slightly different times based on the distance from the server, but each player will hear a synchronized performance.

We believe in this scenario it will be best if the local performance is delayed by the round trip time; otherwise, the performer's part will be heard well ahead of the rest of the orchestra. In the present implementation, data from each performer (note-on, note-off, velocity, etc.) is of course available locally, but it also appears in the data stream from the server. Either the immediate local data or the server stream data can be selected to play the performer's "voice." In this scenario, simply selecting the server stream as the control source for the performer's voice will accomplish the desired delay.

5. NETWORK IMPLEMENTATION

The Global Net Orchestra transmits control information rather than audio between players. Although audio transmission is technically possible, our experience with audio in the earlier Federation of Laptop Orchestras project indicated that scaling up audio from 6 sites to 100 or more would be a great challenge, particularly because there simply are not hundreds of musicians with access to high speed, low latency Internet connections, and even when sites are available, networks are usually not configured to allow sustained high bandwidth, so considerable technical support and cooperation are required to establish reliable communication.

A large-scale peer-to-peer audio network is possible as shown in Figure 3. Each "interior" node has 4 (or more generally, *n*) input/output ports. Each port outputs the sum of the inputs to the other ports. There are also "edge" nodes that represent either a local audio device or network connection that simply transmits/receives audio to/from a remote computer. It can be shown that, in any acyclic connected graph consisting of these nodes, each audio output will consist of the sum of all other audio inputs to the graph. There are no "hot spots," and the maximum path length in terms of edges grows only logarithmically with the size of the network. This is in some sense an ideal configuration for very large networks, and we hope to pursue this idea in the future.

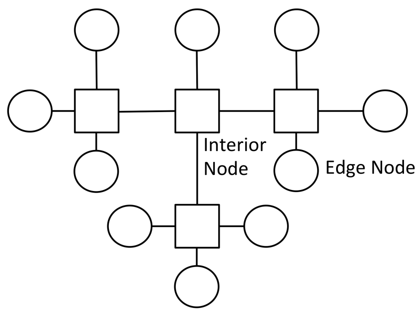

Figure 3. Peer-to-peer audio distribution network with bounded bandwidth requirements at any given node.

Our alternative (Figure 1) has a "hot spot" in the form of the server, but we cope with the potential bandwidth problem by sending only the state of the musicians' performances, described as the current pitch and velocity (loudness). Assuming 100 players, this gives us 200 bytes of data per musician. Transmitting 20 updates per second, we get 4000 bytes per second, or 32 kbps *download* bandwidth from our server to each player, and 3.2 mbps outgoing bandwidth from the server. The *upload* bandwidth to the server is much less because only the local state (pitch, velocity, and some network monitoring information) needs to be transmitted to the server from each musician.

A complex part of the implementation is the network communication system. Unlike audio, where lost packets are annoying but ephemeral, a lost packet with a note-off command or a “start the performance now” command could be disastrous. We use a combination of “stateless” message protocols, redundant transmission (forward error correction), and reliable protocols (with retransmission).

5.1 Stateful and Stateless Protocols

“Stateful” protocols rely on memory of past transmissions at the sender and receiver. Often this reduces duplication and increases efficiency, but duplicated or lost messages can have a lasting effect on the receiver. For example, it might be efficient for the server to send only changes (note-on and note-off messages) to the players, but then it becomes critical to ensure that all messages are delivered reliably. This, in turn, can impose extra latency for retransmission of lost packets, and this is undesirable for music performances.

In contrast, our primary protocol is a “stateless” one. In our case, we continuously retransmit the *entire* state of the performance, that is, every note that is sounding. If a message is lost, or even if the client restarts, the client will quickly obtain a correct and consistent state as soon as it receives the next message.

5.2 Redundant Transmission

In our system, we transmit the state every 50 ms. If a packet is lost, rather than asking for a retransmission, the client simply waits for the next packet and should recover in 50 ms, a much faster recovery time than retransmission could provide. A further advantage of this approach (which is built upon the UDP protocol) is that UDP packets are not subject to flow control. If there is contention between our 50 ms packets and, say, a file transfer, both will lose some packets, but the file transfer (based on TCP) will back off to provide more bandwidth for our application. (TCP interprets lost packets as a sign of network congestion and reacts by reducing the transmission rate, but UDP has no flow control.)

A drawback of this approach is that musical events have to wait an average of half the transmission period of 50 ms before the next packet is transmitted. This would be an unacceptable delay for most music, but given the Internet delays ranging into hundreds of ms, an added delay and jitter of 25 ms seems acceptable.

Originally, we intended to apply this stateless idea to all information exchanged with clients, but there are some cases where this is difficult. A good example is when one wants to send a chat message for display to all the players. How much data should be sent? Do we send the entire chat history every second or two? Can we get by with sending a few copies with sequence numbers to detect duplication? Already, this is getting complicated, so we added additional reliable connections.

5.3 Reliable Protocols

In addition to the low-latency Global Net Orchestra protocols based on UDP, we employ the ZeroMQ library [9], which is based on TCP and provides reliable delivery of messages. ZeroMQ messages are used for (1) chat messages so that users can send short text messages to each other, (2) voice messages from the conductor to all players (these are short announcements transmitted in their entirety before playback so as not to require real-time streaming or high bandwidth), and (3) sending exact start times for performances, some of which are directed by synchronized scrolling scores. ZeroMQ implements a “publish-subscribe” system that makes it particularly easy for the server to broadcast data reliably to all clients.

5.4 Clock Synchronization

Some of the orchestra pieces are “conducted” by a scrolling score (described below), so it is important that scores all be synchronized. This is accomplished in two steps. First, we synchronize clocks with a clock synchronization protocol. Second, we announce the exact starting time of each piece well in advance to eliminate any problems with network latency.

Clock synchronization is accomplished by having the server occasionally reply immediately to a client packet with the current (server) time. Clients remember when each message is sent so that when a reply is received, the client can compute the round-trip time and estimate the offset between the client and server clocks. To prevent large errors from dropped or delayed packets, every client gets a reply every 5 s, and every 50 s, the best round trip time is used to adjust the local clock.

5.5 Network Address Translation and OSC

Network Address Translation, or NAT, is a technique where network packet addresses are mapped from one address space to another as they pass through routers. NAT is commonly used by network service providers because it allows all devices in a home to access the Internet by sharing a single IP address. Normally, this means that UDP packets – commonly used for Open Sound Control [20] – cannot be delivered to a home network because there is no way to address the desired host computer.

Global Net Orchestra clients solve this problem by attaching a reply address to UDP packets. The reply address is translated by NAT, providing the server with an IP address that can be used to return UDP packets to the sender. Typically, Open Sound Control libraries are not set up for bi-directional communication and do not set reply ports or allow for the sending of replies. We created a new implementation of OSC within Serpent to support bi-directional communication where a client is using NAT and the server has a known IP address.

Not all routers and NAT protocols are the same. We found a few cases where our reply packets (server to performer) were blocked. These problems could only be solved by switching to another location and another network. Nevertheless, we were pleased to find a simple way that almost always gets OSC packets through NAT.

Our implementation is freely available within the Serpent project at `sourceforge.net/projects/serpent`.

6. USER INTERFACE

Performers in the Global Net Orchestra use an application written in Serpent [3], which runs on Mac OS X, Windows, and Linux operating systems. The application includes a synthesizer capable of playing 100 audio samples simultaneously when a large orchestra is playing. Each orchestra member creates and uploads samples to a website. Samples are merged and integrated with the application, which players can then download and use. The samples are compressed for downloads, but occupy almost 1 GB of memory at run time.

Figure 4 shows what performers see. At the top is a status display, a graphical keyboard the user can click on to test audio, and audio mix controls. Below is a scrolling "piano roll" display to conduct the orchestra. At the top of this window is an area for receiving and composing chat messages. The area below the chat interface is also used to display instructions and for other modes of interaction, including the directed improvisation using images from Figure 2. There is an option to use a MIDI controller for input, but the user can also simply type on the laptop keyboard. Of course there is also audio output, which consists of the performer's sounds, synthesized immediately, the other orchestra sounds, synthesized according to state messages from the server, and voice messages, delivered from the server.

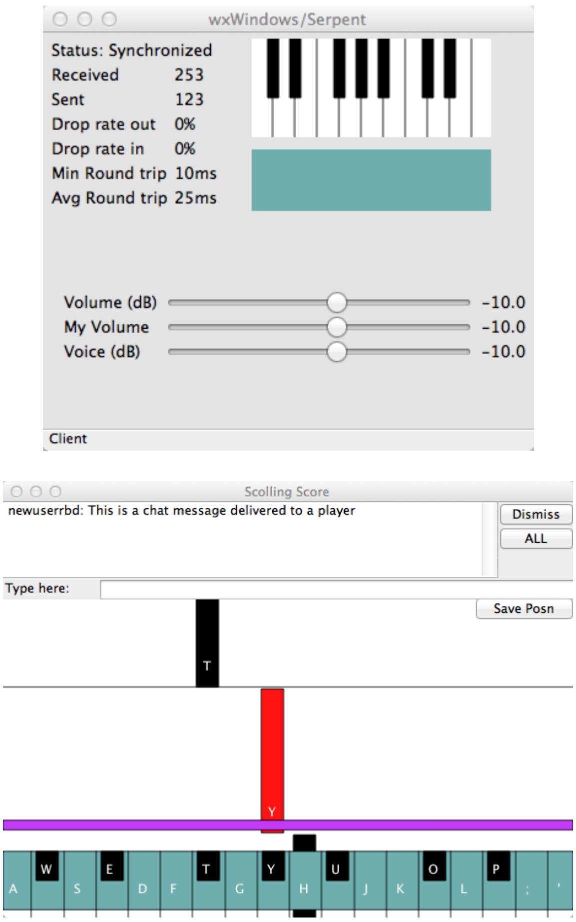

Figure 4. Performer interface.

7. THE PERFORMANCE

The Global Net Orchestra performed on March 1, 2014. The performance included a live audience at the Ammerman Center 14th Biennial Arts and Technology Symposium at Connecticut College and another audience at Carnegie Mellon University. Video from Carnegie Mellon was streamed over the Internet. There were 68 performers connected to our server, and their locations are plotted in Figure 5.[1] The concert was successful, with only one known case of someone being unable to connect or perform.

Figure 5. Locations of performers

The scrolling scores were very effective at keeping the group together. Of course, the scores tend to make performances more mechanical and eliminate the need to listen carefully, but on the other hand, it would be hard to keep a steady tempo or even stay together without some form of conducting. The orchestra played some slow renditions of the chorale from Bach's Cantata BWV 42, which begins "Grant us peace graciously…" and the traditional canon "Dona Nobis Pacem" (Grant Us Peace).

In addition the orchestra played an extended improvisation and a rhythmic piece using one-cycle delays. In both of these pieces, the inability to switch to drum sounds or to sounds with short onsets was a problem. Future versions of the software should give performers more control over articulation.

Throughout earlier rehearsals and the performance, the use of the talk-back feature was extremely valuable. This feature allows the author, serving as "semi-conductor," to record short announcements that are delivered and played through the audio output of every performer's computer. This helps to get everyone's attention much better than chat (text) messages, which can easily be missed. The talk-back messages are distributed via the server, so in principle anyone can send a voice message to the orchestra. However, this is not allowed because of the limited bandwidth.

One performer, blogging about her experience, wrote:

> …for one moment, we were performing together, as faceless to each other as we were to the audience. The Global Net Orchestra was about what we were doing together. It might not have been perfect or even pretty. But it felt like the beginning of something bigger.

[1] Once we generated the map, we were surprised to discover no performers in South America or Africa. Performers were recruited using mailing lists and personal email that should have reached every continent. We plan to recruit harder in these regions next time.

7.1 Network Performance

Since we measure round-trip latency for clock synchronization, we log Global Net Orchestra network statistics every 10 seconds. After our March 1 performance we were able to obtain some statistics. The average round-trip time was 101 ms, with a standard deviation of 116 ms. We measured the minimum round trip time for each connection, and the average of these times is 57 ms with a standard deviation of 76 ms. The difference from minimum to average indicates that the typical packet often encounters significant delays, presumably because of buffers in the network. The longest *minimum* round trip time was 1.3 s, indicating significant network delays for some performers. The average packet loss rate is 0.53% from performer to server, and 0.76% from server to performer. The most common loss rate is 0%, and the second most common is 100%, indicating a total connection loss. In any given 10 s monitoring period, the probability that a player would be disconnected was 0.32%. This is a small number, and most performers were not affected, but there were 12.5 total minutes of disconnected time. Fortunately, we were careful to automate the (re)connection process so that performers could recover from network losses rather quickly.

8. FUTURE WORK

We anticipate future performances of the Global Net Orchestra. The orchestra could benefit from sound design, replacing the uncoordinated personal samples in current use and also adding a way to vary articulation. Long onsets are very appropriate for slow, synchronized pieces, but short percussive sounds are needed for the "drum circle" mode and some improvisation "languages." Scaling the orchestra to larger sizes is an interest; it should be easy to support 250 players with simple optimizations, and more if everyone has fast processors. Beyond that, it probably makes the most sense to synthesize audio at the server and stream compressed audio to each performer. We are also considering extending beyond the boundaries of the earth. It seems possible that humans could return the moon within 5 years. The speed-of-light delay is about 1.25 s, which is easily fast enough for the "drum circle" mode, improvisations, or time-advanced scores. Finally, we would like to see a global drum circle operating continuously with automated direction and the ability for people to connect at any time, day or night (or both).

9. CONCLUSIONS

The Global Net Orchestra performed on March 1, 2014, with over 60 musicians performing live across the globe. While the performers have only limited control (typically keys on their laptop), limited sounds (a small set of samples with limited pitch range), and limited interactivity due to network latency and bandwidth issues, the orchestra offered a unique musical experience. Performers (and the audience too) felt a sense of connection to their musical partners around the world. Here, the speed-of-light delays may have actually enhanced the experience by suggesting just how much physical separation there is between players who nevertheless can perform and interact in real time as one ensemble. We hope the orchestra will contribute in some small way to a feeling of connection between people of all nations.

Acknowledgments
The authors would like to thank the organizers of the Ammerman Center 14th Biennial Arts and Technology Symposium at Connecticut College for their encouragement and for providing a special time and place for the premiere. This work would not be possible without the support of the School of Computer Science at Carnegie Mellon University. Thanks also to students at Carnegie Mellon and to Pauline Oliveros and her students at Rensselaer Polytechnic Institute who helped to test the system and put up with its initial shortcomings. This paper extends an earlier one with new findings, including our actual concert experience, network performance measurements, and details about scaling up the implementation to even more players.

10. REFERENCES

[1] Bryan-Kinns, N. "Daisyphone: The Design and Impact of a Novel Environment for Remote Group Music Improvisation." In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques* (DIS '04). New York: ACM, pp. 135-144.

[2] Burk, P. "Jammin' on the Web – A New Client/Server Architecture for Multi-User Musical Performances." In *Proceedings of the 2000 International Computer Music Conference.* Berlin, pp. 117-120, 2000.

[3] Dannenberg, R. "A Language for Interactive Audio Applications." In *Proceedings of the 2002 International Computer Music Conference*. Göteborg, pp. 509-515, 2002.

[4] Dannenberg, R. 2012. "Laptop Orchestra Communication Using Publish-Subscribe and Peer-To-Peer Strategies." In *Proceedings of the 1st Symposium on Laptop Ensembles and Orchestras* (Available online https://ccrma.stanford.edu/~ruviaro/texts/SLEO_2012_Proceedings.pdf, accessed Feb. 2014), pp. 88-93.

[5] Föllmer, G. "Electronic, Aesthetic, and Social Factors in Net Music. *Organized Sound* 10(3), pp. 185-191, 2005.

[6] Goto, M. and R. Neyama. "Open RemoteGIG: An Open-to-the-Public Distributed Session System Overcoming Network Latency," *IPSJ (Information Processing Society of Japan) Journal*, Vol. 43, No.2, pp.299-309 (in Japanese), 2002.

[7] Gresham-Lancaster, S. "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Music," *Leonardo Music Journal* 8, pp. 39-44, 1998.

[8] Gurevich, M. "JamSpace: Designing a Collaborative Networked Music Space for Novices." In *Proceedings of the 2006 International Conference on New*

Interfaces for Musical Expression (NIME06), Paris, pp. 118-123, 2006.

[9] Hintjens, P. "ØMQ – The Guide." http://zguide.zeromq.org/page:all, accessed Feb. 2014.

[10] Lock, G. *Forces in Motion: The Music and Thoughts of Anthony Braxton*. Da Capo Press, 1988.

[11] Matuck, A. "Telecommunications Art and Play: Intercities SáoPaulo/Pittsburgh." *LEONARDO*, Vol. 24, No. 2, pp. 203-206. (Available online http://studioforcreativeinquiry.org/publications/telecommunications-art-and-play-intercities-sao-paulopittsburgh, accessed Feb. 2014), 1991.

[12] Miletto, E., M. Pimenta, F. Bouchet, J.-P. Sansonnet, and D. Keller "Principles for Music Creation by Novices in Networked Music Environments." *Journal of New Music Research* 40(3), pp. 205-216, 2011.

[13] Mills, R. and Bielharz, K. "Listening Through the Firewall: Semiotics of Sound in Networked Improvisation." *Organized Sound 17*, pp. 16-27, 2012.

[14] Oliveros, P., S. Weaver, M. Dresser, J. Pitcher, J. Braasch, and C. Chafe. "Telematic Music: Six Perspectives." In *Online Supplement to Leonardo Music Journal*, 19. (Available online https://ccrma.stanford.edu/groups/soundwire/publications/papers/chafeLMJ19-2009.pdf, accessed Feb. 2014), 2009.

[15] Robjohns, H. "All About ISDN." *Sound On Sound*. (Available online http://www.soundonsound.com/sos/apr99/articles/isdn.htm, accessed Feb. 2014), April 1999.

[16] Sawchuk, A., E. Chew, R. Zimmermann, C. Papadopoulos, and C. Kyriakakis. "From remote media immersion to Distributed Immersive Performance." *ETP '03: Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*. ACM Press New York, NY, USA. pp. 110–120, 2003.

[17] Smallwood, S., D. Trueman, P. Cook, and G. Wang. "Composing for Laptop Orchestra." *Computer Music Journal* 32(1). pp. 9-25. 2008.

[18] Weinberg, G., R. Aimi, K. Jennings. "Beatbug Network – A Rhythmic System for Interdependent Group Collaboration." In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02),* Dublin, pp. 186-191, 2002.

[19] Weinberg, G. "Interconnected Music Networks: Toward a Theoretical Framework." *Computer Music Journal* 29(2), pp. 23-39, 2005.

[20] Wright, M., A. Freed and A. Momeni. "Open Sound Control: State of the Art." In *International Conference on New Interfaces for Musical Expression*, Montreal, pp. 153-159, 2003.