

peerSynth: A P2P Multi-User Software Synthesizer with new techniques for integrating latency in real time collaboration

Jörg Stelkens
büro </stelkens>
Adlzreiterstr. 14, D-80337 München, Germany
info@stelkens.de, www.stelkens.de

Abstract

In recent years, software instruments have enabled the exchange of musical information over a network and thus, collective music making. Time-dependent delay (latency) occurring between those creating music over asynchronous networks like the Internet presents a pertinent yet, up till now, scarcely examined problem. The author will present a simple process which integrates network latency into the individual musicians' collective playing. This process is part of a P2P multi-user software instrument developed by the author called peerSynth. This real time synthesis program runs on standard PC's, is easily distributed over the Internet and allows a decentralized P2P network to be built up over the Internet. With the help of a specially developed user-interface, the software enables multiple users to collectively make music in both real time and offline sessions independent of time and space. Through these processes, a "boundaryless" music can occur.

1 Introduction

The development of networked computer systems through the help of the Internet has from early on inspired composers and instrument builders to use such technologies for collaborative music making. Accordingly, a number of musicians and composers have developed instruments or written compositions that build upon the exchange of data and establish network structures making collective music performance or composition over spatial distances possible. An early example of such an Internet project is the 1997 public interactive algorithmic composition "Variation for WWW," by Seinoshin Yamagishi and Kohji Setoh from Japan.[1] In this project, users could collectively influence a composition through simple input over an HTML form, the results of which could be heard through an audio-stream. A basic introduction to the histories and methods of numerous musical network projects can be found in Gil Weinberg [2] and Golo Föllmer [3].

1.1 The latency problem

Destructive time lags or so-called latency, occurs during collective music making over asynchronous networks in such real time projects as "quintet.net" by Georg Hadju [4], the downloadable software on the "Eternal Network Music Site" from Chris Brown [5] or the Waag Society's "Keyworx [6]." Other projects, such as "FMOL" from Sergi Jorda [7], "Audioserve" from Matthew John Yee-King [8] or the recent closed "Rocket Network" by Rocket Network Inc.[9], sidestep this problem since they don't establish a real time connection, but only exchange data as files after they have been created.

Further latency issues in real time systems arise in the use of networked live sequencers where relative time information (as opposed to absolute) is carried over the network. A local sequencer takes over the time-correct execution of musical events. Such processes, described as "forward synchronous" systems [10], can be found in projects such as "Webdrum" by

Philip Burk [11] and the still not released "Pazellian" by Donald Pazel [12]. With these systems, however, a further problem arises when multiple users play together locally. Because of the network jitter that occurs through the asynchronous connection, it is essentially not possible for the individual sequencer events to run time synchronously, even with measurement and approximation of the round trip time (RTT). If the users are in the same physical space, the time delays between events will always be audible.

The real time project "Net Resonator," from Kojo Ito [13] et al, takes yet another direction. Pauses, which occur from latency during playing are covered up by post-processed echoes on each of the played tones. The users have to learn how to handle these echoes in their playing, adapting tempo and interaction to the resulting sequence of sound and unconsciously taking into account the latency properties of the medium.

1.2 Latency in collaborative music making

Research and tests on the actual effects of unwanted latency with individual musicians are rare. Qualitative contributions to the subject have been developed by the SoundWIRE research group at CCRMA through its two goals: (1) to realize professional, multi-channel audio streaming over high speed networks and (2), to develop intuitive methods for the measurement of delay over bi-directional network connections. An early study from Schuett [14], for example, has established a so-called Ensemble Performance Threshold (EPT) - the point at which a collaborative musical real time performance breaks down. This research, however, examines only a point to point connection in a two person performance situation. Two performers clap a simple rhythmic figure whereby a delay is either artificially (through electronics) or acoustically (through the change of distance in the acoustic space) generated between the participants. An important result of this work is that the EPT can be demonstrated through a clear

decrease in the speed of the musician's intuitively chosen performance tempo - accordingly, an EPT of approximately 20-30 msec was established. This corresponds to a distance between the performers of approximately 10m (with the speed of sound equaling 344m/sec).

1.3 Latency acceptance and integration

The hope for lower Internet latencies that lie below the EPT will in the near future, if at all, be fulfilled only with Internet connections based within large institutions. For the independent computer musician, access to high-speed networks like ATM or Internet2, as used, for example, in the Sound Exchange project from Mara Helmuth [15], remains, in general, inaccessible. Considering the observations of the aforementioned approaches to musical real time collaboration, the author therefore considers it of little use to ignore or avoid latency problems. Rather, through the author's synthesizer concept, the existence of latency is taken into account as a given. This acceptance of latency as a property of the medium is reflected, for example, in the words of the network and sound installation artist Atau Tanaka, who speaks of a "new musical language" and a particular Internet "acoustics:"

"Transmission delays will be considered a hindrance as long as we try to superpose old musical forms onto the network. Rather, a new musical language can be conceived, respecting notable qualities of the medium, such as packetized transmission and geography independent topology. These features can be said to define the "acoustic" of the network, a sonic space that challenges existing musical notions of event, authorship and time." [16]

Rasch [17] proved in his early experimentation that musicians who play instruments with relatively long attack times (e.g., string instruments) are capable of adapting the physical excitation of the instrument to stay in time during a performance. Similarly, this human musical ability can be utilized to work with the issue of latency by giving the user the ability to gauge the constantly shifting state of lag (caused by network jitter) between him/herself and the other players.

2 peerSynth

The author's software peerSynth's basic idea is to make a gauging of latency possible by changing the synthesized sound of a multi-user software instrument, depending on the current latency situation occurring in connections to the other users.

2.1 User Representations

peerSynth consists of a single program that differs from that of standard Client-Server approaches in that it establishes a real time P2P network through TCP/IP. For this purpose, peerSynth runs as an instance on the computers of the players. High resolution musical parameters (not audio data) are sent and received in real time by the instances. This saves bandwidth therefore making peerSynth ideal for modem connections. If exchange of audio data for the sampling unit of the synthesis kernel is necessary, this can occur serially through specially established TCP/IP connections running in the background without interrupting the real time data exchange. The same routines can also be used for the automatic exchange of session recordings (mp3).

Each of the peerSynth network players is represented in each local instance as a user with his/her own interface and sound synthesis unit(module). This means, for example, that if three users are connected via peerSynth then 3 user interfaces and 3 sound synthesis units are produced by each running instance through the help of object-oriented programming, here referred to as "user representations."

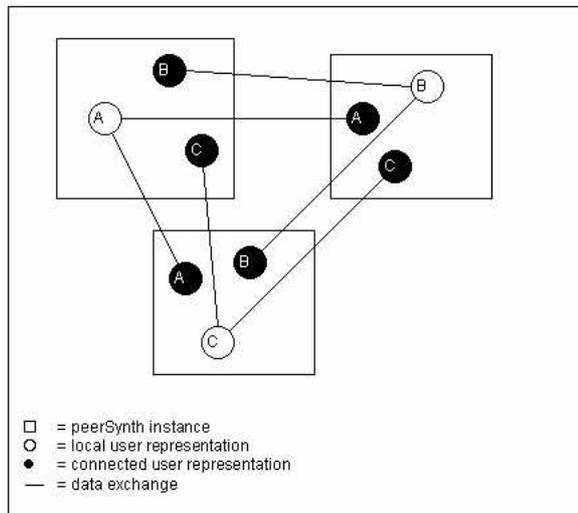


Figure 1: peerSynth user representations in a built-up P2P network

This approach differs from other existing multi-user-instruments that, as a rule are based on the linking together of individual musical user parameters on a server with a simple copy on each connected client.

2.2 Latency as musical parameter

By introducing the user representation, it now becomes possible to process each individual, user-produced sound within each peerSynth instance differently.

Through the help of a constantly updated, automated measurement of the RTT (high resolution ping), the individual latency of one's own peerSynth instance to the connected instances is determined. These diverging values can then be interpreted as musical parameters and influence the sound of each representation.

This process can be interpreted as an individual "perspective" in relationship to the other users, similarly as, for example, in a 3-D multi-player action game. In this case, the game does not look the same for each player - the individual user sees only the chosen perspective, including the representations of other users. With peerSynth, the user cannot actively alter his/her "perspective" of the other users but is instead continually introduced to new "perspectives" resulting from the ever-changing properties of Internet latency.

In this way, the property of the medium is represented directly in the sound of the instrument, enabling the players to adapt their performance accordingly.

2.3 Time to Space

The modulation of sound by latency can take place in different ways or be executed through a user-influenced, switchable synthesis modulation source. In practice, a technique is generated that works particularly well: the sound of a fixed synthesis unit is modified through further signal processing in such a way that a sound with relatively high latency values is heard as “further away” and with low values as “closer by.” This corresponds to the experience of acoustic musicians who play spatially separated in a large room or outside (“alpenhorn effect”). Sound propagation occurs with a corresponding delay and each musician perceives the music that is being played together differently.

peerSynth provides this type of time to space mapping in which the synthesized sound through its increasing latency appears softer, dampened and reverbed. peerSynth’s synthesis unit itself consists of a controllable granular synthesis unit with a subsequent analog delay simulation (“scratch delay”).

2.4 „Ball in a Bowl“ X/Y-Controller

A key element of peerSynth’s user interface is the “ball in a bowl” X/Y controller contained within the author’s real time granular synthesis software “crusherX-Live!,” presented in 2001 [18]. In this software, the point and field of control are assigned adjustable physical and geometric qualities, for example, mass, friction, curvature, etc. These qualities are then simulated in the software in real time directly onto the screen. The point of control can be clicked on and dragged in a particular direction by holding the mouse button down. When the mouse button is released, direction and speed of this movement is measured and simulated according to its corresponding physical qualities.

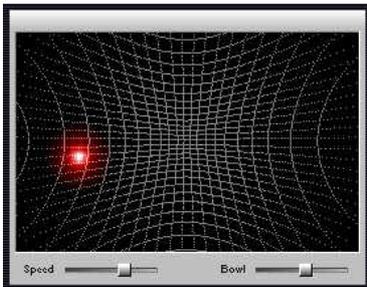


Figure 2: “Ball in a Bowl“ X/Y-Controller

peerSynth provides expanded parameter control of the musical representations (generator, scratch delay, volume and pan), thus resulting in the following advantage: the mouse sequence and consequently, controller data are “smooth,” meaning they contain few transients. When not in use, the controller automatically centers itself. Complex motion and controller data patterns occur which are not random or arbitrary but rather through physical simulation, are intuitively predictable based on the user’s experience of the software.

2.5 Interfaces and programming

peerSynth itself is a standard MDI-Windows application. The MDI field displays the interfaces of the connected user-representations. Three “ball in a bowl” X/Y controllers, which control one’s own synthesis instance, are displayed in each of

the representations (cf. Figure 3). The right slider shows the current “distance” of the “time to space” modulation and while being continually updated through the latency measurement, can also be manually changed. The user representation then sounds accordingly distant. Additionally, a chat window for user communication and a status window with a latency overview (for example, the ping times) can be opened. Users can log onto an already existing peerSynth session over the menu and toolbar where the modern ASIO audio interface (which is characterized by especially short latency times of direct audio output) can be configured.

peerSynth is written using an OOP environment from Borland (Delphi) and is based on modular and switchable real time sound synthesis and interface components developed by the author [19].

2.6 Network establishing techniques

In a LAN, peerSynth can automatically establish its P2P network based on a UDP broadcast request. Connections over the internet are manually established either by entering the IP address of a known participant in an already existing peer-Synth session or by entering a simple domain name (e.g. peersynth.dyndns.org) that is provided by a dynamic DNS service (e.g. www.dyndns.org). As soon as peerSynth is connected with another instance, it will automatically gather information about the other instances in the network and update these accordingly. Simple and variable network topologies and user hierarchies (e.g., conductor, performer, remixer, listener, etc) can be represented using different filters and data connection modes.

3 Conclusion

peerSynth is a new instrument which enables multiple musicians to play together over the Internet. peerSynth neither ignores nor bypasses latency but rather integrates it as a necessary aesthetic property of the medium through a “time to space” technique. The usual client-server model, which has up till now been used in multi-user instruments, is replaced through a modern, self-configured P2P network in which the participants can be linked together in different configurations. Together with current audio and user interfaces a high degree of usability of the synthesizer is enabled. A complex granular synthesis and subsequent “time to space” sound processing technique constitute the unmistakable sonic characteristics of this new instrument.

peerSynth is distributed and available from the author’s website at <http://www.peersynth.com>.



Figure 3: peerSynth user interface with 3 users connected over a P2P network. More information from www.peersynth.com

4 References

- [1] Yamagishi, Seionshin; Setoh, Kohji: Variations for WWW - Network Music by MAX and the WWW, In: Proceedings of the ICMC, 1998; <http://platinum.sfc.keio.ac.jp/~sns/va/index.html>.
- [2] Weinberg, Gil: The Aesthetics, History, and Future Challenges of Interconnected Music Networks, In: Proceedings of the ICMC, 2002.
- [3] Föllmer, Golo: Soft Music, <http://crossfade.walkerart.org>; see also Föllmer, Golo; Ungeheuer, Elena: Netzmusik-Stand der elektroakustischen Musik oder Musik von anderen Planeten? Ein Printchat, In: Ungeheuer, Elena (Ed.): Elektroakustische Musik - Handbuch der Musik im 20. Jahrhundert Band 5 Laaber: Laaber-Verlag, S. 303-316, 2002.
- [4] Hajdu, Georg: quintet.net, <http://www.quintet-net.org>.
- [5] Brown, Chris: The Eternal Network Music Site http://www.mills.edu/LIFE/CCM/Eternal_Network_Music.html.
- [6] Waag Society: KeyWorx <http://www.keyworx.org>.
- [7] Jordà, Sergi: Faust Music On Line (FMOL): An approach to Real-time Collective Composition on the Internet, In: Leonardo Music Journal, Cambridge, Massachusetts: MIT Press, Vol. 9, S. 5-12, 1999; <http://www.iua.upf.es/~sergi/FMOL/>.
- [8] Yee-King, Matthew John: AudioServe <http://www.yeeking.net/index.php?location=AudioServe>.
- [9] Rocket Network Inc.: Rocket Network Homepage was shutdown on March, 2003.
- [10] Brandt, Eli; Dannenberg, Roger B.: Time in Distributed Real-Time Systems, In: Proceedings of the ICMC, 1999.
- [11] Burk, Philip: Webdrum, <http://www.transjam.com/webdrum/webdrum.html>.
- [12] Pazel, Donald P. et al.: A distributed interactive music application using harmonic constraint, In: Proceedings of the ICMC San Francisco, USA, 2000.
- [13] Ito, Koji et al. NetRezonator, <http://netrezonator.imgsrc.co.jp>.
- [14] Schuett, Nathan: The Effects of Latency on Ensemble Performance, <http://www-ccrma.stanford.edu/groups/soundwire/performdelay.pdf>.
- [15] Helmuth, Mara: Sound exchange and performance on internet2, In: Proceedings of the ICMC, 2000.
- [16] Tanaka, Atau: Speed of Sound, In: Machine Times. V2_organization, Rotterdam, 2000; <http://sensorband.com/atau/globalstring>.
- [17] Rasch, Rudolf A.: Timing and synchronisation in ensemble performance, In: Sloboda, John A (Ed.): Generative Processes in Music - The Psychology of Performance, Improvisation and Composition Oxford: Clarendon Press, p. 70, 1988.
- [18] Stelkens, Jörg: crusherX-Live!, <http://www.crusher-x.de>.
- [19] Stelkens, Jörg: büro </stelkens>, <http://www.stelkens.de>