Network Latency Adaptive Tempo in the Public Sound Objects System

Álvaro Barbosa [†‡]
[†] Music Technology Group
Pompeu Fabra University
Calle Ocata 1, 08003
Barcelona, Spain
+34 93 542 21 04
abarbosa@iua.upf.es

Jorge Cardoso
[‡] Research Center for Science and Technology of the Arts (CITAR)
Portuguese Catholic University
Rua Diogo Botelho 1327,
4169-005 Porto, Portugal
+351 22 616 62 91
jccardoso@porto.ucp.pt

Gunter Geiger
[†] Music Technology Group
Pompeu Fabra University
Calle Ocata 1, 08003
Barcelona, Spain
+34 93 542 21 04
ggeiger@iua.upf.es

ABSTRACT

In recent years Computer Network-Music has increasingly captured the attention of the Computer Music Community. With the advent of Internet communication, geographical displacement amongst the participants of a computer mediated music performance achieved world wide extension. However, when established over long distance networks, this form of musical communication has a fundamental problem: network latency (or net-delay) is an impediment for real-time collaboration. From a recent study, carried out by the authors, a relation between network latency tolerance and Music Tempo was established. This result emerged from an experiment, in which simulated network latency conditions were applied to the performance of different musicians playing jazz standard tunes.

The Public Sound Objects (PSOs) project is web-based shared musical space, which has been an experimental framework to implement and test different approaches for on-line music communication. This paper describe features implemented in the latest version of the PSOs system, including the notion of a network-music instrument incorporating latency as a software function, by dynamically adapting its tempo to the communication delay measured in real-time.

Keywords

Network Music Instruments; Latency in Real-Time Performance; Interface-Decoupled Electronic Musical Instruments; Behavioral Driven Interfaces; Collaborative Remote Music Performance;

1. INTRODUCTION

Research work in the Network-Music field has recently been published in surveys from Álvaro Barbosa in 2003 [1], Gill Weinberg in 2002 [2] and Dante Tanzi in 2001 [3], which describe and categorize several different systems, following diverse architectures and topologies.

These systems make use of a long distance communication media with specific characteristics which must be dealt with. Network latency and Jitter represent the most distinguishable difference from presential music collaboration paradigms, since music performance is traditionally bounded to the notion of real-time synchronism between instruments and performers.

It can be demonstrated that at a globe level there are physical limitations in current network technology, which will always introduce higher latency than the minimum acceptable values for real-time acoustic collaboration [1] [4] [5].

A particular approach to face such scenario is accepting net-delay as a natural element when creating music over the internet. The thought that net-delay is the particular acoustics of Internet and that composers should try to *find a musical language that works on this time axis* is clearly expressed by the experimental artist Atau Tanaka in [6]. The concept of an Internet acoustic space and the influence of network conditions in acoustic communication has also been addressed by Chris Chafe from the SoundWIRE group, at Stanford University's Center for Research in Music and Acoustics (CCRMA) [7].

2. TEMPO AND LATENCY

A number of experiments have been carried out with the purpose of determining the maximum amount of communication latency which can be tolerated between musicians in order to keep up with a synchronous performance.

Some of the most significant results regarding the effects of time delay on ensemble accuracy were published in 2004 by Chris Chafe and Michael Gurevish [8]. From the experiment conducted at CCRMA it is clear that by increasing the communication delay between pairs of subjects trying to synchronize a clapping steady rhythm, the subjects tend to slow down the performance rhythm (Tempo).

Similarly, an experiment carried out by the authors in June 2004 at the Sound and Image Department from the Portuguese Catholic University aimed, amongst other goals, to study the relationship between Tempo and Latency.

In the experiment, simulated network latency conditions were applied to the performance of four different musicians playing jazz standard tunes with four different instruments (Bass, Percussion, Piano and Guitar).

In a studio setup, musicians would listen to the feed-back from their own instruments trough headphones with delay. Their performance was synchronized with a metronome over several takes with different tempos (Beats Per Minute – BPMs). For each take the feed-back delay was increased until the musician wasn't able to keep up a synchronous performance.

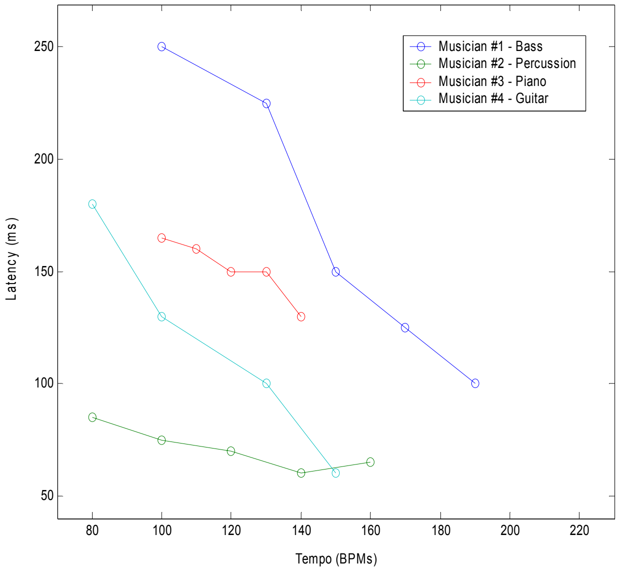

Figure 1. Self-Test for latency tolerance in individual performance of 4 different musicians

The Graphic on Figure 1 shows that, regardless of the instrumental skills or the music instrument, all musicians were able to tolerate more feed-back delay for slower Tempos, therefore, it is clear that there is an inverse relationship between Tempo an Latency. After obtaining these results the authors proceeded to implement this concept on the Public Sound Objects (PSOs) system, aiming to achieve a network-music instrument which incorporates latency as a software feature, by dynamically adapting its tempo to the communication delay measured in real-time.

3. THE PUBLIC SOUND OBJECTS

The Public Sound Objects (PSOs) project is web-based Collaborative Virtual Environment focused on music performance, developed at the Music Technology Group of the Pompeu Fabra University. This project has been an experimental framework to implement and test different approaches for on-line music communication. A preliminary specification of the system was published in [9], and the first prototype was implemented in December 2002[1].

The overall system architecture was designed along the following key aspects: (a) It is based on a Centralized Server Topology supporting multiple users connected simultaneously and communicating amongst themselves through sound; (b) It is a permanent public event available both to a "real world" and on-line virtual audience.

In this system the raw materials provided to the users for manipulation during a performance are Sound Objects, according to Pierre Schaeffer's definition "*any sound phenomenon or event perceived as a coherent whole (...) regardless of its source or meaning*" [10].

These Sound Objects are triggered at the server-side real-time sound synthesizer according to the user's action. Since the Feed-Back from other user's performance is strictly auditory, the characteristic which makes a Sound Object distinguishable from the overall *soundscape* is the key element that permits the awareness of the individual action of a user over his performance.

4. THE PSOs ARCHITECTURE

The PSOs system is composed by the PSOs Server and by multiple PSOs Clients. Clients handle a visual interactive interface and the server handles all computation regarding the sound synthesis and transformation..

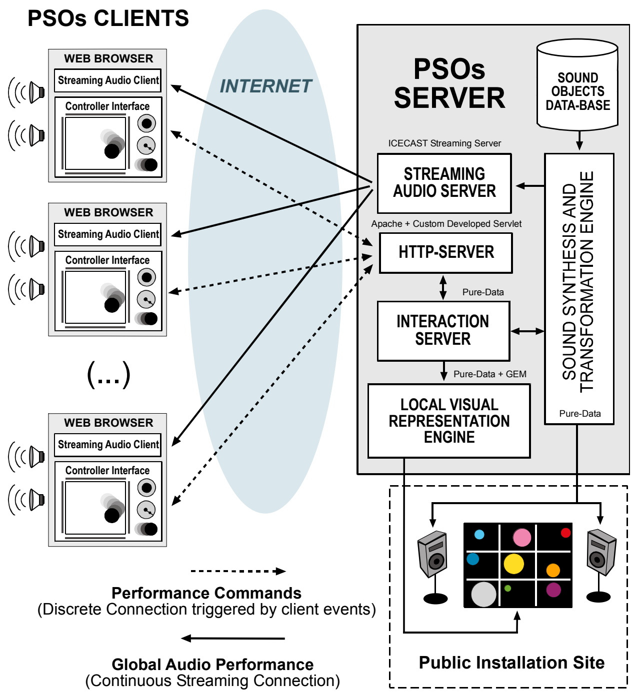

Figure 2. The PSOs System Architecture

Clients communicate with the server through HTTP by sending and receiving packets of data. At the server packets are received by a Web application that reroutes them to the Interaction Server – a module of the PSOs Server that manages clients, instruments and the events generated by the PSOs Client. Depending upon the type of data packet received, a sound can be generated by the Synthesis and Transformation Engine and streamed back to the client by the Streaming Audio Server, or the visual representation of the client can be updated at the installation site by the Local Visual Representation Engine, or both.

Server and Clients are composed by different modules:

[1] The PSOs experimental system is publicly available on-line from the address: http://www.iua.upf.es/~abarbosa/

4.1 HTTP Server

Clients connect to the PSOs Server through standard HTTP connections. Although our initial choice was to implement UDP based communications – faster than a TCP based protocol like HTTP – the idea had to be abandoned, since most firewalls block all unknown UDP traffic which meant that a great number of users would not be able to access our server. In order to overcome these restrictions a communication system was realized using a "firewall ALWAYS ALLOW" protocol: HTTP. For this a server application was implemented, using the Java Servlet technology, which acts as a proxy between the PSOs Client applet and the Interaction Server.

4.2 Interaction Server

The Interaction Server is a central piece in the PSOs Server. It's a Pure Data (PD) module that receives data packets in the form of UDP datagrams from the clients (through the HTTP Server) and acts accordingly to the type of packet received.

A custom PD object had to be implemented for the reception of the UDP datagrams – which was called Extended Netreceive [`xnetreceive`] – since existing objects for this propose don't allow PD to acquire the IP address and port number of the client that initiated the communication.

4.3 Synthesis and Transformation Engine

The Synthesis and Transformation Engine is responsible for the sound generation in response to the PSOs Clients' generated events. This engine is a PD patch automatically loaded by the Interaction Server. The engine has nine synthesis modules that correspond to the Sound Object's instruments available to users. The sound generated by these modules is streamed in MP3 format, using the [`shoutcast~`] PD object to an audio streaming server. Users can choose one of these modules from the client entry screen.

4.4 The Client User Interface

Once selected the Sound generating engine (instrument), the web-browser loads a controller interface java applet, which connects to the interaction server, registers and initializes a user session.

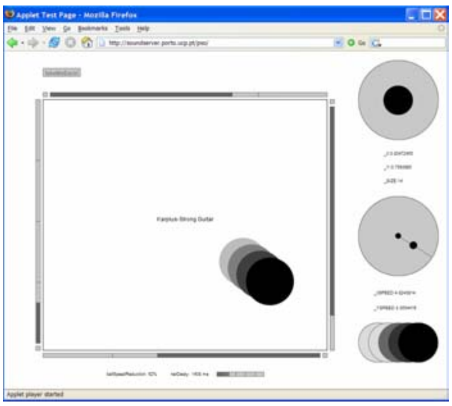

Figure 3. PSOs Client controller interface

The graphical user interface presented in Figure 4, follows a metaphor of a ball that infinitely bounces on the walls of an empty room. When the ball hits one of the walls a network message is sent to the central server where the corresponding Sound Object is triggered, played through a specific source speaker and simultaneously streamed back to the user in a stereo mix of all the sounds being triggered at the moment.

The ball moves continuously and the user can manipulate its size, speed, direction, tail extension and each wall's acoustic texture. Normalized values are then sent to the server and mapped to synthesis parameters. The wall's acoustic texture matches the Sound Object's pitch (individual pitch values can be assigned to each wall, allowing the creation of melodic and rhythmic sound structures) and the ball's tail extension corresponds to the number of replicas of the delay applied to the Sound Object.

4.5 Local Visual Representation Engine

The Local Visual Representation Engine outputs the visual representation of the bouncing ball model of all the connected PSOs Clients, at the servers physical location. It consists of a PD patch that uses the Graphics Environment for Multimedia (GEM) external for graphics output, using information from each client to update the state information. The visual setup is composed by a video wall with nine screens arranged in a 3 by 3 matrix. Each screen is assigned to an instrument in the same order as they are presented to the user in the PSOs Client interface.

Figure 4. PSOs Installation Site setup

The clients are represented at the installation site as spheres with different colours, sizes and speed. Each client is assigned to a screen in the video wall which also limits the movement of the corresponding sphere, i.e., the limits of each screen are mapped to the limits of the PSOs Client's window. The engine only has accurate information when clients send packets to the server. The rest of the time, the position of the bouncing ball has to be interpolated based on the information from the last packet.

4.6 Network Latency Adapted Tempo

The idea of a network music instrument which dynamically adapts to internet network-latency was implemented recently by Jörg Stelken in the peerSynth Software [11]. PeerSynth is a per-to-peer sound synthesizer which supports multiple users displaced over the internet, measuring the latency between each active connection and dynamically lowering the sound volume of each user's contribution in the incoming *soundscape*, proportionally to the amount of delay measured in his connection. Stelkens followed a real world metaphor were, in fact, the sound volume of a sound source decreases with the distance to the receiver, which also implies increasing acoustical communication latency. A similar approach was followed in the AALIVENET System [12].

The PSOSs system approaches this same idea, but addressing a less immediate, but equally relevant, relation between musical characteristics and communication latency. It implements a Network Tempo Adaptive Latency feature, which dynamically reduces the performance tempo according to the latency measured in real-time between the client and the server.

In the Bouncing Ball user interface the musical tempo correspond to the ball speed. The reduction factor applied to the ball speed is presented in the user interface and its calculated so that it averagely guarantees that the ball will not hit the walls twice without the sound triggered by the time the first hit arrives to the client. The main idea is that *the ball will go as fast as your connection allows you to go*. This way the effect of latency is much less confusing, permitting the user to have a much better awareness of the relationship between a hit on the wall and the corresponding triggered sound.

5. PSOs USER STUDY AND EVALUATION

The complete PSOs System, including the physical setup at the server site, was installed at the Portuguese Catholic University Campus in Porto between 7 and 14 of October 2004. During this trial period several client instances were installed on campus and 109 subjects tested the system and answered questionnaires. Some of the average results extracted from this opinion pool are presented in Figure 6.

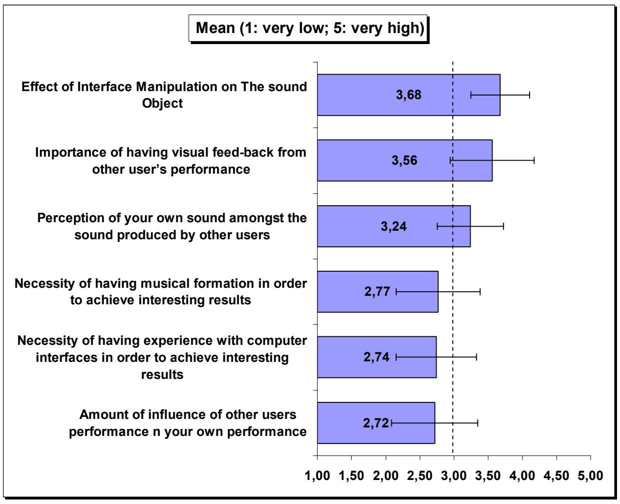

Figure 5. PSOs Preliminary Evaluation Results

Some of these results meet our expectations: (1) The interface is effective establishing a relation between the user action and its effect on the correspondent Sound Object; (2) The Sound Objects available at the current setup allow acoustic differentiation in the global *soundscape*; (3) It is a system accessible to the general public, without requiring previous music formation or previous GUI manipulation skills. On the other hand it seams that in general users feel the need to have a visual representation of other user's behavior, and maybe for this reason, the collaborative effect amongst performers is relatively low.

6. CONCLUSIONS AND FUTURE WORK

For over two years the Public Sound Objects project has successfully performed as an experimental framework to implement and test different approaches for on-line music communication. The recent user study and evaluation confirmed some of the main system design ideas to be effective, however it also suggests further improvements, such as a visual representation of other users at the client instance, and possibly sound synthesis at the client side using a PD web-browser plug-in.

The Network Tempo Adaptive Latency, implemented in the PSOs latest version, represents a significant improvement in the system usability and further developments will be explored in order to improve latency tolerance, specifically in the bouncing ball interface, using spatialization features.

7. AKNOWLADGEMENTS

The Authors would like to thank Martin Kaltenbrunner for his work in the early development stage of PSOs and Alexander Carôt for his collaboration in the Delay *VS* Tempo Experiment. This research is supported by “Fundação para a Ciência e Tecnologia” – Portugal.

8. REFERENCES

[1] A. Barbosa, *Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation*. Leonardo Music Journal, Volume *13,* Issue 1, pp. 53-59, MIT Press, Cambridge MA, 2003.

[2] G. Weinberg. *The Aesthetics, History, and Future Challenges of Interconnected Music Networks*. 349-356. 2002. Proceedings of the International Computer Music Conference.

[3] D. Tanzi, *Observations about Music and Decentralized Environments*. Leonardo Music Journal, Volume *34,* Issue 5, pp. 431-436, 2001.

[4] N. Schuett. *The Effects of Latency on Ensemble Performance*. 2002. Stanford University.

[5] N. Lago and F. Kon. *The Quest for Low latency*. 33-36. 2004. Proceedings of the International Computer Music Conference (ICMC2004).

[6] Atau Tanaka, *Musical Implications of Media and Network Infrastructures: Perturbations of traditional artistic roles.* In *Hypertextes hypermédias, nouvelles écritures, nouveaux langages*. pp. 241-250, Hermes Science Publications, 2001.

[7] C. Chafe, S. Wilson and D. Walling. *Physical Model Synthesis with Application to Internet Acoustics*. 2002. IEEE - Signal Procesing Society. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Orlando - Florida (ICASSP2002).

[8] C. Chafe, M. Gurevich, Grace Leslie and Sean Tyan. *Effect of Time Delay on Ensemble Accuracy*. 2004. Proceedings of the International Symposium on Musical Acoustics, Nara - Japan (ISMA2004).

[9] A. Barbosa and M. Kaltenbrunner. *Public Sound Objects: A shared musical space on the web*. 2002. IEEE Computer Society Press. Proceedings of International Conference on Web Delivering of Music (WEDELMUSIC 2002) - Darmstadt, Germany.

[10] P. Schaeffer, *Traité des Objets Musicaux.*, 1966.

[11] J. Stelkens. *peerSynth: A P2P Multi-User Software with new techniques for integrating latency in real time collaboration.* 2003. Proceedings of the International Computer Music Conference (ICMC2003).

[12] M. Spicer. *AALIVENET: An agent based distributed interactive composition environment*. 2004. Proceedings of the International Computer Music Conference (ICMC2004).