nJam User Experiments: Enabling Remote Musical Interaction from Milliseconds to Seconds

Nicolas Bouillot
CEDRIC-CNAM
292 rue St Martin
75141 Paris cedex 03, France
bouillot@cnam.fr

ABSTRACT

Remote real-time musical interaction is a domain where end-to-end latency is a well known problem. Today, the main explored approach aims to keep it below the musicians perception threshold. In this paper, we explore another approach, where end-to-end delays rise to several seconds, but computed in a controlled (and synchronized) way depending on the structure of the musical pieces. Thanks to our fully distributed prototype called nJam, we perform user experiments to show how this new kind of interactivity breaks the actual end-to-end latency bounds.

Keywords

Remote real-time musical interaction, end-to-end delays, synchronization, user experiments, distributed metronome, NMP.

1. INTRODUCTION

Networked Musical Performances (NMP) are systems where physically remote musicians play "live" music together via a communication network. Such a remote musical interaction brings its own perceptive constraints: audio acquisition quality, end-to-end delays (including sound cards, operating systems and networks delays), and various dimensions as time (acquisition period, temporal consistency [4], tempo), spatiality and size [7]. In other words, NMP systems manage abstractions that could simulate traditional interactions and ideally provide the feeling of co-presence among users.

Today, one of the main challenging task is to manage delays between remote musicians [1, 14]. The existing NMP systems can be divided into two categories: *instantaneous* NMPs and *user controlled delays* NMP. In the first category, NMP systems are designed with low latency protocols in order to create the temporal illusion of being close to each other [14]. Such real time audio streaming systems are described in [13, 12] and the first entertainment event using live audio streaming is described in [8]. [15] provides a MIDI based *instantaneous* NMP system.

In the *instantaneous* NMPs domain, it is widely recognized that end-to-end delays must be kept below human perception. In [16], N. Schuett evaluate the user performance threshold thanks to a user experiment and a tempo analysis: the threshold was set around 50ms. Then, user experiments performed by E. Chew *et al.* in [5, 6] introduce additional delays in musicians feedbacks. The musicians opinions (subjective evaluation) state the maximum delay to 65 ms.

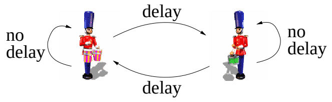
Distance	Distance (km)	Delay (ms)
Lille - Perpignan	883	2.94
Miami - Seattle	5 309	17.7
earth circumference/2	20 000	66.6

However, network delays are today physically bounded by light speed. The previous table shows us the light propagation delays between points on earth (we assume that light speed is 300000m/s). The last value (66.6ms) shows us that NMP with a user performance threshold to 65ms is unachievable across the whole Internet, particularly between far users using DSL and interconnected via multiple Internet Service Providers.

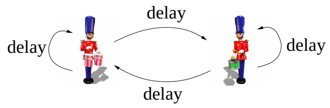
The second category (*user controlled delay* NMPs) aims to provide a musical interaction between musicians despite high end-to-end latencies. The delay management in these systems includes a knowledge of some music parameters (a tempo and a duration in musical units) allowing to compute a "semantic" delay (related to the played music). Examples of such NMP are RMCP/VirJa [11] (using MIDI audio acquisition), ninJam[1] and nJam (our prototype previously described in French in [2]). These systems introduce a delay chosen by the musicians and configured using musical units: a tempo in Beat Per Minutes (BPM) and a duration in quarter note(s). Our approach includes a synchronized global and consistent metronome that help the musicians to anticipate the "semantic" latency.

In this paper, we first describe the nJam's functionalities (section 2). Then, we perform user experiments for the *user controlled delay* NMPs (section 3): couples of musicians (located in different rooms over a LAN) experiment nJam. We use the same approach (subjective evaluation) as [5, 6]: for each couple, various end-to-end latencies are rated by the

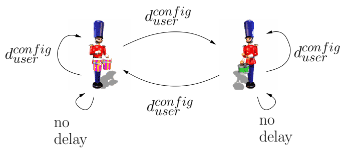
[1] `http://www.ninjam.com`



(A) N. Schuett's audio configuration



(B) E. Chew's audio configuration



(C) nJam's audio configuration

Figure 1: *audio configurations in NMP user experiments*

musicians. With three different scenarios, we explore the effectiveness and the difficulties of the musical interaction through high latency systems, leading us to understand how *user controlled delay* NMPs can lead to comfortable and effective musical interactions. Finally, we conclude in section 4.

2. NJAM

Figure 1 shows audio configurations provided by Schuett [16] Chew [5] and nJam. Our one allows the musicians to hear their own direct sound without delay, but also simultaneously with the global and synchronized mixing, delayed from the value configured (denoted d_{user}^{config}).
Musicians configure nJam with a message as the following one:

```
start 180 12 1 3
```

where 180 (first parameter) is the tempo in Beat Per Minutes (BPM) and 12 (second parameter) is the duration of the semantic lag (d_{user}^{config}) specified in quarter notes. Tempo and duration allow nJam to compute the semantic lag duration in other time units.

The nJam configuration let musician hearing two audio streams that run on two different temporalities. This setting comes with side effects: harmonic modulations or rhythmic changes could produce harmonic or rhythmic dissonances for a duration equivalent to the local lag (d_{user}^{config}). Even if dissonances are sometimes used consciously by musicians (modern music, Jazz, traditional world music, etc.), these new kinds of dissonances must be evaluated in a user point of view.

Technically, nJam is prototyped as a jMax [9] object and support multi-sites interactions. It is composed of the four followings components, that run without central server:

- a **streaming engine** that uses the Real time Transport Protocol ([17]) and IP multicast. It enables a full duplex multi-musicians communication among physically distant users
- a **multi-streams synchronization protocol** that provides a consistent mixing for each musician
- a **latency adaptation** that provides a local feedback latency according to the musicians configuration (a musical duration and a tempo) for a session
- a **global metronome** built over the synchronization that gives a beat and help musicians to anticipate the latency.

Before playing, musicians have to set the semantic lag (metronome and duration parameters). Then, nJam runs the streaming engine, the synchronization, the latency adaptation and the metronome. At this time nJam provides a global mixing rhythmically synchronized (according to the metronome) with the music played locally by the musicians.

Figure 2 shows the impacts of the different components during the interaction between two users (Alice and Bernard). In a user point of view, the global mixing and the local direct tempo are in phase.

The **streaming engine** uses IP Multicast to perform a group communication among musicians (the IP network manages the forwarding of a datagram to subscribers according to the destination group address). Then each nJam instance manages buffers to send and receive audio streams from others. We use 44100 Hz 16 bit PCM audio. This streaming engine maintains constant latencies between each couples. Then each nJam instance performs its own mixing of the audio streams. IP multicast allows the system to run in a fully distributed way, without central synchronization server.

As shown in the literature [6, 3, 16], high uncontrolled latencies produce inconsistencies and affect the musical interactivity. In this way, the **multi-streams synchronization protocol**[2] provides the simultaneity property among global mixings. In other words: "if two audio samples coming from different sources are mixed together for a musician, then other musicians will mix these samples together". This protocol can be seen as a physical clock synchronization that provides a consistent vector of physical clocks. More details of this protocol are given in [2] and [4]: after initialization, the protocol introduces delays inside reception buffers to obtain the simultaneity property. This is illustrated in figure 2, where d_{alice}^{algo} is the delay added inside the Alice local buffer receiving its own samples. At this time, the nJams' global mixings are "simultaneous", but each global mixing are still not in phase with the music played by the local musician (beats are still not synchronized).

Such a rhythmical "phase" is obtained via the (**latency adaptation**) component, by adding an additional delay (d^{adapt}) inside the reception buffers. It raises the perception of the global mixing to the latency configured initially by musicians (d_{user}^{config}). For example, Alice's nJam instance computes its adaptation delay as follow: $d_{Alice}^{adapt} = d_{Alice}^{config} - d_{Alice}^{algo}$. Then, Alice's local buffers are all delayed of d_{Alice}^{adapt} time units. Let us remark that the simultaneity property is maintained. Figure 2 shows us that the latency adaptation builds a rhythmical synchronization between the music played and the global mixing.

[2]In [4], it is called the perceptive consistency protocol.

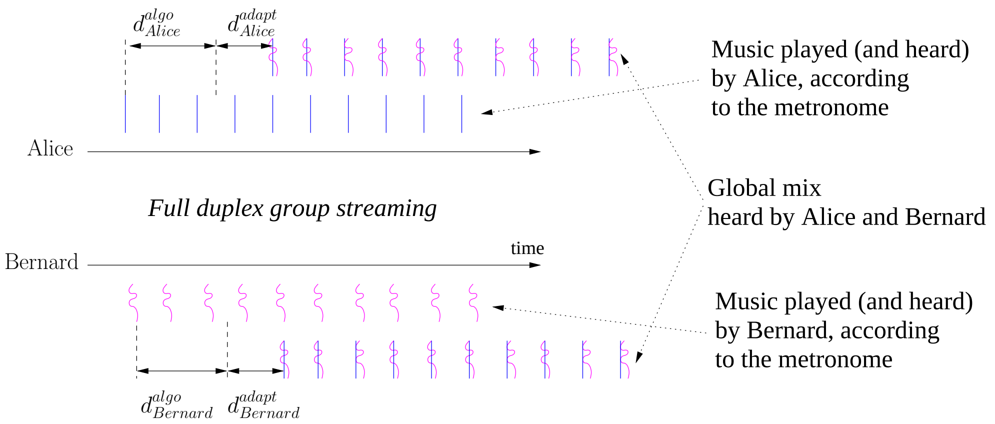


Figure 2: ***the nJam solution***

The third and fourth parameters of the nJam initialization message are used to configure the metronome period: "1 3" means that: a measure is composed of three quarter notes and each quarter notes are played. A **metronome** beat (or event) is computed thanks to the vector of physical clocks provided by the synchronization: when the synchronized current time modulo the metronome period is equals to 0, a metronome event is produced. This allows the nJam instances to share consistent metronomes.

3. NJAM USER EXPERIMENTS

As shown in section 2, nJam provides a musical interaction with local and perceptible end-to-end latencies that can be set to several seconds. With the nJam user experiments, we aim to answer the following questions :

1) Does nJam's abstractions (user controlled delays and global metronome) enhance the remote musical interactivity ?

2) Is musical interaction achievable with an end-to-end delay equals to several seconds ?

3) What kind of music musicians are able to play with nJam ?

In order to answer these questions, we experiment three scenarios. In the first one, we ask two musicians to play a Spanish musical pattern together in the same room, and then via nJam and an Ethernet LAN. For this first scenario, we mute the metronome volume to run several experiments instances with various end-to-end delays, from 300ms to 9.6s. The second scenario is similar to the first one, but we activate the global metronome. The following table shows the latency experimented during these scenarios, but also the same duration specified in quarter notes, if the music is performed at a 100 BPM tempo[3]:

Latency	300ms	600ms	2.4s	7.2s	9.6s
At 100 BPM	1 e.n.	1 q.n.	4 q.n.	12 q.n.	16 q.n.

In both firsts scenarios, we use the same musical pattern (figure 3[4]), taken from the flamenco traditional music. This

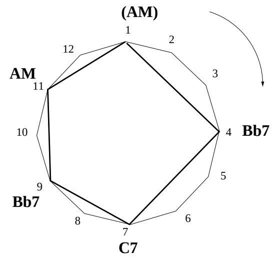


Figure 3: ***The Spanish traditional pattern***

pattern is 12 quarter notes longer. Then, if repeated, the duration of a loop takes 7,2 seconds. On figure 3, the 12 quarter notes are numbered from 1 to 12 and the inscribed polygon summits show the quarter notes for which musicians may play the specified chord. For example, musicians may play a Bb7 at the fourth quarter note.

We choose such a pattern for three reasons : (1) it can be looped with a 7.2s period (at 100 BPM), (2) it is rhythmically asymmetric and (3) the whole majority of the musicians experimenting nJam are not accustomed with this kind of pattern.

The third scenario is technically the same as the second, but the musicians can choose the music they play. Then, we set the end-to-end latency according to the pattern or the song they want to play. In these experiments, only one end-to-end delay is chosen for each couple of musicians.

Musicians comment on and evaluate experiments by answering at the question "estimate the facility/difficulty to play music via a network with the current configuration". The answer is rated from 1 to 7 (as in [5]) where 1 means "very difficult", 4 means "ok with training" and 7 means "very easy".

In addition, for each experiment and for each musician we record the following audio streams: (1) the metronome mixed with the music played by the local musician[5] (2) the music played mixed with the global mixing and (3) the

[3] e.n. means eighth note and q.n. means quarter note(s).
[4] We use a notation close to [10].

[5] During the first scenario, the metronome is muted. Thanks to an audio editor, we can measure the pattern period and then the tempo really played by the musicians.

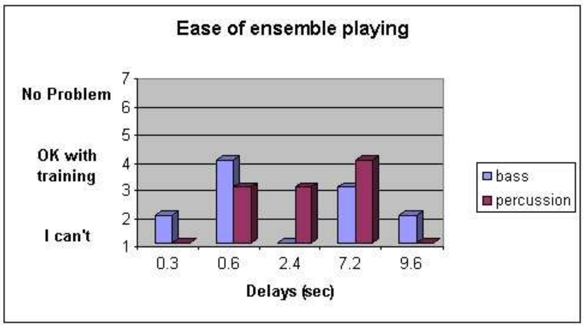


Figure 4: ***user ratings of the first scenario***

global mixing only. These records allow us to estimate the musicians rhythms accuracy, but also to estimate the "phasing" between the global mixing and the music played by musicians[6].

3.1 First scenario results

We perform this scenario with one couple of musicians (a bassist and a percussionist) that have never played together before. As they are not accustomed to play flamenco pieces, they provided additional concentration efforts to perform the pattern, even when they play it together in the same room (see video).

With the audio records, we notice that musicians choose different strategies to play via nJam. The bassist follows the rhythm of the global mixing (in fact, the percussionist beat). This is confirmed by the global mixing recordings.

The percussionist chooses a leader strategy: he accelerate or decelerate its tempo to be consistent with his own feedback (excepted for the 300ms end-to-end latency experiment). We measure the real tempo played by the percussionist during these experiments with an audio editor. Results are given in the following table:

Latency	300ms	600ms	2.4s	7.2s	9.6s
Tempo (BPM)	93	95-98	116-122	117	105

Figure 4 shows us the users' opinions about the first scenario. We notice that the 600ms and 7.2s experiments obtain the best ratings. This could be explained by the fact that these latencies are correspond to a quarter note at a 100 BPM tempo and to the pattern duration.

For the 600ms experiment, the percussionist remarks that "the beat is given by the feedback": this is consistent with the tempo measurements, where the percussionist play at the tempo closest to 100 BPM.

The success of the 7.2 seconds experiment could be explained by the fact that the latency is sufficiently high to superimpose the played pattern on the delayed pattern. This can be heard on the percussionist's record (2): he accelerates and decelerates to phase with his own feedback.

[6]Videos and audio recording are available at the following URL:
`http://cedric.cnam.fr/~bouill_n/nJamUserExperiment/`

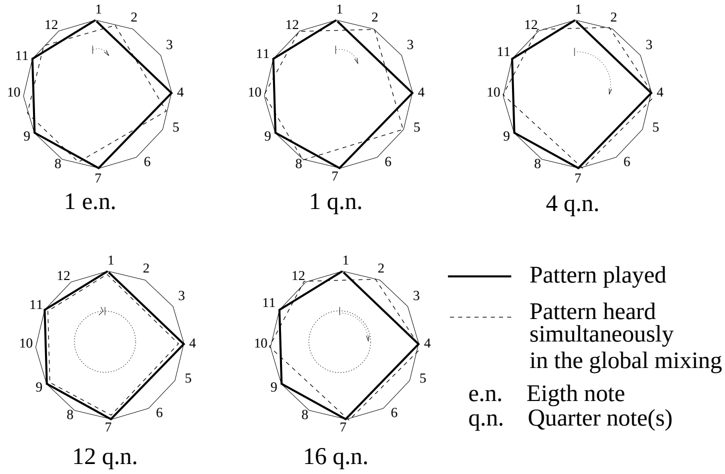


Figure 5: ***The phasing between the pattern played and the pattern heard***

Experiments with other latencies obtain worst ratings because the musicians feel rhythmically disturbed. The followings points conclude this scenario analysis :

- Musicians think that they are able to play music without metronome although a high latency but after training
- The acceptable perceivable latencies seems to be related to the tempo and the music played
- Finding the appropriate tempo to anticipate the delay is difficult for musicians

3.2 Second scenario results

During the second scenario, we run the same experiments (various latencies), but we activate the global metronome: each musician hears locally a beat that provides a consistent global tempo. Then, it is useless to measure the tempo played by the musicians as they always play at 100 BPM, according to the metronome settings. We experiment this scenario with two couples of musicians: the bassist/percussionist of the first scenario and a couple guitarist/flamenco singer. As I play guitar, I do not rate the question.

Users' ratings are shown by figure 6: the 7.2 seconds experiment obtains results higher than others, excepted the high rate given by the singer at the 2.4 seconds experiment (4 quarter notes). The result of this last experiment would be explained later (see singer's remarks).

With the audio recordings, we notice that musicians are always listening to the other musicians (the mixing) to play the pattern, whatever the experiment. We remark that musicians correctly perform the pattern for the 7.2s experiment only, that is consistent to the musicians' ratings.

We can also hear that during the 4 and 16 quarter notes experiments, musicians perceive difficulties to accurately perform the Spanish pattern. This probably comes from the phasing problem described by the figure 5. The pattern played by the musician is phasing with the mixing on the

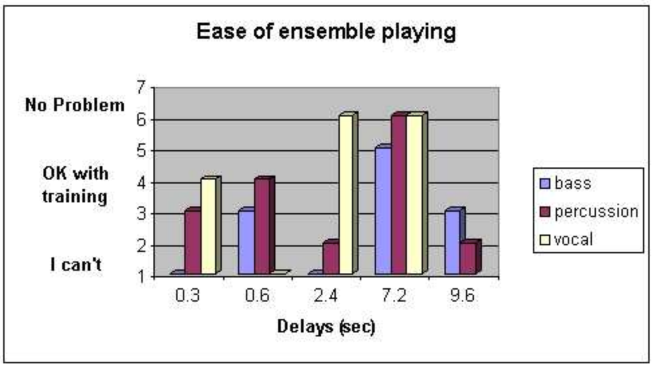


Figure 6: ***user ratings of the second scenario***

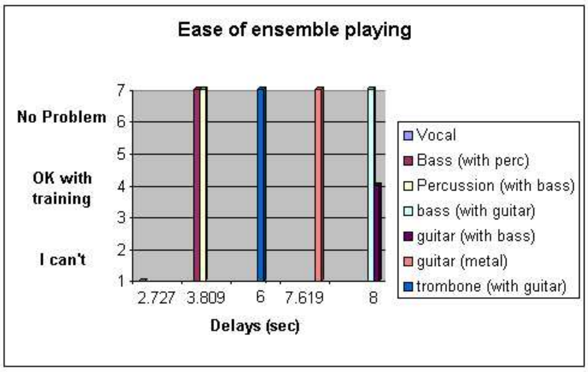


Figure 7: ***user ratings of the second scenario***

7.2s experiment only.

The following points give the singer observations for different experiments:

- 4 quarter notes latency: *This is ok because the feedback comes when I am not singing, but the guitarist seems to have difficulties, I heard it.*
- 12 quarter notes latency: *Really good because the latency correspond to the measure, I hear my voice as an other chorister*

This scenario shows us that the metronome enhances remote musical interaction because musicians are not focusing any more on the tempo accuracy. We also notice that the playing easiness is particularly enhanced when the configured latency is chosen according to a loop duration.

3.3 Third scenario results

In this scenario, musicians play a pattern or a song they have chosen. After a short explanation of the system, the delay is set according to their choices. In this way, we experiment only one delay for each couple. We perform this scenario with five couples. As I participate to three of them, I do not rate the question. A description of the different settings is given by the following points:

1) the bassist/percussionist couple from previous scenarios chooses a rhythmic pattern with a 3.809 seconds delay (8 quarter notes at 126 BPM)
2) the couple singer/guitarist from the second scenario chooses a Cuban song (*not pattern based*) with a 2.727 seconds delay (4 quarter notes at 88 BPM)
3) a couple bassist/guitarist chooses a Jazz pattern to perform an improvisation with a 8 seconds delay (16 quarter notes at 120 BPM)
4) a couple of two guitarists chooses a rock pattern (a riff) to perform rock solos with a 7.619 seconds delay (16 quarter notes at 126 BPM)
5) a couple trombonist/guitarist chooses a Bossa/Jazz pattern to perform an improvisation with a 6 seconds delay (16 quarter notes at 160 BPM)

Figure 7 shows us this scenario results. It obtains the maximum ratings, excepted the for song played by the couple singer/guitarist (third couple). This can be explained by the fact that the song is not pattern based: the rhythmic (and harmonic) superimpositions are "illogical"and remain continuously during the performance. In this case, harmonic chord changes are hard to perform. However, interactivity and synchronized start seems to remain possible (see also the blues recording during our 2003 rehearsal[7]).

With the pattern based music experiments, nJam obtains the maximum ratings. These ratings do not depend on the musicians levels (from one year of studies with a teacher to eight years of academy). Hereafter, we provide some remarks coming from the musicians about these experiments:

- trombonist: *I am accustomed to sample pedals (or "looper"). I do not feel any difficulty with nJam*
- bassist: *I can stop listening to the metronome after a couple of loops, but if I am lost, it helps me to keep the tempo. In fact, a mistake is repaired after an entire loop*
- percussionist: *It is possible to create a new kind of playing with this latency parameter. This would be a new improvisation concept. For example, musical questions and answers would take place on two different temporalities: the present and the past !*

4. CONCLUSION, DISCUSSION AND FUTURE WORKS

In this paper, we present our prototype (nJam) and its user experiments. It is built of (1) a streaming engine that provides multi-users group listening (2) a fully distributed consistency protocol that synchronizes remote mixings, (3) a latency adaptation configured by musicians that delays the mixing to be rhythmically "in phase" with the music played and (4) a global and distributed metronome.

In order to evaluate the nJam effectiveness, we conduct a set of experiments based on three scenarios. First scenario shows that without metronome, remote musical interactivity is achievable if delays correspond to some structural duration inside the musical piece. These durations (7.2s in our experiment) are closely related to tempo, measure(s) and

[7] `http://cedric.cnam.fr/~bouill_n/`

loops that constitute the structure of the piece. Then, we show (second scenario) that nJam's metronome is a significant enhancement to the users' comfort. While we impose a musical pattern during first and second scenarios, the musicians were free to choose the music and the adaptation delay for the third one. This scenario obtains the maximum ratings for various latencies from 2.7s to 8s.

Experiments show that nJam is particularly efficient when it is used to play music based on loops with a constant tempo (as many kinds of music like some kinds of Jazz, African and Asian world music, electronic music, etc.). In fact, the simultaneous hearing of the music played and the global consistent mixing could produce dissonances during changes. However, as we successfully experiment a blues during the "Resonances 2003" rehearsal, we think that training could allow musicians to perform other kinds of musical pieces.

As mentioned by the musicians during experiments, the interaction provided by nJam is not limiting as it provides a new kind of musical interaction: different temporalities (the present and the past) are mixed together and allow musicians to play with. Questions and answers, harmonic and rhythmic changes could be performed in a new way. Future works will include controlled delays inside the artistic creation and/or composition. Technically, other "user oriented" abstractions as spatiality and haptic will be experimented in order to enhance the NMPs' interaction.

5. ACKNOWLEDGMENTS

I would like to thank Jeremy Cooperstock for its helpful assistance during the work plan elaboration of these user experiments. I would also thank the peoples who played during the experiments or lend us microphones and camera: Laurent Alibert, Patrice Bouillot, Jose Camarena, Julien Cordry, Michael Escande, Jean-Frederic Etienne, Jean-Marc Farinone and Olivier Veneri.

6. REFERENCES

[1] R. Bargar, S. Church, A. Fukuda, J. Grunke, D. Keislar, B. Moses, B. Novak, B. Pennycook, Z. Settel, J. Strawn, P. Wiser, and W. Woszczyk. AES white paper: Networking audio and music using internet2 and next-generation internet capabilities. Technical report, AES: Audio Engineering Society, 1998.

[2] N. Bouillot. Un algorithme d'auto synchronisation distribuee de flux audio dans le concert virtuel reparti. In *Proc. of The Conference Francaise sur les Systemes d'Exploitation (CFSE'3)*, La Colle sur Loup, France, October 2003.

[3] N. Bouillot. The auditory consistency in distributed music performance: a conductor based synchronization. *ISDM (Info et com Sciences for Decision Making)*, (13):pp. 129–137, February 2004.

[4] N. Bouillot. Fast event ordering and perceptive consistency in time sensitive distributed multiplayer games. In *CGAMES'2005 7th International Conference on Computer Games*, pages 146–152, Angouleme, Nov. 2005.

[5] E. Chew, A. Sawchuk, C. Tanoue, and R. Zimmermann. Segmental Tempo Analysis of Performances in User-Centered Experiments in the Distributed Immersive Performance Project. In *Proceedings of the Sound and Music Computing Conference*, Salerno, Italy, November 24-26 2005.

[6] E. Chew, R. Zimmermann, A. Sawchuk, C. Papadopoulos, C. Kyriakakis, C. Tanoue, D. Desai, M.Pawar, R. Sinha, and W. Meyer. A second report on the user experiments in the distributed immersive performance project. In *5th Open Workshop of MUSICNETWORK*, 2005.

[7] J. Cooperstock. Interacting in shared reality. In *HCI International, Conference on Human-Computer Interaction*, Las Vegas, July 2005.

[8] J. R. Cooperstock and S. P. Spackman. The recording studio that spanned a continent. In *IEEE International Conference on Web Delivering of Music, WEDELMUSIC*, Florence, Italie, 2001.

[9] F. Dechelle, R. Borghesi, N. Orio, and N. Schnell. The jMax environment : an overview of new features. In *ICMC : International Computer Music Conference*, 2000.

[10] J. Diaz-Banez, G. Farigu, F. Gomez, D. Rappaport, and G. Toussaint. El compas flamenco: A phylogennetic analysis. In *proceedings of BRIDGES: Mathematical Connections in Art Music, and Science*, pages 61–70, Winfield, Kansas, 2004.

[11] M. Goto and R. Neyama. Open RemoteGIG:an open-to-the-public distributed session system overcoming network latency. *IPSJ JOURNAL*, 43(2):299–309, February 2002. (in japanese).

[12] X. Gu, M. Dick, U. Noyer, and L. Wolf. NMP - a new networked music performance system. In *Proceedings of the 1st IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME'04)*, Dallas, USA, November 2004. IEEE.

[13] D. Konstantas, Y. Orlarey, S. Gibbs, and O. Carbone. Design and implementation of an ATM based distributed musical rehearsal studio. In *Proceedings of ECMAST'98, 3rd Eur. Conf. on Multimedia Applications, Services and Techniques*, Berlin-Germany, 26 - 28 May 1998.

[14] Z. Kurtisi, X. Gu, and L. Wolf. Enabling network-centric music performance in wide-area networks. *Commun. ACM*, 49(11):52–54, 2006.

[15] J. Lazzaro and J. Wawrzynek. A case for network musical performance. In *11th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, New York, 2001.

[16] N. Schuett. Effects of latency on ensemble performance. Stanford University, May 2002.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. Network Working Group Request for Comments: RFC 3550, July 2003.