NMP

- A new networked music performance system

Xiaoyuan Gu, Matthias Dick, Ulf Noyer, Lars Wolf
Technical University Braunschweig
Institute of Operating Systems and Computer Networks (IBR)
Mühlenpfordtstraße 23, Braunschweig 38106, Germany
E-mail:{xiaogu, dick, unoyer, wolf}@ibr.cs.tu-bs.de
URL: http://www.ibr.cs.tu-bs.de
TEL: +49 531 391 3283

Abstract*—Although IT has penetrated into nearly every aspect of the work and life of human beings, music professionals still stick to the traditional way of carrying out rehearsals and concerts. Music performance in this way requires physical presence of the participants and has a number of inherent limitations. We introduce in this paper a prototype of a novel Networked Music Performance system that enables the music professionals and enthusiasts to play with each other through cyberspace. An application like this is bandwidth demanding, highly delay-sensitive and requires the synchronization of the audio streams. Hence, the support from underlying end-systems and networks is critical. However the current source coding mechanisms and the best-effort nature of the Internet poses many challenges to achieve the desired quality of service. We have implemented the prototype in a Local Area Network environment on Linux PCs. The system contains four major interactive components and enables two different application scenarios of real-time rehearsal and rehearsal on-demand. Multi-channel audio transport and different audio compression schemes are supported. Our evaluation results based on both subjective and objective measurements show that the system suffices the audio quality level for the target application. In the future we will extend our system to larger scale and consider more realistic network conditions in the Internet. Moreover, the support for MPEG-4 AAC codec for even better quality of multi-channel audio streaming and enhanced audio quality under the same bandwidth constrains will be added.

Keywords-networked entertainments, real-time audio streaming, online music performance, distributed musical rehearsal.

I. INTRODUCTION

Information technology, in particular the Internet, has penetrated into nearly all aspects of the work and life of human beings. The World Wide Web represents one of the most common used ways for research and information retrieval, collaborative work, networked entertainments etc. With the great proliferation of the Internet and wide spread availability of broadband, the market in networked entertainment is growing. Networked music, exampled by Internet radio stations and online music stores, has been well established and exploited. Besides the usage of the Internet as music databases, the research communities have seen in recent years the interests in exploring the unique, complex and multi-dimensional nature of the Internet for new paradigms of creating and constructing music on-the-fly. This gives rise to a number of new applications like networked musical compositions, networked conducting, and distributed musical rehearsal. We focus in this paper on the third category: distributed musical rehearsal, which we call Networked Music Performance (NMP) hereafter.

Despite the wide utilization of the Internet, for centuries, music professionals around the world especially those who are engaged in the performance of classical music, have stick to the traditional way of carrying out rehearsals and concerts. Music performance in this way requires physical presence of the musicians and has a number of inherent limitations. First, it is not always an easy task to find out a common timeslot for everybody for a face-to-face meeting. Life is even harder when the participants are spread in different places, due to time and costs for traveling. Finding a player of the desired level can sometimes be frustrating as well, because of the limits of personal contacts. Also it could happen that the musicians use music books/notes of slightly different versions. Hence, there is a basic need to improve the way music enthusiasts and professionals alike perform, for the reasons of flexibility, economy, efficiency, productivity and creativity.

Networked Music Performance represents such a concept that musicians who are physically separated can carry out real-time rehearsals or concerts across the network with acceptable audio quality. Aimed at solving the aforementioned problems which occur in the traditional music performance, NMP is a challenging application where a number of factors complicate this task: 1) *Bandwidth demanding*: As experimented in [1], real-time audio streaming based teleportation (the type of application NMP belongs to) is one of the most bandwidth intensive applications in today's networks. Transmission of mono PCM (raw) CD-quality audio requires a data rate of 0.7Mb/s [2]. When stereo/multi-channel or high definition sound (high sampling rate e.g. 48k/96k/192kHz or better quantized e.g. using 24 bit) is needed, the network can be further stressed. To efficiently use network bandwidth, audio compression represents an essential need. *2) Highly delay-sensitive:* Due to the fact that human hearing is very sensitive to delayed or missing information in music, especially that played on fine acoustic instruments, the pre-buffering mechanism that is common in most Internet music systems today simply will not help when contents are generated on-the-fly and intensive interactivity is a must. Research results [3] have indicated that typical tolerable one-way delay for real-

time interactive applications is in the order of 100 milliseconds. In the case of distributed musical rehearsal, the requirement is even more stringent. Jitter is another issue here. If one of the components that are responsible for audio processing does not have data to process or play out, unpleasant stuttering of the audio, ranging from hardly perceptible to intolerable, occurs. For optimal audio quality, jitter has to be kept at a minimum and one-way latency has to be controlled at about 20 ms [3]. *3) Strict requirement on audio stream synchronization:* Due to the characteristics of the application, multiple audio streams from musicians located at different places have to be synchronized to form consistent music presentation. However, various components such as the clocks of the computers, latencies from the soundcards and their drivers, network interface cards, and network components, rhythm adjustments among different players [2], all raise difficulty for synchronization. This calls for support from both end-systems and networks. Yet, the current source coding mechanisms and the best-effort nature of the Internet poses many challenges on the way to achieve this goal.

We introduce a prototype of a novel *Networked Music Performance* system that enables the music enthusiasts to play with each other through cyberspace. The prototype is implemented on Linux platforms in C++ and QT using RTP for audio streaming with client-server architecture. Currently PCM, DPCM, ADPCM and MP3 are the supported codecs and the work for adding MPEG-4 AAC is ongoing. The streamed contents are mixed and re-encoded at the server to support multi-channel audio. Afterwards they are multicasted to the clients. Two kinds of application scenarios are supported: real-time rehearsal and rehearsal on-demand. Real-time rehearsal (see Figure 1) refers to a scenario that all performances of the participants are generated on-the-fly by real human musicians, with genuine multiparty cooperation nature. On the opposite side is rehearsal on-demand (see Figure 2) in which only the local musician is doing live performance. All the performances of the rest partners are generated from the service provider using pre-recorded contents that emulate the remote musicians. Hence, one can try to play a multi-party concert even if he or she is the only existing musician. The two scenarios compensates for each other in the sense that the former offers better interactivity and flexibility, while the later provides better possibility and quality.

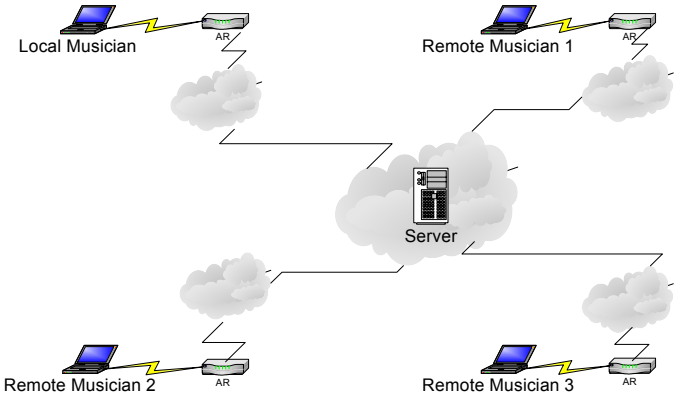

Figure 1. Description of real-time rehearsal

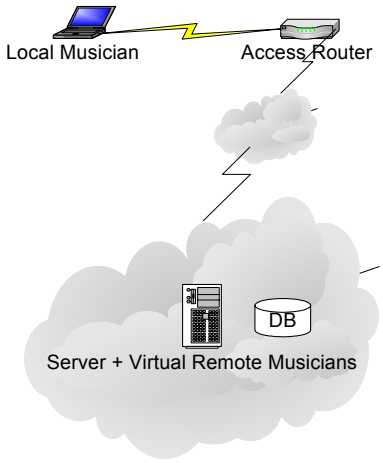

Figure 2. Description of rehearsal on-demand

The remainder of the paper is structured as follows: we review the related work in Section 2. The design and implementation of the NMP system is detailed in Section 3. This is followed by the evaluation of our approach in Section 4. We summarize our experiments and outline the future work in the last section.

II. RELATED WORK

To date, only a few studies devoted to the field of networked music performance have been published. A survey of the literature and online materials has shown that many studies focus on mono audio transport over ATM network, and most of these works involve only two parties. Some of them [4] use MIDI to transport synthetic audio contents that allow putting aside some aspects of the high quality natural audio. Therefore, it is difficult to make a direct analogy with the real-time natural audio streaming. In [5] and [6] the master class approaches are discussed, which have on the one side the instructor and the audience, and on the other side the music performers. It basically lacks of the peer-to-peer and multi-party nature of the networked music performance and there is no need to synchronize the parallel audio streams. Virja [7] is an example which enables distributed MIDI-based collaborative Jazz performance, but it misses a mechanism to synchronize the audio streams and results in intolerable shifting that undermines the auditory consistence. As a step forward, Xu [8] and Cooperstock [9] experimented with PCM audio streaming for real-time performance. A recording studio was used to sample performance given in another country where all the musicians were commonly located. It did not raise any issue of tele-interactions among the musicians. A closer work is the SoundWire Project [10]. Professional level audio was streamed over the Internet2 network to see how musicians could play with latencies introduced in the network. In this experiment, the player could perform with the music pieces but without rhythm in sync at the streaming engine. Perhaps the most comparable approach is the “Conductor Driven Scheme” [2] that allows remote musicians to play together in real-time and cross the Internet. In this work auditory inconsistence was well dealt with where different audio streams were synchronized by the server (the virtual conductor). However, like most of the above approaches,

neither the topic of multi-channel natural audio nor different audio compression schemes was investigated. In addition, none of them provide the possibility of rehearsal on-demand. These aspects are covered in our work.

III. DESIGN AND IMPLEMENTATION

In this section we first describe the overall architecture of the NMP system and justify the reasons for the chosen design. We will then explain the implementation of NMP in more details.

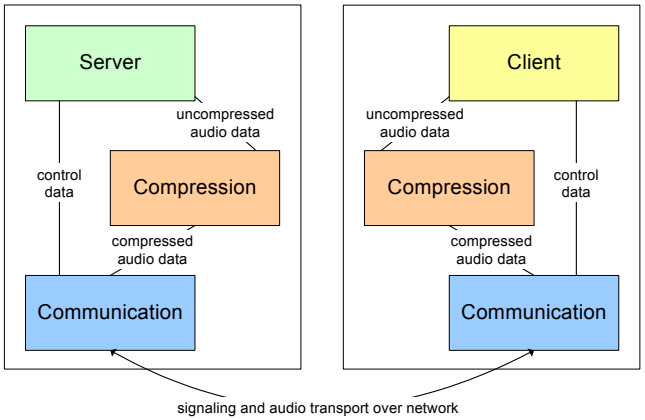

Figure 3. The global architecture

A. System Architecture

As depicted in Figure 3, the NMP System consists of four major components: server, client, communication and compression, as well as the interactions between the individual components.

1) Server

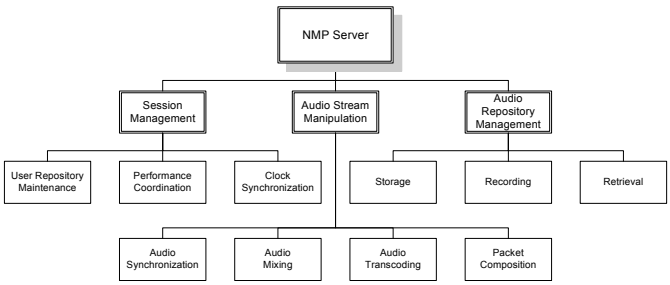

Figure 4. The functional modules of the NMP serve

Due to the multi-party collaborative nature of the NMP system, a point for centralized control is helpful and efficient for such features like session management, audio stream manipulation, and audio repository management. Figure 4 describes the functional modules of the NMP server which is dedicated to this purpose. *Session management* is further divided into several sub-modules. First, *User Repository Maintenance* realizes how the user specific information like user account, user preferences (e.g. the type of music, the style of music, preferred composers etc), user's offers (e.g. music performance pieces, type of instrument or voice part, skill level, relative location in the orchestra etc) are maintained. Second, *Performance Coordination* is where the negotiation for the session set-up (e.g. choose the instrument, partners, music piece, style, relative location in the orchestra, instrument tuning, rhythm control, count-down etc.) is carried out. Third, *Clock Synchronization* is the task to enforce all system clocks of the clients are unified with that of the server, using the well-known Network Time Protocol (NTP) [11]. *Audio stream manipulation* involves audio synchronization, audio mixing, audio transcoding as well as packet composition. The NTP timestamp is the only time base for the whole application. Audio mixing refers to the procedure of multi-channel audio creation. A client solicits its channel association during session set-up phase and the request is confirmed by the server if there is no conflict. Up to six independent channels are supported in NMP. All clients are identified by a unique value. The audio content from those clients using the same channel ID are merged at the server. For the rehearsal on-demand scenario, audio transcoding is performed. The audio content originated by the client is merged with that retrieved by the server from its audio sample repository (corresponding to the remote musicians for the client), and audio compression is applied to minimize the bandwidth consumption while keeping decent audio quality. Packets from different clients bear their own sequence numbers and timestamps. The server uses this information to decide which packets should be used to composite a compound packet with multi-channel audio content (see Figure 5).

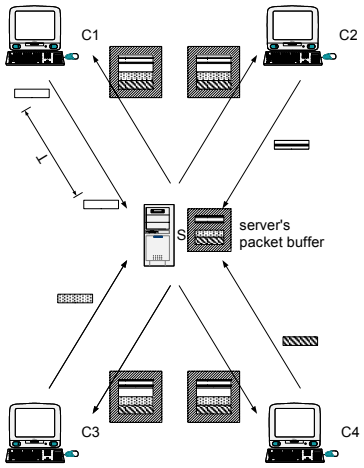

Figure 5. Multi-channel audio composition

For mixing the incoming packets, a simple algorithm is used. All clients use their own random sequence numbers and they are embedded in the packets the clients send using the RTP SN field. When a packet with certain sequence number arrives at the server, the server begins to collect and merge other packets with the corresponding sequence numbers representing the same timestamp. Even if some of the packets get lost with all others being received for a given time point, due to the tight time constrains, the received packets are delivered without waiting for the delayed packets. The synchronization of the clocks is essential for calculating the reference sequence numbers. Fail to do so will cause packet drops at the server. The sequence number space of the clients and server are all independent from each other. The server's sequence numbers provide the master sequence numbers. *Audio Repository Management* fulfills the requirement of recording and storing the performance examples. They can be

used as emulation of remote musicians in the rehearsal on-demand case or for playback of the recorded live performance in the real-time rehearsal case upon requests from the clients.

2) Client

The NMP client provides the users with access to the NMP service. Its architecture is described in Figure 6. The NMP client is equipped with three major functionalities: to enable user interactions with the system for service configuration, to allow the access to sound card hardware, and to synchronize the clock of the end system with that of the server. Part of these functionalities is provided through a graphical user interface (GUI). Service configuration refers to tasks like instrument/voice part selection and tuning, music piece determination, rhythm control, performance starting point signaling, partner selection etc., which are part of the service set-up for NMP. Sound card manipulation represents also a need at the client. Duplex processing is a must as the client has to record the performance of the local musician while playing back that of the remote musicians. Access to the sound hardware allows controlling the related latency that is due to the buffers used by the sound card. Buffers are usually handled differently among various sound card manufacturers or even drivers of different versions of the same sound card. In our design, an audio input/output abstraction layer is used to hide the internals of sound access and complexity of buffer size adjustment for the application. In addition, this simplifies the task of optimizing the setting of sound card buffer size without dealing too much with hardware specific details. Another important issue is the synchronization of the client system clock with that of the server to ensure all the clients use a unified time. Again, NTP is used for this purpose. Under ideal conditions NTP can achieve an accuracy of about 200 μs in most local area networks.

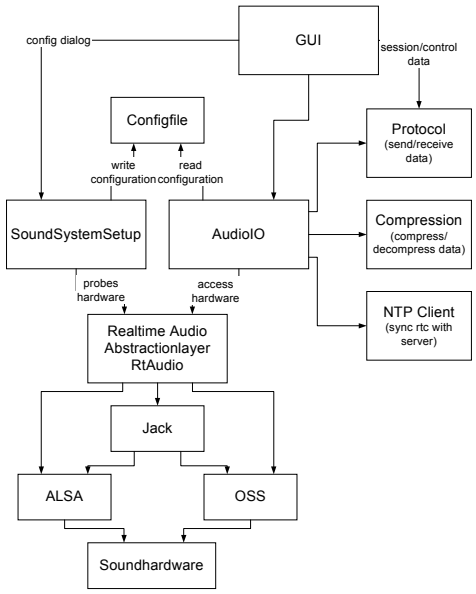

Figure 6. NMP client architecture

3) Communication

The function of the communication component is to provide the mechanisms of actual audio data transport on the one hand, and signaling and session negotiation on the other hand. The corresponding two modules are real-time transport of audio streams enabled by the implementation of RTP [12] over UDP, and session management for negotiation of the service parameters and conveying session info enabled by the implementation of a session management protocol over TCP. The reason we have adopted RTP is that it has been the de-facto standard for years on real-time packet-switched multimedia data transport, and it provides such functionalities like source identification, payload format indication, time stamping, multicast support etc. that are mandatory for our application. Audio transport covers mainly two parts: the unicast transmission of mono audio streams from the clients to the server, and the multicast transmission of the multi-channel audio stream from the server to the clients.

A session management protocol is needed for user group management and service info signaling. This includes tasks like session creation, initialization and update, guiding instrument tuning before actual performance, rhythm indication, remaining time count-down for denotation of the starting-point of the performance (called count-down hereafter) etc. A reliable transport service is needed for the transmission of the signaling messages. We ensure this by using TCP, and on top of that, we designed our own session management protocol called Music Session Protocol (MSP). All message-objects are encapsulated using TCP segments and sent over the link.

4) Compression

High-quality interactive real-time audio streaming is characterized by mass audio data generation and the resulting stress to the network bandwidth. To enable such application for home Internet users who are typically constrained by the last-mile bottleneck link, compression of the audio data is critical whenever bandwidth efficiency matters. At least one additional benefit is that much less data needs to be stored once buffering or disk storage (in the case of rehearsal on-demand) occurs along the path. However, as mentioned earlier, the real-time rehearsal scenario that we concern most is also highly delay-constrained. Our investigations showed that today's mainstream low/middle speed link oriented audio codecs like MP3 or MP4-AAC usually need some greater amount of audio samples to be accumulated before any audio compression related processing can be applied. Filling-in the input buffer of the audio codec indicates a non-negligible start-up compression delay. To complicate this issue further, the computer's sound card commonly adopts also a buffer that must be filled before the next processing unit (e.g. audio compression) is served. Thanks to Moore's Law the computing overhead of audio compression is a secondary issue on a state-of-the-art machine. Hence, we have a tradeoff between service latency and bandwidth efficiency, and the usefulness of NMP in the sense of the latencies that are introduced is somehow proportional to the codec's buffer size. We performed further investigations to choose an appropriate audio codec for the real-time rehearsal scenario and fine-tune the operational parameters. We selected an Adaptive Differential Pulse-Code Modulation (ADPCM) codec, which requires no buffering and has very little computational overhead. Extremely low latency in the end-system is

achieved where only a few audio samples are collected and packed in a single packet after compression, which is then sent to the network. The compression gain that can be achieved by ADPCM (a fixed compression ratio of 4:1) is significantly below that of MP3 or MP4-AAC (compression ratio up to 12:1 with CD quality audio). However it is the delay that dominates the usefulness of NMP, and thus we decided to trade bandwidth efficiency for latency. Due to their compression gain and bandwidth efficiency, MP3 and MP4-AAC are used for the rehearsal on-demand scenario. Since start-up service latency is relatively relaxed in this case, the buffering mechanism provided by the client software should be able to mask off the latencies involved in the data compression and decompression. This is also much more economical as the server usually has to store mass amount of pre-encoded performance samples for certain instruments or voice parts. Also for the network, bandwidth resource is significantly saved when compressed audio content is streamed over the link. The compression component is designed as a library providing a common interface, with the consideration that changing the codec should not affect the rest part of the system, and the software should be easy to upgrade. The actual implementation of the compression scheme and the choice of the codec are up to the application. The interface is used to define the data types and operations to be provided by this component, and how the compression service can be accessed. The workflow of audio compression is depicted in Figure 7.

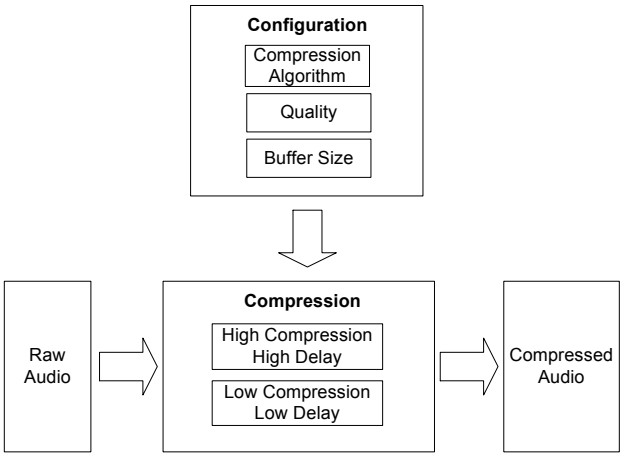

Figure 7. Workflow of audio compression

B. Implementation

The first consideration regarding implementation is choosing the right operating system. The decision was in favor of Linux platform due to several reasons. First, Linux offers two different sound driver architectures and hence flexibility. The older and simpler one is the Open Sound System (OSS) [13]. On the opposite side is the Advanced Linux Sound Architecture (ALSA) [14] with more advanced features and which is usually bundled with the latest Linux distributions. Second, all sound card drivers are open source, which makes it much easier to manipulate the sound card directly, as compared to other commercial platforms or less user-friendly free platforms. The implementation has been done in C++.

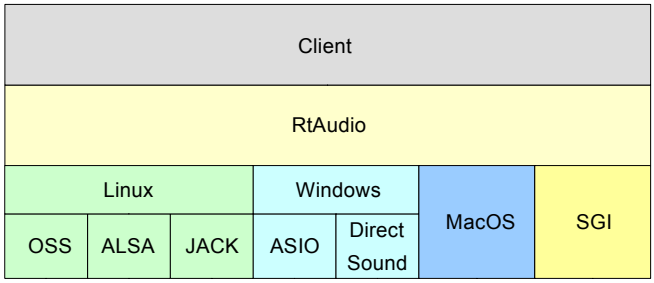

Figure 8. Client audio IO

We use RtAudio (see Figure 8) as the abstraction layer for access to the audio hardware. RtAudio [15] can easily handle buffer management and also provides an abstraction to the underlying drivers. Also portability can be ensured by using a common wrapper for buffer configuration to achieve the lowest possible latency. Common C++ libraries are used to provide portability. This includes reusing the ccRTP library with customization and optimization for audio transport, and the highly evolved MP3 encoder LAME for audio compression. Also the client GUI is written with the cross-platform QT-library by Trolltech [16]. Figure 9 shows the main window of the application and Figure 10 shows an example configuration dialogue.

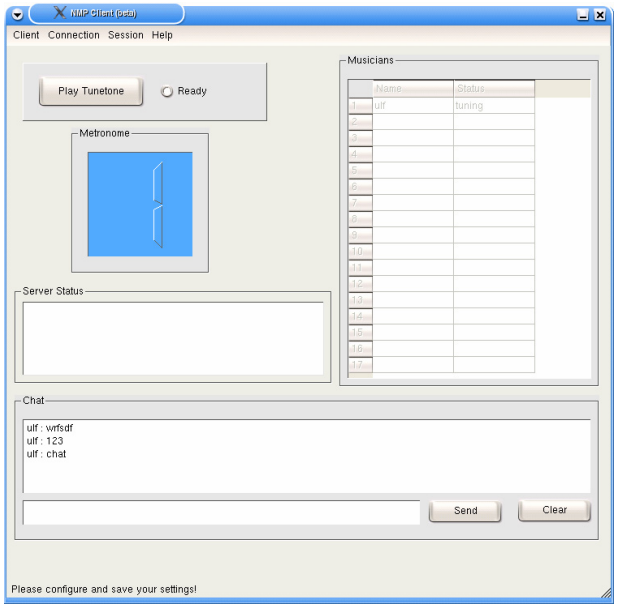

Figure 9. Main window of NMP client

The hardware used for the implementation and evaluation consists of: Pentium IV 3 GHz, 512 MB RAM Debian Sarge Linux PC with kernel 2.6.5/2.6.6 and other lower profile machine as clients, dual Xeon 3 GHz, 2 GB RAM SuSE 9.1 Linux Server with 2.6.4 kernel as server, interconnection via fast Ethernet switches, Creative Audigy 2 sound cards, Creative 7.1 channel speaker systems, Sony and Philips professional-class microphones, Tektronix TDS 220 oscilloscope and XC6 Sweep Generator for latency measurement.

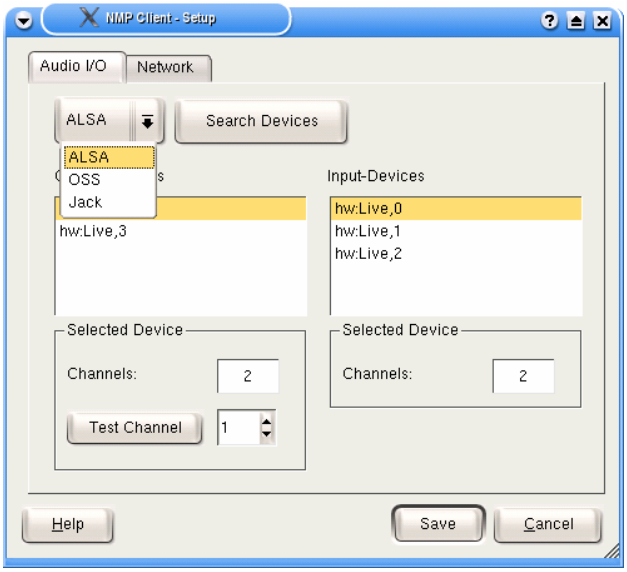

Figure 10. Client audio property configuration

IV. EXPERIMENTAL EVALUATION

The goal of the evaluation was to understand the impact on the audio quality by several factors introduced by the application itself. Further we wanted to learn about the subjective acceptance of the application. Since our tests were performed in a local area network, which satisfies the bandwidth requirements and had a relatively low transmission delay (<< 1 ms), we do not consider the network impact such as network transmission delay, delay jitter or bandwidth fluctuations at this stage. The factors selected for the analysis include: distortion introduced by audio compression scheme, distortion introduced by mixing the streams on the server, single-hop latency and end-to-end delay. All audio quality tests were performed within a controlled test environment and, therefore, unexposed to realistic network conditions that could cause additional distortions.

A. Test Procedure

To evaluate the audio codec distortion we used two coding schemes: Adaptive Differential Pulse-Code Modulation (ADPCM) and MPEG-1 layer 3 (MP3). The measurement metrics utilized were Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS). PSNR is an objective non-human based value derived from Mean Square Error (MSE) which represents the noise between the original and the compressed signals [17][18][19]. The output of the PSNR measurement is expressed in decibels (dB). Typical PSNR values are 50 dB for AM radio quality, 70 dB for tape and FM radio quality, and 90 dB for CD quality [20]. MOS [21] is a common metric used for subjective quality measurements. It is human based and can take values in range from 1 to 5. A MOS value of 1 means that the observed quality is unacceptable; while a MOS value of 5 represents outstanding quality level or no noticeable difference compared to the original content. The guidelines to the test persons concerning the MOS values are described in TABLE I. It should be noted that the objective measurements can be used to validate the audio quality based on industry standards, although the subjective quality still has the most important impact on the acceptance of the deployed codec. For the end-to-end delay test we have used a real-time two channel oscilloscope Tektronix TDS 220 and a Wavetek sweep generator in order to achieve accurate measurement results.

B. Test Sequences

Ten different audio sequences with various characteristics and durations were used as sources for the audio quality tests. These publicly available sequences have been selected from three different sources (see TABLE II): Tektronix, MIT and the LAME project.

TABLE I. MOS SCALE

Score	Direction
5	The sample sounds exactly like the original.
4	Some distortion is audible but the quality is still very good.
3	The quality is still acceptable but there is a clear loss of detail and audible distortion. It sounds like FM radio.
2	The sample sounds like AM radio. Music quality is barely acceptable.
1	Neither music nor voice can be reproduced. Listening to such a low quality is simply undesirable.

TABLE II. AUDIO TEST SEQUENCES

Sequence	Characteristics	Source	URL
400	400 Hz wave signal	Tektronix	ftp://ftp.tek.com/tv/test/streams/Elemement/MPEG-Audio/400.wav
1kz	1 kHz wave signal	Tektronix	ftp://ftp.tek.com/tv/test/streams/Element/MPEG-Audio/1kz.wav
Bass	Chorus, bass voice	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
Castanets	A classic "pre-echo" test case	LAME	http://lame.sourceforge.net/download/samples/castanets.wav
Harp	Harpsichord	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
Horn	Horn	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
Quar	Chorus	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
Spfe	Trumpet	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
Vioo	Violin	MIT	http://sound.media.mit.edu/mpeg4/audio/sqam/
youcandothat	Guitar	LAME	http://lame.sourceforge.net/download/samples/youcantdothat.wav

C. Test Configurations

Four test configurations A-D were used for audio quality and end-to-end transmission delay measurements. First we investigated the distortion introduced by the compression scheme. This was accomplished with test configuration A (see Figure 11), where two different codecs, namely ADPCM and MP3, were used for compression of 10 test samples. The compression ratios for the ADPCM and MP3 codecs were set to 4 and 12 respectively. The quality value for the LAME encoder has been set to 5 (mid-level). The original test samples were raw audio. We implemented our own method for

calculating the PSNR values of each codec. The subjective quality measurements were performed by 15 test persons in one room with relatively low background noise level.

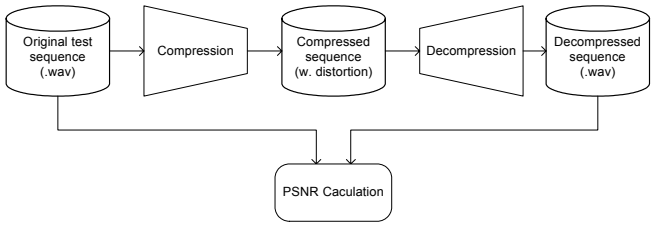

Figure 11. Test configuration A - distortion due to compression

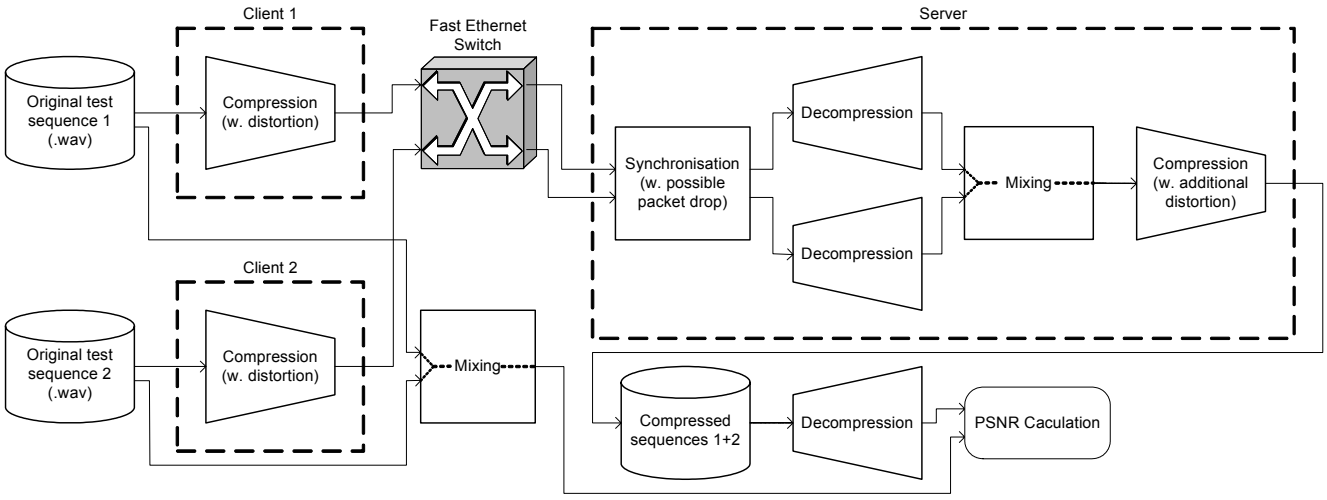

Figure 12. Test configuration B - end-to-end processing distortion

Test configuration B was built to evaluate the audio stream distortion introduced by the processing at both the clients and the server. This configuration includes two client and one server machines, which were connected together via a Fast Ethernet switch (see Figure 12). The settings of the codecs as well as the test room remain the same as what were used in test configuration A.

A single client processing delay has been measured using the test configuration C. We have used a Wavetek sweep generator to form an input signal for the client machine. The single hop delay consists of four main parts: audio capture and playback latencies, compression processing time and decompression processing time. We have connected the output from the sweep generator as well as the output of the client's sound card to the oscilloscope in order to measure the time difference which represents the whole client's processing delay (see Figure 13). We have performed the latency measurements only for the ADPCM compression scheme, because only this codec has been used for the real-time scenario. The settings of the ADPCM codec remained the same as used in the previous test configurations. The input buffer size of the audio hardware has been set as low as 256 bytes (the minimum possible setting).

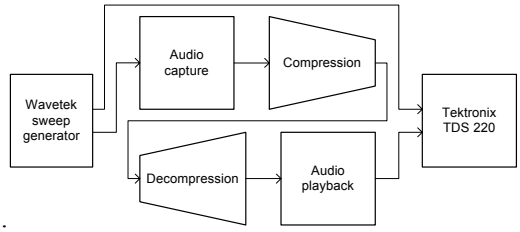

Figure 13. Test configuration C – single-hop latency measurement

In order to measure the end-to-end delay between two clients we have implemented the test configuration D. This configuration includes two clients and one server, which were connected together via a Fast Ethernet switch (see Figure 14). The end-to-end delay comprises four major components: overall client processing time, two-way network transmission delay, server synchronization delay, as well as server decompression, mixing and compression processing time. Since the network transmission delay in our test scenario was much less than 1 ms, it was neglected in our evaluation. All settings remained the same as what were used in the test configuration C.

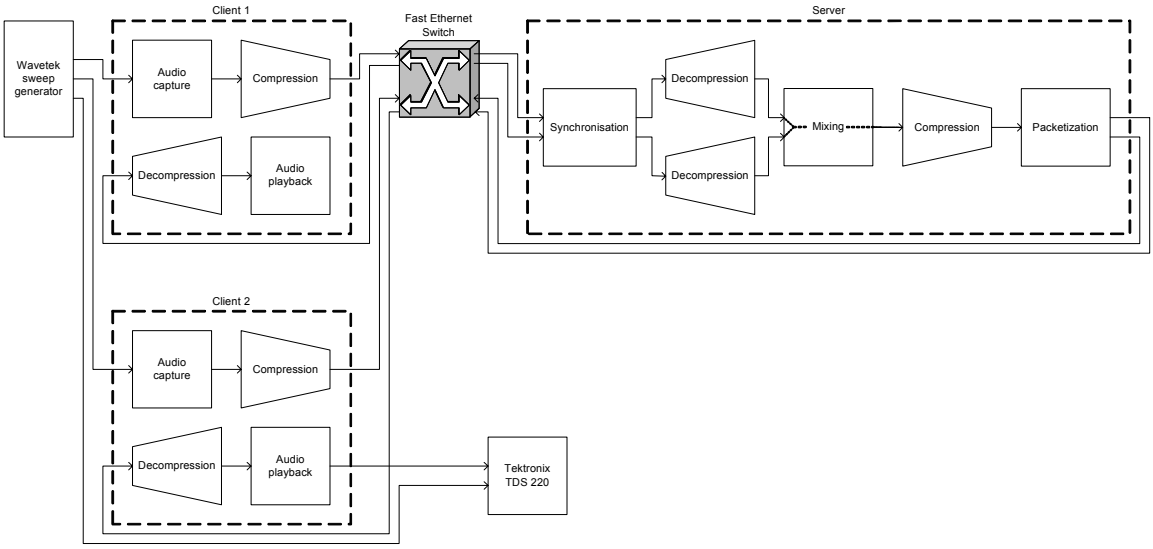

Figure 14. Test configuration D – end-to-end latency measurement

D. Analysis of the Results

1) Test Configuration A

The PSNR and MOS values of the ADPCM codec measured for the ten audio sequences in test configuration A ranged from 41.58 dB to 71.48 dB and from 2.36 to 5.00 respectively, as shown in Figure 15 and Figure 16. The MOS output for eight of the ten audio sequences exceeded a value of 3. The measured PSNR values for most sequences in this test configuration showed different behavior compared to the corresponding MOS values. In other words, PSNR and MOS values were only weakly correlated. For example, the PSNR value of 66.44 for the sequence *trpt* was the second best in our test for ADPCM, while its corresponding MOS value of 3.27 ranked at 7 in all 10 test sequences. For the ten audio sequences, the average MOS value was 3.87 and the corresponding PSNR was 57.16 dB.

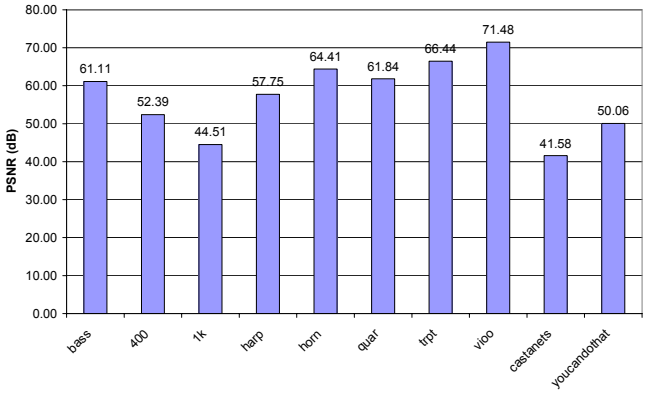

Figure 15. ADPCM objective test results for configuration A

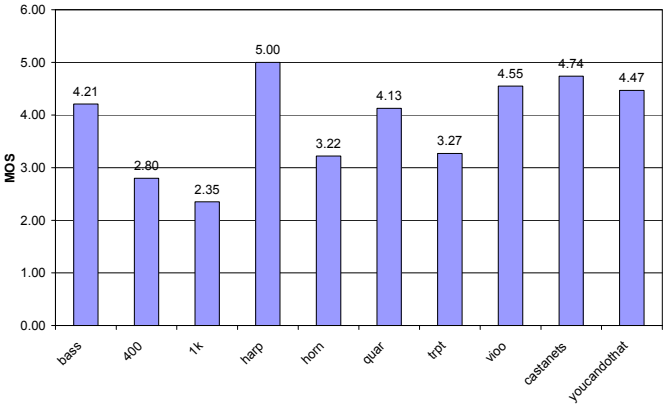

Figure 16. ADPCM subjective test results for configuration A

Figure 17 and Figure 18 give the acquired audio quality measurements for the MP3 codec in test configuration A. The average PSNR value was 44.84 dB which was less than that for the ADPCM; however, the MOS averaged 4.84, which was almost 1.0 greater than that for the ADPCM. The better MOS values of the MP3 compression scheme could be due to the psycho acoustic model used in the compression algorithm.

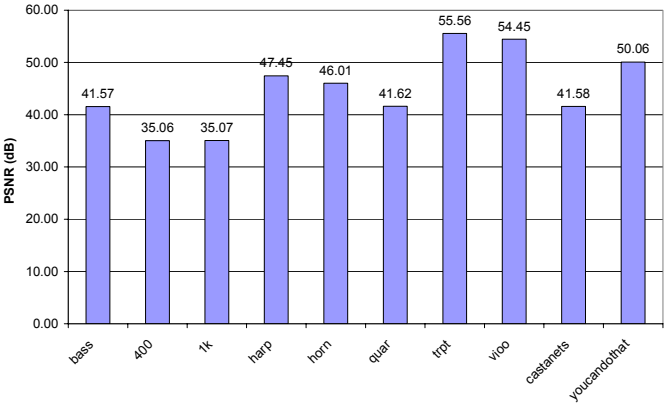

Figure 17. MP3 objective test results for configuration A

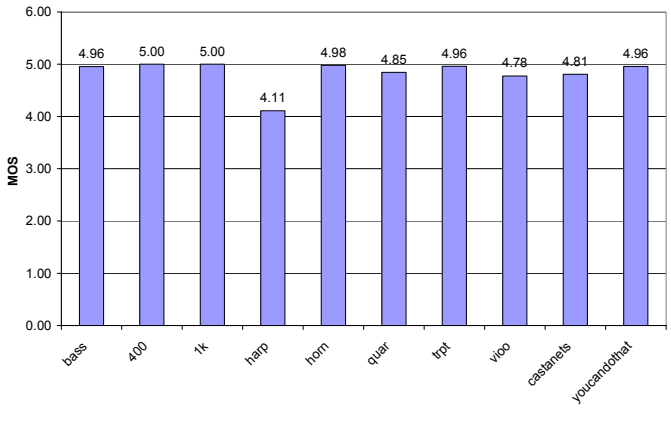

Figure 18. MP3 subjective test results for configuration A

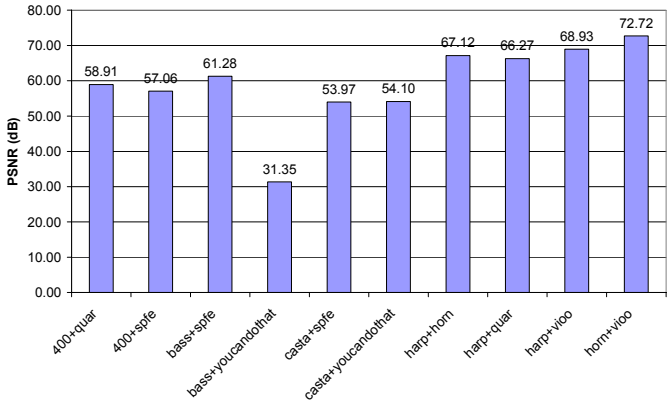

Figure 19. ADPCM objective test results for configuration B

2) Test Configuration B

The pairs of audio sequences were transmitted using the test configuration B with ADPCM codec with PSNR measurements ranging from 31.35 dB to 72.72 dB and MOS values from 2.95 to 4.08, which are presented in Figure 19 and Figure 20 respectively. The average MOS and PSNR values in this configuration were 3.55 and 59.17 dB respectively. The measured PSNR values for the MP3 codec varied between 31.36 dB and 76.59 dB as shown in Figure 21. The MP3 MOS values were between 3.26 and 4.95 (see Figure 22). Again, no significant correlation between the PSNR and MOS values has been observed. The average MOS for the MP3 codec was as high as 4.34 (0.7 better than ADPCM).

3) Test Configuration C

The delay measurements regarding to the test configuration C have shown that our prototype was able to provide a single hop processing delay as low as 10.5 ms. The audio capturing and playback consumed each 2.67 ms. The rest part of 5.17 ms belongs to compression, decompression as well as other end-system overhead.

4) Test Configuration D

The average end-to-end delay for test configuration D achieved a value of 22.5 ms, although the delay jitter reached 10 ms in a few test cases. We suppose that this relatively high value for delay jitter might come from the process scheduler of the operating system. This will be investigated and validated in the future.

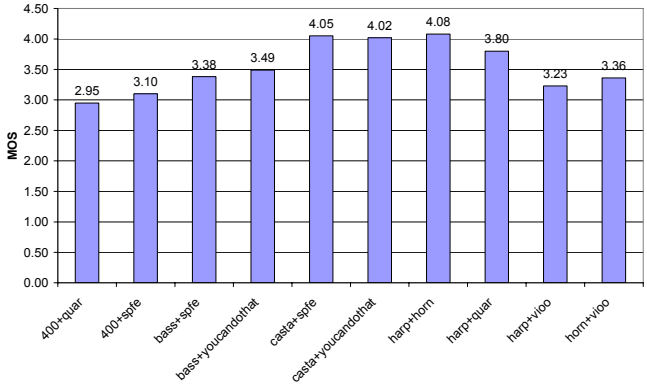

Figure 20. ADPCM subjective test results for configuration B

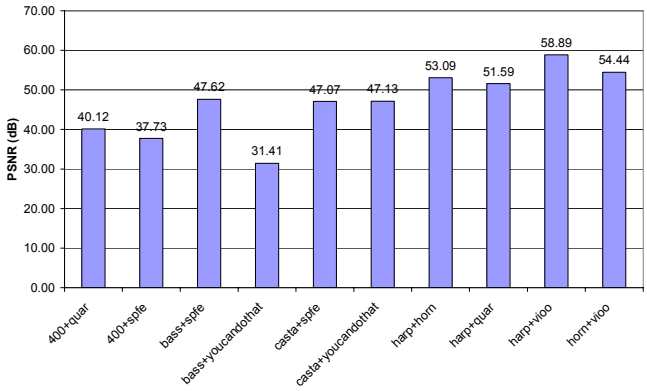

Figure 21. MP3 objective test results for configuration B

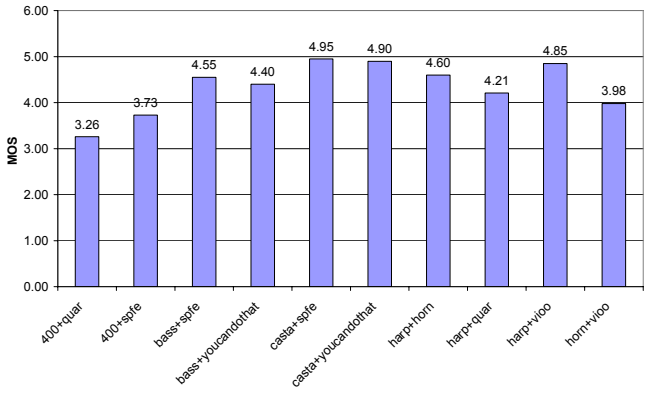

Figure 22. MP3 subjective test results for configuration B

5) Summary

The evaluation results have shown that our prototype ensures the strict real-time demands of audio data transmission for networked music performance while providing acceptable audio quality. According to [22], the upper bound for the affordable skew for interactive audio applications is 120 ms. Our prototype has achieved far better results (e.g. about 100 ms less latency), and makes it promising to extend our application to larger-scale networks.

V. CONCLUSIONS AND FUTURE WORK

We proposed a new application of networked music performance. We designed and implemented a prototype of the system using a test-bed in a LAN that suffices the real-time constrains and the required audio quality for interactive multi-channel audio delivery. In the future, we will extend the scale of the application to networks spanning across larger physical distances and supporting more spontaneous users. We also plan to add the support of MPEG-4 AAC codec to further improve audio quality at the same bandwidth constrains, and to handle multi-channel real-time audio streaming better. Realistic network conditions outside the LAN will be considered in the next step to investigate the performance of the application in larger networks. Then also delay jitter, its impact and counter-measure approaches will be more intensively studied. A more sophisticated evaluation model will also be developed to compare different approaches with higher fidelity. In addition, regression analysis of the collected data should be performed to provide more detailed results.

ACKNOWLEDGMENT

This work benefited from the discussions with, and implementation/technical support from, these people at TU-Braunschweig: D. Brökelmann, Z. Kurtisi, J.-P. Hoeft, S. Thelemann, M. Karray, D. Schilgen, H. Longwitz, O. Meynberg, J.-H. Hanne, M.-C. Sinnreich, S. Olinski, V. Schomerus, S. Schulze, P. Hasse, A. Frambach, C. Pinkernell, and G. Gunkel.

REFERENCES

[1] William T.C. Kramer, "SCinet: testbed for high-performance networked applications", IEEE Computer, pp. 47-55, June 2002

[2] Nicolas Bouillot, "The auditory consistency in distributed music performance: a conductor based synchronization", ISDM (Info et com Sciences for Decision Making vol. 13(0), pp. 129-137, March 2004.

[3] Chris Chafe, Michael Gurevich, Grace Leslie and Sean Tyan, "Effect of time delay on ensemble accuracy", In Proc. of the International Symposium on Musical Acoustics, (ISMA2004), Nara, Japan, March-April 2004.

[4] Fernando Lindner Ramos, Marcio de Oliveira Costa and Jônatas Manzolli, "Virtual studio: distributed musical instruments on the web", in Proc. of IX Brazilian Symposium on Computer Music, Campinas, Brazil, August 2003.

[5] Dimitri Konstantas, "Overview of a telepresence environment for distributed musical rehearsals", in Proc. of ACM Symposium on Applied Computing (SAC'98), pp. 456-457, Atlanta, US, February 98.

[6] Young J.P. and Fujinaga I., "Piano master classes via the Internet", in proc. of the International Computer Music Conference (ICMC'99), pp. 135-137, Beijing China, October 1999.

[7] Goto M., Hidaka I., Matsumoto H., Kuroda Y. and Muraoka Y, "A jazz session system for interplay among all players", IPSJ Journal, vol. 43, pp. 299-309, 2002.

[8] Xu, A. et al., "Real Time Streaming of Multi-channel Audio Data over Internet", Journal of the Audio Engineering Society, vol. 48, number 7-8, July-August 2000.

[9] Cooperstock J. and Spackman S., "The recording studio that spanned a continent", in proc. of IEEE International Conference on Web Delivering of Music, WEDELMUSIC01, Florence, Italy, November 2001.

[10] Chafe C., Wilson S., Leistikov R., Chisholm D., Scavone G., "A simplified approach to high quality music and sound over IP", in Proc. of COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 2000.

[11] David L. Mills, "Network time protocol (ver. 3), specification, implementation and analysis", RFC 1305, March 1992.

[12] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: a transport protocol for teal-time applications", RFC3550, July 2003.

[13] 4 Front Technologies, "OSS programmer's guide", ver. 1.11, November 2000, URL: http://www.opensound.com/pguide/oss.pdf.

[14] The advanced Linux sound architecture (ALSA) project, URL: http://www.alsa-project.org.

[15] Gary P. Scavone, "The RtAudio tutorial", URL: http://www.music.mcgill.ca/~gary/rtaudio/

[16] Trolltech, "QT 3.3 whitepaper", URL: http://www.trolltech.com/pdf/whitepapers/qt33-whitepaper-a4.pdf.

[17] M. Reisslein, J. Lassetter, S. Ratnam, O. Lotfallah, F. H. P. Fitzek, and S. Panchanathan, "Traffic and quality characterization of scalable encoded video: a large-scale trace-based study, part 1: overview and definitions", Arizona state university, dept. of electrical engineering, technical report, December 2003.

[18] ANSI T1.TR.74-2001, "Objective video quality measurement using a Peak-Signal-to-Noise-Ratio (PSNR) full reference technique", American National Standards Institute, Technical Report, 2001.

[19] ANSI T1.801.03-1996, "Digital transport of one-way video signals-parameters for objective performance assessment", American National Standards Institute, 1996.

[20] A. J. Aude, "Audio quality measurement primer", Intersil Corp., Application Note AN9789, February 1998.

[21] International Telecommunication Union, "Subjective video quality assessment methods for multimedia applications," Rec. ITU-T P.910, September 1999.

[22] R. Steinmetz, "Human perception of jitter and media synchronization," in IEEE Journal on Selected Areas in Communications, vol. 14, issue 1, pp. 61 - 72, January 1996.