Network-centric Music Performance: Practice and Experiments

Xiaoyuan Gu, Matthias Dick, Zefir Kurtisi, Ulf Noyer, and Lars Wolf, Technische Universität Braunschweig

ABSTRACT

Advances in information technology and the great proliferation of the Internet have changed nearly every aspect of the work and life of human beings. Despite progress in networked entertainment, many music professionals and enthusiasts are still sticking 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 a novel system called network-centric music performance (NMP) that enables multiparty music performance through cyberspace. Our target is to support real-time multichannel natural audio streaming over the network, using audio compression schemes that can provide acceptable audio quality. A system like this is bandwidth-demanding and highly delay-sensitive, and requires synchronization of the audio streams. Hence, support from the underlying end systems and networks is critical. However, the current source coding mechanisms and the best effort nature of the Internet pose many challenges to achieve the desired quality of service. We have implemented a prototype of NMP, and exploited end system and network influences on NMP. The work was done in a LAN environment using Linux PCs. The system enables two different application scenarios: real-time rehearsal and rehearsal on demand. Real-time multichannel audio transport and different audio compression schemes are supported. Our evaluation results based on both subjective and objective measurements show that the system provides sufficient audio quality level for the target application in such an environment. The scalability test also revealed that the system scales well with increase of clientele. In the future we will extend our system for networks spanning larger distances and experiment with more realistic network conditions in the Internet.

INTRODUCTION

With the great proliferation of the Internet and widespread availability of broadband, the market of networked entertainment is growing. Networked music, exemplified by Internet radio stations and online music stores, has been well established and exploited. Besides the usage of the Internet for music databases, research communities have seen in recent years interest in exploring the unique, complex, multidimensional 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.

Despite wide utilization of the Internet, for decades many music professionals, especially those who are engaged in the performance of classical music, have stuck to the traditional way of carrying out rehearsals and concerts. Music performance in this way requires the physical presence of the musicians and has a number of inherent limitations like fixing a common time and place, finding players of the desired skill level, and synchronization of sheet music. Hence, there is a basic need to improve the way music enthusiasts and professionals alike perform for the sake of flexibility, economy, efficiency, productivity, and creativity.

Network-centric music performance (NMP) represents such a concept whereby 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 that occur in traditional music performance, NMP is a challenging application where a number of factors complicate this task.

Bandwidth-demanding: As reported in [1], real-time audio streaming-based teleportation (the category to which NMP belongs) is one of the most bandwidth-intensive applications in today's networks. Transmission of mono pulse code modulation (PCM) (raw) CD-quality audio requires a data rate of about 0.7 Mb/s. When stereo/multichannel or high-definition sound (high sampling rate, e.g., 48/96/192 kHz, or better quantized, e.g., using 24 bits) is needed, the network is further stressed (up to 27.6 Mb/s for six channels). Hence, audio compression is needed for efficient usage of the available network bandwidth.

Highly delay-sensitive: Due to the fact that human hearing is very sensitive to delayed or missing data in music, especially that played on fine acoustic instruments, the prebuffering mechanism common in most Internet music sys-

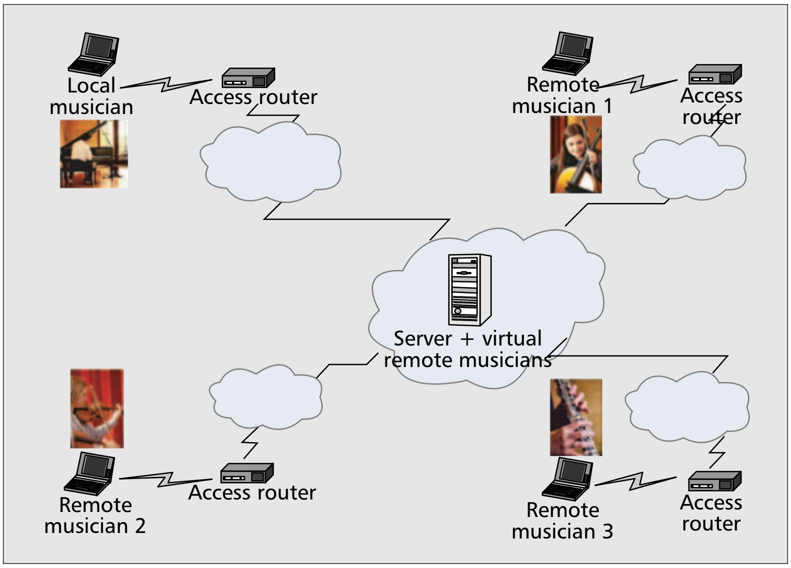
0163-6804/05/$20.00 © 2005 IEEE

tems today is not applicable when contents are generated on the fly and intensive interactivity is a must. Research results [2] have indicated that typical tolerable one-way delay for real-time interactive applications is in the order of 100 ms. In the case of distributed musical rehearsal, the requirement is even more stringent. Another important issue is delay jitter. If one of the components responsible for audio processing does not receive the data to process or play out in time, unpleasant stuttering of the audio can lead to drastic service quality degradation.

Strict requirement on audio stream synchronization: Multiple audio streams from musicians located at different places have to be synchronized within certain time intervals. However, various components such as system clocks of computers, audio hardware latencies, network components, as well as rhythm adjustments among different players make synchronization a challenging task. Both network support and application adaptation are needed in order to cope with these issues.

NMP enables music enthusiasts to play with each other through cyberspace. Two kinds of application scenarios are supported: real-time rehearsal and rehearsal on demand (Fig. 1). Real-time rehearsal refers to a scenario where all performances of the participants are generated on the fly by live musicians, with genuine multiparty cooperation. The other scenario is rehearsal on demand in which at least one of the performers is not a live musician. The performance is actually generated from the server at the service provider side using prerecorded contents that emulate a remote musician. Hence, in an extreme case, one can try to play a multiparty concert even if he or she is the only live musician. Besides self-entertainment, rehearsal on demand can solve the problem when a certain musician is missing or too far away to play at a concert. In addition, it helps in providing some contents that could be unavailable in a certain real-time rehearsal session.

It should be noted that like any system, NMP has its theoretic application boundary. Due to physical limits (signal propagation speed of about 2/3 lightspeed), according to the characteristics and constraints mentioned earlier (maximum 100 ms delay), NMP is not applicable to a scenario in which the maximum data propagation path between one client and another via the server is more than 20,000 km. In reality, however, delays at end systems and networks are hard to avoid, which tightens the application boundary of NMP. Within this scope, the second issue is QoS provisioning. Without network support for bandwidth prediction and reservation using some approach like integrated services (IntServ) or a variant, NMP over the current Internet will be harsh for sound maniacs. Therefore, we assume that given a high quality of service, our application could be employed within a medium-sized country. As an example, the end-to-end transport delay over the Internet between two cities in a medium-sized country like Germany (e.g., between Braunschweig and Karlsruhe, physical distance 470 km, 12 hops) has been measured as low as 11 ms. However, we would like to point out that significant delays occur even in real-life concerts. For example, on a large stage at a symphony concert, diagonally located players on the edges can have a distance of 30 m, and about 100 ms for propagation of sound is needed. Hence, the conductor usually serves as the synchronization point. This problem can be alleviated in some cases if microphones and monitor speakers are used (e.g., in a rock concert). Moreover, we believe that a differentiation between professionals and amateur players must be made. For the former, usually studio-quality fans, a stringent limit on the boundary of real-time rehearsal is important, and rehearsal on demand will also help. However, it should be mentioned that we do not target NMP as a studio replacement. Rather, NMP can help make studio music rehearsal flexible and easier. For some amateur players who are unable to follow the exact beats even sitting in the same living room, the conditions for real-time rehearsal can be more relaxed.



■ **Figure 1.** *Scenarios of network-centric music performance.*

The rest of the article is structured as follows. We review the related work. The design and implementation of the NMP system is detailed, followed by an evaluation of our approach. We then summarize our experiments and outline future work.

Related Work

To date, only a few studies devoted to the field of network-centric music performance have been published. A survey on literature and online materials has shown that many studies have been focused on mono audio transport over asynchronous transfer mode (ATM) networks, and most of these approaches involve only two parties. Some of them [3] use MIDI to transport synthetic audio contents that allow putting aside some aspects of high-quality natural audio. Therefore, it is difficult to make a direct analogy with real-time natural audio streaming. In [4, 5] master class approaches are discussed, which have on one side the instructor and the audience, and on the other side the music perform-

Audio transport is performed in two ways: performance of each client is transmitted using unicast to the server; the processed audio of all participants is then delivered using multicast (if applicable, otherwise unicast is used) from the server back to all clients.

ers. It basically lacks the multiparty nature of NMP, and there is no need to synchronize parallel audio streams. Virja [6] is an example that enables distributed MIDI-based collaborative jazz performance, but there is no mechanism to synchronize the audio streams, and it results in intolerable shifting that undermines auditory consistency. As a step forward, Xu [7] and Cooperstock [8] experimented with PCM audio streaming for real-time performance. A recording studio was used to sample a performance given in another country where all the musicians were commonly located. It did not raise any issue of tele-interactions among the musicians. During the September 2004 Internet2 Member Meeting, the HYDRA Project unveiled its research results in the Miró Quartet: Live & Virtual Gala Event (http://dmrl.usc.edu/hydra.html). Capable of delivering HDTV-level video streams and 10.2-channel immersive audio streams, the HYDRA has, however, limited itself to uncompressed linear high-fidelity PCM audio. Also, all musicians were located at the same site; the live performance was captured and delivered over the Internet2, and then played back for an audience at another site. Again, multiplayer collaborative performance via a network was not realized. Closer is the SoundWire Project [9]. 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 players 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" [10] that allows remote musicians to play together in real time and across the Internet. In this work, auditory inconsistency was dealt with well, with different audio streams synchronized by the server (the virtual conductor). However, like most of the above approaches, neither multichannel natural audio nor different audio compression schemes were investigated. In addition, none of them provide the possibility of rehearsal on demand, and there is no scheme designed with home Internet users and the mass commercial market in mind. These aspects are covered in our work.

Design and Implementation

In this section we first describe the overall architecture of the NMP system and then explain the individual functional modules in greater detail.

System Architecture

The NMP system consists of two major components: the server as the centralized management unit, and the client that provides a musician with access to the system. Because of their presence at both the client and server, we discuss communication, clock synchronization, and audio coding as separate modules.

Although at this stage of the project we have adopted a client-sever architecture, we do not preclude the possibility to support a peer-to-peer (P2P) architecture in the future. The reason the former was chosen is that we think a point for centralized control is necessary for tasks like session management and synchronization, real-time audio content processing (e.g., mixing, transcoding even with hardware), and so on. These are computation-intensive operations that are not appropriate for low-profile client machines. In addition, as mentioned earlier, NMP is designed for home Internet users, to whom bandwidth efficiency usually matters. One of the advantages of a server in place is that a client will not necessarily deliver the same audio content redundantly to each of the other participants. This is mandatory, however, in the case of P2P, which demonstrates symmetric traffic volumes between uplink and downlink. On the contrary, using an NMP server saves the uplink bandwidth for the client on one hand, and squeezes the downlink bandwidth through compression on the other hand. In this way, bandwidth efficiency can be much better exploited to reduce overall network traffic. This not only benefits system scalability, but is also more in line with most Internet applications that are asymmetric by nature. Besides, audio repository management, the key to rehearsal on demand, requires mass storage capacity for recorded audio sequences, which can be challenging for most clients. Also, the audio sequences are supposed to be maximally reused/shared in the commercial rehearsal on demand service for any possible user, rather than being proprietary for only a small group of people. However, P2P might be taken into account with advances of hardware technologies, and if bandwidth efficiency is not a concern and no compression is needed, or if real-time rehearsal is the only thing users care about. Last but not least, grid computing can also play a role, in particular to share the workload of the NMP server for a system with a large clientele.

Communication — The communication component provides transport of audio data implemented by Real-Time Transfer Protocol (RTP) over User Datagram Protocol (UDP), and signaling and session negotiation using a session management protocol over TCP. We have adopted RTP as it is a widely accepted protocol for real-time transport of packet-switched multimedia data. Furthermore, it provides all mandatory functionalities needed for our implementation. Audio transport is performed in two ways: performance of each client is transmitted using unicast to the server; the processed audio of all participants is then delivered using multicast (if applicable; otherwise, unicast is used) from the server back to all clients.

Session management is implemented by our Music Session Protocol (MSP). MSP includes tasks like session creation, initialization and update, guiding instrument tuning before the actual performance, rhythm indication, and remaining time countdown for denotation of the starting point of the performance (called countdown hereafter). We use TCP as a reliable underlying transport service for session data transport.

Two types of streaming services are supported in NMP: real-time audio streaming and audio on demand. In both cases, audio data transport has bounded delay, while the former, used for real-time rehearsal has more stringent boundaries. The latter, which suits rehearsal on demand, is pseudo real-time in the sense that audio data is recorded in advance at the server, and delivered to the client upon request. It usually adopts a client-side buffer to smooth out the dynamics in source coding and network transmission, but this introduces startup service delay as well. Due to the tight delay budget for real-time rehearsal, error-resilient encoding schemes like forward error correction (FEC) and multiple description encoding (MDC) should be used with care, as the resilience comes at a price for additional processing overhead and a sacrifice of network bandwidth.

Clock Synchronization — All clients of a session have to be synchronized for a unitary beat and precise timestamps of the audio packets. This is accomplished through the Network Time Protocol (NTP) with initial synchronization during session setup and periodical refreshments during the session. The accuracy of this NTP-based synchronization is about 200 µs in local area networks under optimal conditions. We are aware of considerable clock skews while using NTP for intercontinental communication. However, in metropolitan area networks or WANs within small countries the inaccuracy of NTP is within a few milliseconds, which is still sufficient for our application.

Audio Coding — High-quality interactive real-time audio streaming is characterized by mass audio data production that is in our case multiplied by the number of sessions and their sizes. For rehearsal on demand, all source audio sequences are stored at the server, and audio compression is critical. However, for real-time rehearsal delay has higher priority. Our investigations showed that today's mainstream audio codecs for low-/medium-speed links like MP3 or MPEG-4 AAC require substantial input buffering that results in nonnegligible startup compression delay. For example, even the low-delay profile of MPEG-4 AAC has about 20 ms for such delay. Also, delay is introduced by the input/output buffering at the sound card. Queuing, transmission, and propagation in the network contribute additional delay too. Hence, we have a trade-off between latency and bandwidth efficiency. The usefulness of NMP in the sense of service latency is proportional to the total buffer size. We performed further investigations to choose an appropriate audio codec for the real-time rehearsal scenario and fine-tune the operational parameters. To achieve the lowest possible latency at the end system, an appropriate codec must be able to handle the same audio block size used by the audio device so that audio data is compressed and delivered right after being processed by the hardware. On the other hand, real-time audio compression schemes like G.711 and G.721 are designed for voice communication (8 kHz spectrum) and are therefore inadequate for our application.

We accordingly selected the free lossless audio codec (FLAC, http://flac.sourceforge.net) and an adaptive differential PCM (ADPCM) codec for real-time rehearsal. The later is lossy and requires no buffering. As for the former, it is a lossless codec, and audio frame size can be handled down to the minimum size used by the audio device. Both have very little computational overhead. Compared to ADPCM's fixed compression ratio of 4:1, the coding efficiency of FLAC depends on the type of audio and ranges from 20 to 90 percent. Due to the lossy nature of ADPCM, we prefer FLAC and regard ADPCM only as a substitute for situations where better utilization of bandwidth is demanded.

Due to their superior compression gain, MP3 and MPEG-4 AAC are used for rehearsal on demand. Since startup service latency is relatively relaxed in this case, the prebuffering mechanism provided by the client software should be able to mask off the latencies involved. This is also much more economical regarding storage space at the server, which usually has to deal with a vast amount of pre-encoded performance sequences for instruments or voice parts. Also for the network, bandwidth resource is significantly saved when compressed audio contents are streamed over the link.

Due to their superior compression gain, MP3 and MPEG-4 AAC are used for rehearsal on demand. Since startup service latency is relatively relaxed in this case, the prebuffering mechanism provided by the client software should be able to mask off the latencies involved.

Server — With the architectural considerations mentioned earlier, we have designed the NMP server containing three major components:

Session management, which manages accounts and user-specific data (preferences, skills, instruments, location in the orchestra, channel association, etc.) and handles the negotiation for session setup (e.g., choosing an instrument, partners, music piece, style, instrument tuning, rhythm control, countdown). It also performs synchronization as described above.

Audio stream manipulation, which involves synchronization, mixing, and transcoding of audio as well as packet composition. Audio mixing refers to the procedure of multichannel audio creation. A client solicits its channel association during session setup, and the request is confirmed by the server if there is no conflict. Clients are identified by unique values. 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 sequence 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 compose a compound packet with multichannel audio content. The audio mixing and synchronization on the server are based on an algorithm that uses the sequence number field of the RTP header and processes all the packets that arrive in a certain time interval.

Audio repository management, which fulfills the requirement of recording and storing the performance examples. They can be used to emulate remote musicians in the rehearsal-on-demand case or for playback of a recorded live performance in the real-time rehearsal case upon request of the clients.

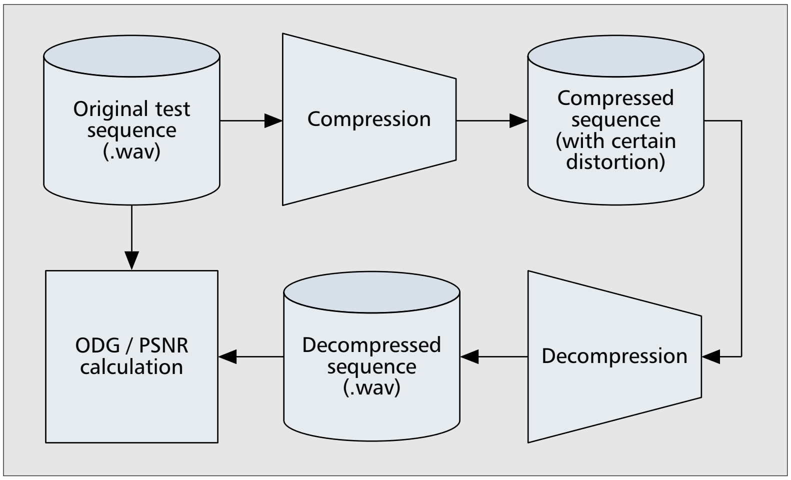

Original test sequence (.wav)
Compression
Compressed sequence (with certain distortion)
ODG / PSNR calculation
Decompressed sequence (.wav)
Decompression

Figure 2. *Test configuration A.*

Client — The NMP client provides end users access to the NMP service. Its architecture is described in our previous work in [11]. The NMP client is equipped with three major functionalities: access to the sound card hardware, an interface for service configuration, and clock synchronization with the server. Service configuration refers to tasks like instrument/voice part selection and tuning, music piece determination, rhythm control, performance starting point signaling, and partner selection, which are part of the service setup for NMP. The sound card is accessed in duplex mode to perform playback of the remote musicians' audio while recording local music performance. As buffering has to be set up individually for each sound card, we use an I/O abstraction layer that handles configuration and utilization of sound card buffers.

IMPLEMENTATION

We chose Linux as the platform for implementation of our prototype for flexibility reasons: along with lots of existing free and open source code hardware drivers and functional libraries, we are able to adapt the source code to our needs. The implementation was done in C++.

We use RtAudio (http://www.music.mcgill.ca/~gary/rtaudio/) as the abstraction layer for access to the audio hardware. RtAudio can easily handle buffer management and also provides an abstraction to the underlying drivers like Open Sound System (OSS) (http://www.opensound.com/pguide/oss.pdf) or Advanced Linux Sound Architecture (ALSA) (http://www.alsa-project.org). For maximum portability, common C++ libraries like ccRTP for audio transport, LAME as the MP3 audio codec, and QT as the GUI toolkit are reused.

The hardware used for implementation and evaluation consists of standard PCs connected via fast Ethernet switches, professional audio equipment, oscilloscopes, and sweep generators.

EXPERIMENTAL EVALUATION

The goal of the evaluation was to understand a number of factors that impact the audio quality, sourced from either the application itself or during data transmissions. The tests were performed in a local area network that satisfies the required bandwidth and delay boundary. Network conditions such as additional delay, jitter, and packet loss were emulated using the NIST Net network emulator (http://www-x.antd.nist.gov/nistnet). The factors selected for the analysis include distortion introduced by audio compression schemes, audio processing on the server, and network-congestion-related packet losses. In addition, end-to-end processing delay measurements were performed in order to prove the real-time constraint. Finally, the scalability of the NMP server in terms of computational complexity (i.e., number of supported clients and parallel sessions) was evaluated.

TEST PROCEDURE

To evaluate the distortion caused by the audio codec, we used two coding schemes: ADPCM and MP3. The measurement metrics utilized were peak signal-to-noise ratio (PSNR), mean opinion score (MOS), and objective difference grade (ODG).

PSNR is an objective non-human-based value derived from mean square error (MSE), which represents the correlation between the original and compressed signals. The output of the PSNR measurement is expressed in decibels. Typical PSNR values are 50 dB for AM radio quality, 70 dB for tape or FM radio quality, and 90 dB for CD quality [12]. MOS [13] is a common metric used for subjective quality measurements. It is human-based and can take values 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 from the original content.

As the PSNR does not reflect the perceptibility of distortions, and subjective assessments are time-consuming and expensive, better objective metrics have been developed in recent years for estimation of perceptual audio quality. One example is the Perceptual Evaluation of Audio Quality (PEAQ) [14] recommended by the International Telecommunication Union — Radiocommuniction Standardization Sector (ITU-R), which includes a low-complexity "basic" version and a complex "advanced" version for higher accuracy. The quality reflecting output variable ODG is calculated on a set of model output variables (MOVs) and corresponds to the subjective difference grade (SDG) in the subjective domain. The scale for ODG ranges from 0 (imperceptible impairment) to –4 (very annoying impairment). It should be noted that PEAQ models an underlying subjective experiment of listening test. Thus, for a particular audio sequence the ODG value may not correspond with the subjective quality rating of a certain listener.

For the end-to-end delay test we used a real-time two-channel oscilloscope and a sweep generator in order to achieve accurate measurement results.

In order to estimate the upper bound of the number of supported concurrent users and sessions, scalability tests on the NMP server were performed. A single state-of-the-art server machine was used. The focus here was to investigate to what extent the server is able to handle the related computational overhead. Other issues like storage capacity and network bandwidth were bypassed for the moment. Clients and sessions were emulated for this purpose. For each client, a thread was instantiated for tasks such as decompression. For each session, a thread was created for routines such as compression. For simplicity, in each test all sessions were set to equal sizes. The upper bound of the numbers were decided once the system load reached 100 percent.

Test Sequences

Ten different audio sequences with various characteristics and durations were used as sources for the tests. These publicly available sequences were selected from three different sources. Further details are available at http://www.ibr.cs.tu-bs.de/projects/nmp/audioseq.html.

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, where two different codecs, ADPCM and MP3, were used for the compression of 10 test samples (Fig. 2). The compression ratios for the ADPCM and MP3 codecs were set to 4 and 12, respectively. The quality parameter for the LAME encoder was set to 5 (mid-level). The original test samples were raw audio. The subjective quality measurements were performed by 15 test persons in one room with relatively low background noise level.

The impact of network parameters such as additional transportation delay and packet loss on the subjective audio quality was evaluated using test configuration B. This configuration includes two client machines and one server machine, which were connected via a NIST Net router (Fig. 3). The additional delay was set to 15 ms, and the packet loss ratio varied between 0.1 and 5 percent. The settings of the codecs as well as the test room remain the same as in test configuration A.

A single client processing delay was measured using test configuration C. We used a Wavetek sweep generator to form an input signal for the client machine. The delay consists of four main parts: audio capture and playback, compression processing, and decompression processing. We have connected the output of the sweep generator as well as the output from the client's sound card to the oscilloscope, in order to measure the time difference that represents the total client processing delay. We performed the latency measurements for the FLAC and ADPCM compression schemes, because only these codecs were used for the real-time scenario. The settings of the ADPCM codec remained the same as in the previous test configurations, while we left default settings for FLAC (i.e. compression complexity of 5) and adjusted the frame size to the audio device's input buffer size. This is always set to the minimum possible setting of 256 bytes.

In order to measure the end-to-end delay between two clients, we implemented test configuration D. This configuration includes two clients and one server, which were connected via a Fast Ethernet switch. 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. All settings remained the same as in test configuration C.

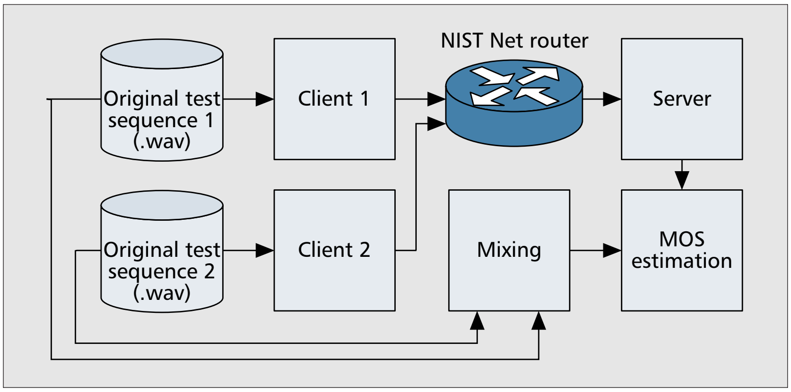

NIST Net router
Original test sequence 1 (.wav)
Client 1
Server
Original test sequence 2 (.wav)
Client 2
Mixing
MOS estimation

Figure 3. *Test configuration B.*

Analysis of Results

Test Configuration A — The PSNR and MOS values of the ADPCM codec measured for the 10 audio sequences in test configuration A ranged from 41.58 to 71.48 dB and from 2.36 to 5.00, respectively. The MOS output for eight of the 10 audio sequences exceeded a value of 3. The measured PSNR values for most sequences in this test configuration showed different behavior from the corresponding MOS values. In other words, PSNR and MOS values were only weakly correlated. For the 10 audio sequences, the average MOS value for ADPCM was 3.87 and the corresponding PSNR was 57.16 dB. However, the ODG values averaged at –2.65 and varied in a range between –0.89 and –3.80. Despite a positive correlation of 0.52 between MOS and ODG, there is a remarkable difference between the values. According to the average ODG, ADPCM achieves very poor quality; the subjective test results, however, show that the quality is still fair. Therefore, we consider using ODG only for evaluation of high-quality audio codecs such as MP3 and MPEG-4 AAC.

The average PSNR value for the LAME MP3 compression scheme was 44.84 dB, which was less than that of the ADPCM; however, the MOS averaged 4.84, which was almost 1.0 greater than that of the ADPCM. The better MOS values for MP3 are due to the psychoacoustic model used in the compression algorithm; the PSNR and MOS values of MP3 are even less correlated than the corresponding values of ADPCM. By contrast, the ODG values have a good match with the corresponding MOS values in this scenario (Fig. 4). Therefore, we conclude that ODG is a more suitable metric for objective quality evaluation of MP3. A detailed discussion of the results was presented in [11].

Test Configuration B — Pairs of audio sequences were transmitted using test configuration B. We used the same test pairs as in our previous work [11]. The average MOS values for ADPCM in this configuration are represented in Fig. 5. As we can see, packet losses have a considerable impact on perceived audio quality. Indeed, if the loss ratio exceeds 1 percent, the quality becomes very low. Therefore, error resilience mechanisms are indispensable in achieving acceptable audio quality in today's networks of best effort nature.

Test Configuration C — The delay measurements for test configuration C showed that our prototype was able to provide a processing delay as low as 10.5 ms. Audio capturing and playback each consumed 2.67 ms. The remaining 5.17 ms belongs to compression, decompression, and other end system overhead.

Test Configuration D — The average end-to-end delay for test configuration D achieved 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.

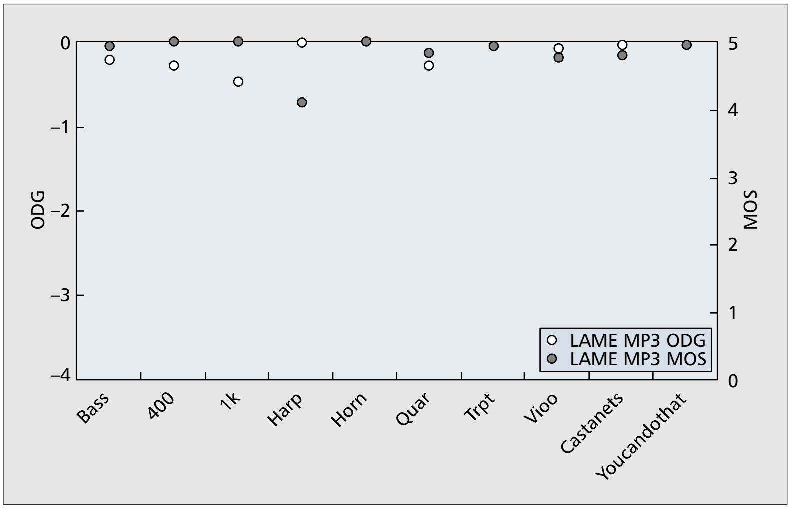


Figure 4. *MOS and ODG values for MP3 and configuration A.*

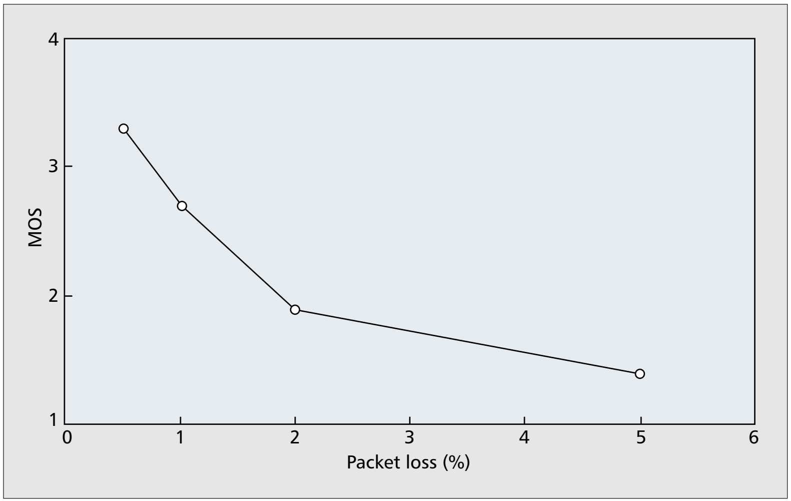


Figure 5. *MOS values for ADPCM and configuration B.*

Scalability Test — According to the results presented in Fig. 6, a single NMP server can handle a magnitude of 1000 concurrent clients. By varying the size of a session (or, in other words, the average number of participants in a session), the number of supported concurrent sessions changes reversely, while the total number of clients remains at nearly the same level.

Conclusions and Future Work

In this article we have introduced a new system called network-centric music performance. We designed and implemented a prototype of NMP using a testbed in a LAN with Linux PCs. We evaluated the objective and subjective audio quality of the application under different configurations with various test sequences. The impacts of network conditions on the performance and scalability of the NMP server were also investigated. We found that there is loose coupling between the objective (using PSNR) and subjective (using MOS) measurement results. We therefore adopted PEAQ and achieved ratings better coupled with MOS values. The system suffices for real-time constraints and perceptual audio quality for interactive multichannel natural audio delivery in such an environment, and scales well with increasing numbers of users.

Despite the results achieved in LANs that satisfied the real-time constraint, we are aware of the problems our prototype will face when extended to larger-scale best effort networks. In such networks the load is highly dynamic, and bandwidth fluctuations, packet losses, and delay jitter can occur. These are undesired for our application and will lead to severe service quality degradation. Therefore, our ongoing work is on quality of service support for NMP, delay and loss prediction for end system adaptation schemes, and error resilience and concealment mechanisms. Support for DCCP will be integrated into NMP for obvious reasons of congestion control for audio data delivery.

Extensive tests and performance analysis in the live Internet environment have to be carried out to further prove feasibility and determine application boundaries. To exploit scalability of a large NMP system, grid solutions will be taken into account. We have also started to extend the current implementation of MPEG-4 AAC toward an optimized low delay profile to improve the processing delay for rehearsal on demand. In addition, regression analysis of the collected data will be performed to provide more detailed results. By doing so, we hope to stimulate the progress of the evolution of music performance in an IT age, for professionals and enthusiasts.

Acknowledgment

We would like to thank the Communications Research Center Canada and Opticom GmbH for providing their PEAQ measurement tools CRC-SEAQ and OPERA 3.5 for our evaluation work.

References

[1] W. T. C. Kramer, "SCinet: Testbed for High-Performance Networked Applications," *IEEE Comp. Mag.*, vol. 35, no. 6, June 2002, pp. 47–55.

[2] C. Chafe *et al.*, "Effect of Time Delay on Ensemble Accuracy," *Proc. Int'l. Symp. Musical Acoustics,* Nara, Japan, Mar.–Apr. 2004, pp. 277–80.

[3] J. Lazzaro and J. Wawrzynek, "A Case for Network Musical Performance," *Proc. ACM NOSSDAV '01*, Port Jefferson, NY, June 2001, pp. 157–66.

[4] D. Konstantinas, "Overview of a Telepresence Environment for Distributed Musical Rehearsals," *Proc. ACM Symp. Appl. Comp.*, Atlanta, GA, Feb. 1998, pp. 456–57.

[5] J. P. Young and I. Fujinaga, "Piano master classes via the Internet," *Proc. Int'l. Comp. Music Conf.*, Beijing, China, Oct.1999, pp. 135–37.

[6] M. Goto *et al.*, "A Virtual Jazz Session System: VirJa Session," *Proc. Int'l. Comp. Music Conf.*, Hong Kong, Aug. 1996, pp. 346–49.

[7] A. Xu *et al.*, "Real-Time Streaming of Multichannel Audio Data over Internet," *J. Audio Eng. Soc.*, vol. 48, no. 7–8, July–Aug. 2000, pp. 627–41.

[8] J. R. Cooperstock and S. P. Spackman, "The Recording Studio that Spanned a Continent," *Proc. IEEE Int'l. Conf. Web Deliv. Music*, Florence, Italy, Nov. 2001, pp. 161–69.

[9] C. Chafe *et al.*, "A Simplified Approach to High Quality Music and Sound over IP," *Proc. COST-G6 Conf. Digital Audio Effect*, Verona, Italy, Dec. 2000, pp. 159–64.

[10] N. Bouillot, "The Auditory Consistency in Distributed Music Performance: A Conductor Based Synchronization," *ISDM Info Sci. for Decision*, vol. 13, no. 0, Mar. 2004, pp. 129–37.

[11] X. Gu *et al.*, "NMP — A New Networked Music Performance System," *Proc. 1st IEEE Int'l. Wksp. Net. Issues in Multimedia Entertainment*, Dallas, TX, Nov. 2004, pp. 176–85.

[12] A. J. Aude, "Audio Quality Measurement Primer," Intersil Corp., app. note AN9789, Feb. 1998.

[13] ITU-R Rec. BS.1116-1, "Methods for the Subjective Assessment of Small Impairments in Audio Systems including Multichannel Sound Systems," 1997.

[14] ITU-R Rec. BS.1387, "Method for Objective Measurements of Perceived Audio Quality," Nov. 2001.

Biographies

XIAOYUAN GU [StM](xiaogu@ibr.cs.tu-bs.de) is a research staff member and Ph.D. candidate at the Institute of Operating Systems and Computer Networks (IBR) of Technische Universität Braunschweig (TUBS). He received his M.Sc. in 1999 in information and communication technology from the International University in Germany. From September 2001 to September 2003 he was with the Panasonic Multimedia Communication European Laboratory as a research engineer. His research interests include network architecture, cross-layer design, and networked entertainment. He is a student member of ACM and ACF.

MATTHIAS DICK (dick@ibr.cs.tu-bs.de) received his diploma (M.Sc.) degree in computer science from TUBS in 2002. He is currently pursuing his Ph.D. at the IBR. His research interests include adaptive audio and video streaming, real-time applications, and QoS in mobile and wireless communications.

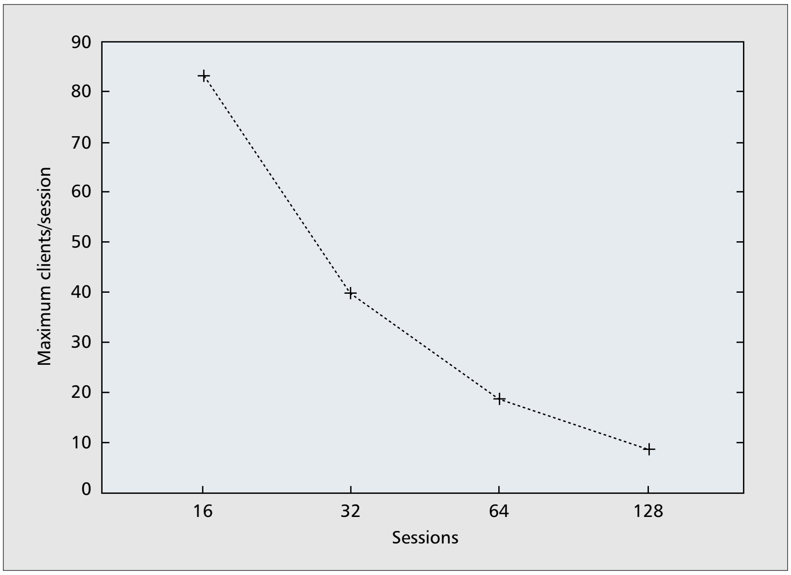


Figure 6. *Scalability test results.*

ZEFIR KURTISI (kurtisi@ibr.cs.tu-bs.de) is a research staff member and Ph.D. candidate at the IBR. He received his diploma (M.Sc.) degree in computer science from the University of Paderborn, Germany, in 1997. From 1997 to 2002 he worked as research engineer at the Sensormatic European Design Center, Munich, where he developed embedded remote video server solutions. His research interests are eLearning, multimedia coding, and content distribution.

ULF NOYER (unoyer@ibr.cs.tu-bs.de) received his diploma in January 2005 and is now pursuing his M.Sc. degree at the IBR. He has been working on TCP behaviors in wireless communication systems and networked music performance.

LARS WOLF [M] (wolf@ibr.cs.tu-bs.de) is a professor of computer science, and head of the IBR and the Computer Science Department at TUBS. He received his diploma (M.Sc.) degree in 1991 and doctoral degree in 1995, both in computer science. He became an associate professor at the University of Karlsruhe in 1999, after leading the Multimedia Networking Research Group for three years at TU Darmstadt. He has authored and co-authored eight books and book chapters, and published more than 90 papers. He serves as chair and TPC member of numerous conferences and workshops. He is an editorial board member of *Elsevier Journal of Computer Communications*, *Baltzer Journal of Telecommunication Systems*, *Kluwer Journal of Multimedia Tools and Applications*, and the journal *Praxis in der Informationsverarbeitung und Kommunikation* (PIK). He is a member of ACM, GI, and ACF.