Audio Engineering Society

Convention Paper

Presented at the 123rd Convention
2007 October 5–8 New York, NY

Network Music Performance with Ultra-Low-Delay Audio Coding under Unreliable Network Conditions

Ulrich Krämer[1], Jens Hirschfeld[1], Gerald Schuller[1], Stefan Wabnik[1], Alexander Carôt[2], and Christian Werner[3]

[1] *Fraunhofer Institute for Digital Media Technology, 98693 Ilmenau, Germany*

[2] *International School of New Media (ISNM), University of Lübeck, 23554 Lübeck, Germany*

[3] *Institute of Telematics, University of Lübeck, 23538 Lübeck, Germany*

Correspondence should be addressed to Gerald Schuller (`shl@idmt.fraunhofer.de`)

ABSTRACT

A key issue for successfully interconnecting musicians in real-time over the Internet is minimizing the end-to-end signal delay for transmission and coding. Anyhow, the variance of transmission delay ("jitter") occasionally causes some packets arrive too late for playback. To avoid this problem previous approaches are working with rather large receive buffers while accepting larger delay. In this paper we will present a novel solution that keeps buffer sizes and delay minimal. On the network layer we are using a highly optimized audio framework called "Soundjack" and on the coding layer we are working with an ultra low-delay codec for high-quality audio. We analyze and evaluate a modified transmission and coding scheme for the Fraunhofer Ultra-Low-Delay (ULD) audio coder, which is designed to be more resilient to lost and late arriving data packets.

1. INTRODUCTION

Whenever musicians play music in real-time over a network, which is usually called Network Music Performance (NMP), there are two main requirements: low signal delay (also called "latency") and high audio quality. Especially in case of an Internet transmission, the packet travel time tends to be quite variable. To compensate for this so called "jitter" most NMP systems introduce large sample buffers on the receiver-side. Unfortunately, this is contradictory to the idea of low signal delays.

Currently, there is a trade-off between low latency and high audio quality. Low input buffer sizes lead to short delays, because an incoming packet can quickly pass the buffer to be played out. A problem occurs if a packet is overly delayed on its way from the sender to the receiver, due to network jitter. It comes in after the input buffer has already run empty and must be dropped. In this case we may encounter an audio drop-out. Of course, we can reduce the risk of such an underrun by enlarging the input buffer size, but this, in turn, increases the signal delay.

In this paper we will show that there is a technical solution for achieving low latency and high audio quality at the same time: Unlike other approaches we address the problem of network jitter not by enlarging the size of the network buffer but by introducing an error resilient audio coding scheme. This way, less channel capacity is used, which in turn reduces jitter variance.

In our implementation we combine two technologies that originally were developed independently from each other: Soundjack and Ultra Low Delay Audio Coding (ULD). In [1] we already showed that this symbiosis is a possible way to enable NMP also in narrow-band networks, as for instance DSL connections.

The Soundjack framework is able to reduce buffering delays in the recording/transmission/playback chain to a minimum by accessing sound card and network directly. Depending on the actual network conditions between participating musicians, Soundjack allows all relevant sound card and network parameters to be adjusted in order to achieve the required signal latency for live music performances [2].

The Fraunhofer Ultra Low Delay Audio Codec is especially developed for applications with tight delay restrictions such as Network Music Performance. Its typical encoding/decoding delays range from 5.3 to 8 ms, or even down to 1.3 ms for specialized versions. The audio quality of the ULD codec is about comparable to MP3's audio quality at the same bit rate. The ULD does not use transform coding due to its inherent delay, but utilizes a psychoacoustically controlled pre-filter with a subsequent predictive coding stage.

The rest of this paper is structured as follows: in Section 2 we define the goals of our investigation, in Section 3 we present the problems we have to deal with to reach our goals. Section 4 includes previous approaches to create an error resilient coding scheme as well as a description of the ULD scheme and our new approach. In addition, we specify the setup of our listening tests in Section 5 and discuss the results of the tests in Section 6. The last section sums up the results and gives some conclusions.

2. GOAL

Our goal is to subjectively evaluate the jitter and dropout behavior of low buffered audio streams in the combination of Soundjack with the ULD coding scheme. In comparison to our previous Soundjack/ULD system [1] and our approach in [3], a better performance in terms of error robustness should be achieved. This should enable Network Music Performances with lower delay and higher audio quality in the presence of packet errors.

3. PROBLEM

Network Music Performances with tight delay boundaries, such as the realistic jam approach (RJA) [4], require data to be sent as fast as possible across a network. In an ideal scenario the one-way delay between two musicians should not exceed 25 ms. This latency does not only depend on the physical distance between two peers [1], but also on additional delay relevant aspects, such as routing and switching delays. Furthermore, the resulting overall delay is not constant, but varies with time.

3.1. Network Properties

At the Transport Layer, the ISO/OSI protocol stack (see Figure 1) offers the choice between either the *Transmission Control Protocol (TCP)* [6] or the *User Datagram Protocol (UDP)* [7]. The former works with network acknowledgments in order to confirm the arrival of sent data packets. While this principle improves the reliability of the message transport, it unfortunately introduces higher delay times. In comparison, UDP simply sends the data from the sender to the receiver without guaranteeing reliable transmission and hence provides the quickest signal delivery [9]. NMP with its strict delay boundaries requires this fast way of message transport.

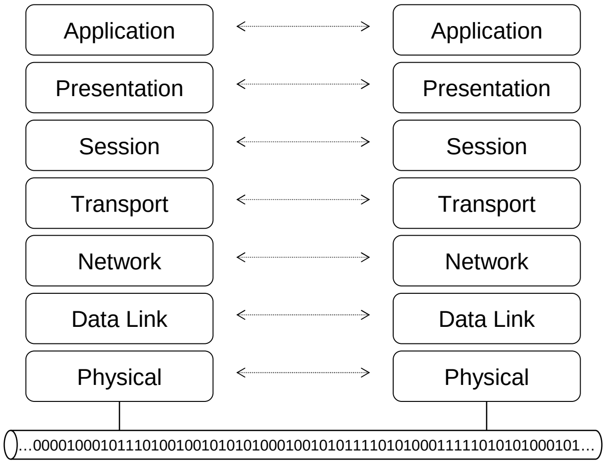

Fig. 1: ISO/OSI Layer Model (after [5])

On the Network Layer the Internet offers only one possible choice, which is the *Internet Protocol (IP)* [8].

Below the Network Layer, the Internet consists of various LAN architectures such as Ethernet, Token Ring, or ATM. These determine the technologies in the Data Link and the Physical layer. The main problem for high-traffic, real-time IP transmissions is the shared medium access, which is evident for any of these LAN technologies. If an IP packet containing time critical audio data should be sent over the network medium, this sending process might be blocked by other network traffic. Hence, the sending process must be postponed until exclusive media access is granted. These blocking times result in variations of the end-to-end delay. This phenomenon is called network jitter [9].

The currently most often applied LAN technology is the Ethernet. Compared to its competitors, it has the advantage of simplicity and robustness. But this causes a significant disadvantage in terms of real time traffic. Since Ethernet transmits frames with their full length it is possible that larger data packets block the transmission medium for a significant amount of time. In turn, small real time packets cannot pass the medium until the blocking data has been fully transmitted [9]. The maximum size of data packets is typically adjustable by the user (the so called *Maximum Transfer Unit (MTU)* and in general does not exceed 1,500 bytes. So-called jumbo frames of 9,000 bytes are an exception but can be applied under certain desired network setups. Assuming a 10 Mbps Ethernet component, a 1,500 byte packet would block the medium for 1.2 ms. With a 100 Mbps connection the propagation time for a 1,500 byte packet is 10 times lower at 0.12 ms.

Until the mid 90s, *Asynchronous Transfer Mode (ATM)* had been considered as the preferred network architecture for the two lowest ISO/OSI-Layers [9]. Due to its time slot based signal delivery it might be the first choice for Network Music Performances. ATM splits the data into cells of 53 bytes, which requires less transmission time and could reduce the jitter to a minimum.

Other examples for network architectures are the *Plesiochronous Digital Hierarchy (PDH)* or the *Synchronous Digital Hierarchy (SDH)*[9], but none of them has reached a major breakthrough on the network engineering market. What has happened instead, is their replacement with Ethernet networks. Ethernet has become the most important applied technology within the last 10 years. In Germany for example, the DFN (Deutsches Forschungsnetz) has been replacing an SDH backbone by Gigabit Ethernet over the past decade.

3.2. Delay Measurements

In order to thoroughly evaluate the problem of network jitter in real networks and its effects on NMP applications, we conducted numerous tests in various setups. At first we assumed that the problem of jitter is mainly critical in setups with long distance connections, because here the UDP packets must be typically routed via many IP subnets, each one possibly causing some amount of jitter.

Surprisingly, a serious amount of network jitter is not only observable in setups with many hops between sender and receiver but occurs even in very small network setups. Even inside a single subnet consisting of only four hosts we detected a surprisingly high variance of packet delay. Additionally, our tests showed that the amount of jitter is strongly dependent on the traffic inside the subnet. If the network load in a subnet exceeds ten percent of its maximum capacity, jitter becomes noticeable. This is presumably caused by the queuing algorithms used in switches, but currently we have no comprehensive explanation for this phenomenon.

Our experiments showed that the amount of traffic that is transferred over the network has a strong effect on the amount of jitter. In the following we will show the results of an experiment that has been carried out under a realistic network load situation on the campus of the University of Lübeck.

We used an audio reference stream of 48 kHz, 16 bit sampling, 1 channel and 128 samples per frame, representing 2.6 ms of music. This generates data packets of 256 bytes when using uncompressed PCM encoding. By applying the ULD data compression this audio signal was encoded using 33 bytes per packet.

In our experiment we transmitted 52 seconds of our audio stream between two hosts, both located on campus (distance: 5 km, 6 network hops). In a first run, we transmitted the PCM encoded signal and in a second run we transmitted the ULD encoded one. There might be minor changes in the network load situations between these two runs, but we repeated the whole experiment several times in order to ensure that the presented results are reproducible.

In Figure 2 we see a histogram of the occurring round trip times (RTT) between the two hosts when using PCM (packet size: 256 bytes) and ULD (packet size: 33 bytes) encoding. The one way packet delivery time equals about half of this value.

Because the load situation in the network has not significantly changed between the two runs, it is not surprising that the median of both measures is identical (4 ms). However, as we see in the diagram the variance is clearly higher when using PCM encoding. By applying the ULD codec we can obviously reduce the amount of jitter.

In order to show the effects of the network jitter on the received audio data, we plot the number of the audio dropouts occurring in blocks of 1,000 subsequent packets in Figure 3. In both runs we set the maximum network buffer size to two packets, corresponding to a buffer delay of 5.2ms. If the time interval between the arrivals of two subsequent packets exceeds this value, a buffer underrun occurs resulting in a disruption of the audio data stream, which has the potential to produce strong artifacts for the audio playback. From the histogram in Figure 2 we can estimate the fraction of packets exceeding the buffer time. We need to divide the round trip time

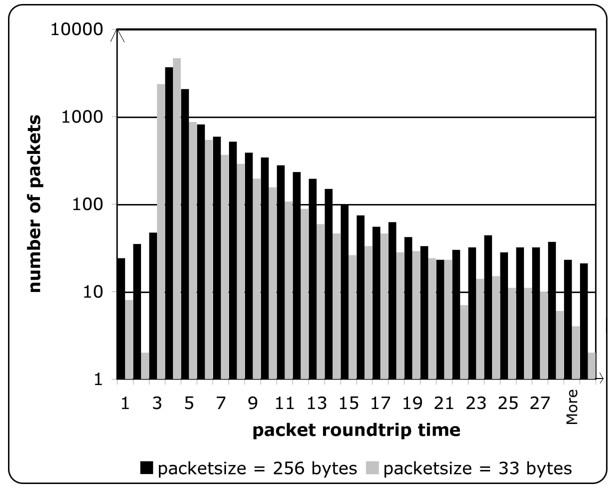

Fig. 2: Histogram of network round trip times between two University hosts (August,1st, 2007, 5:00 pm)

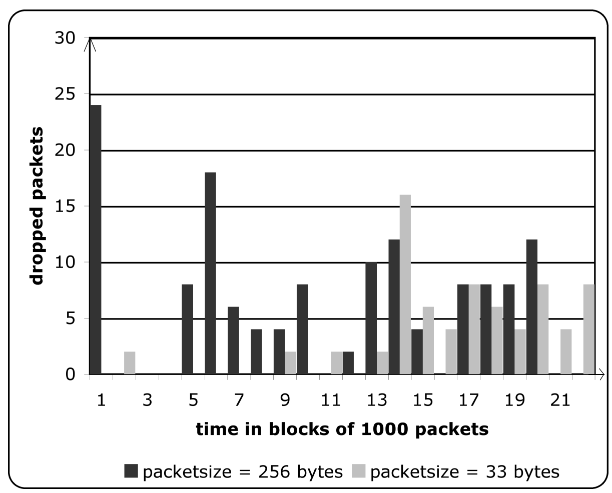

Fig. 3: Dropouts between the two University hosts (August,1st, 2007, 5:05 pm)

by 2, and add the buffer time to the median delay value. Packets with delays above this value are lost.

As one can see, the PCM encoding produces significantly more dropouts than ULD encoding. Although in block #14 the ULD encoded stream produces slightly more dropouts than the PCM version it is clearly visible that we can reduce the overall dropouts by using data compression.

Even though there was always enough bandwidth for our audio stream, obviously network jitter and as a consequence late arriving packets were present. In order to avoid packet losses, an increase of the receiver's network buffer would be necessary, but this would also produce higher delays. Hence, our approach is to keep the buffer small and to reduce the number of lost packets by applying ULD compression.

Besides this, we have to mention that in fact the most efficient solution for reducing packet losses is the use of overprovisioned network connections such as Gigabit backbones. Research carried out by Sinha [12] shows that the use of overprovisioned network connections reduces the jitter to a barely noticeable minimum.

In DSL networks our reference audio stream introduces a significant and high amount of packet overhead of 375 kbps. With an uncompressed audio stream this results in a total amount of more than 1 Mbps. Due to the small DSL upload bandwidth between 128 kbps and a maximum of 1 Mbps, this overhead prevents the use of uncompressed audio signals with conventional DSL connections. With an ULD coded payload of, for instance, 96 kbps, this results in 471 kbps and in turn allows the use of certain DSL lines [1].

In order to investigate the properties of DSL connections we conducted another experiment with two hosts that were both connected with the Internet over two 16 Mbps DSL links. Both hosts were located in Lübeck (distance: 2 km, 4 network hops) and ULD compression had been applied. Figure 4 shows the RTT histogram and Figure 5 depicts the corresponding dropout characteristic.

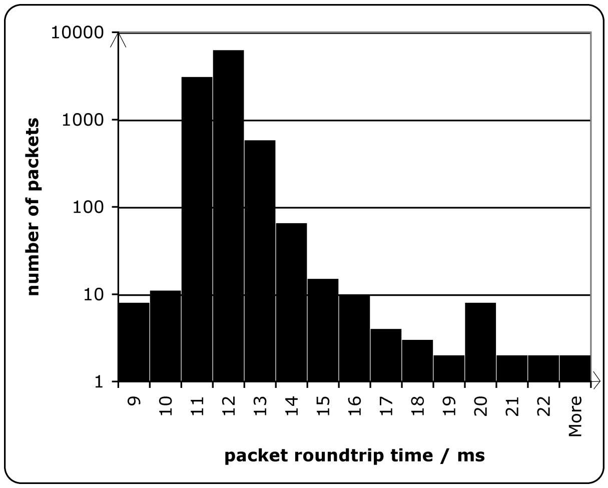

Fig. 4: Histogramm of network round trip times between two hosts interconnected using two Deutsche Telekom 16Mbps DSL links (July 30th, 2007, 2:00 am)

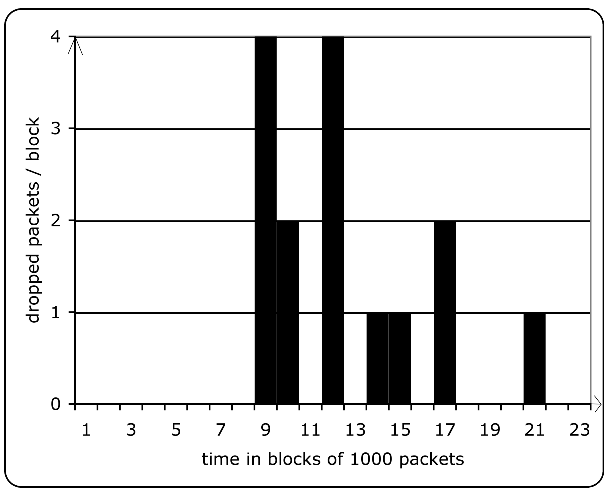

Fig. 5: Dropouts between two Deutsche Telekom 16Mbps DSL Connections (July 30th, 2007, 2:00 am)

4. APPROACH

4.1. Previous Approaches

A lot of literature has been published on error concealment and error resilience of audio and speech coding systems. In context with the ULD Coder, especially speech coding resilience techniques are of interest, as the ULD's redundancy reduction stage resembles traditional speech coders. Recently, investigations on the resilience of Voice over IP (VoIP) has been given a lot of attention. These are of particular interest, because the Soundjack/ULD combination has to face the same network conditions as VoIP does. Still, the constraints are quite different. Whereas delay is a critical issue in voice communication, too, the tolerable limit is much higher for internet telephony than for NMP. Because of this fact, concealment techniques such as applying adaptive delays [10] or interpolation of the lost blocks [11] cannot be employed in our context. For multi channel audio signals, Sinha et al. proposed an interpolation scheme that takes into account the correlation between adjacent channels in a multi channel scenario [12]. As we normally just transmit one mono instrument channel, we cannot take advantage of this strategy. A simple solution for NMP is to transmit uncompressed PCM coded packets. This way, an error due to packet loss is not propagated. The drawback is that the bit rate increases. Narrow channels, like the uplink channel of an asymmetric DSL line, might not support this increased bit rate. Even if this channel supports the increased bit rate, the higher bit demand generally leads to a wider spread of the delay jitter, and hence to increased packet losses, as can be seen in our measurements in section 3.2. These show that the packet loss rate because of jitter will increase the closer the bit rate is to the channel capacity.

4.2. Overview Ultra Low Delay Coder

The Ultra Low Delay Coder is a perceptual audio coder that uses separate coding blocks for the two goals of irrelevancy and redundancy reduction (Fig. 6). First the input signal is analysed and a masking threshold is calculated for every block of 128 samples. From the masking threshold, filter coefficients for the pre-filter are calculated and transmitted alongside with the audio data. The pre-filter uses these coefficients to normalize the input signal with respect to the masking threshold.

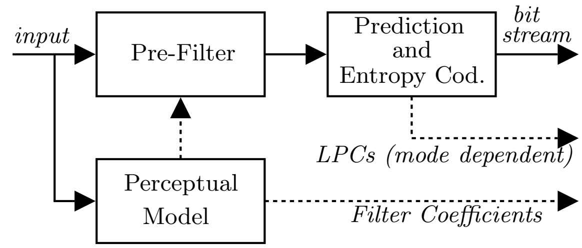

Fig. 6: ULD Encoder block diagram

Quantization is applied together with the prediction in a closed loop fashion. The predictor is part of the redundancy reduction stage. It minimizes intra-channel correlation, or – in other words – spectrally whitens the pre-filtered signal. Prediction can be used in either a forward or a backward adaptive way. After prediction, entropy coding can be applied to reduce the average bit demand.

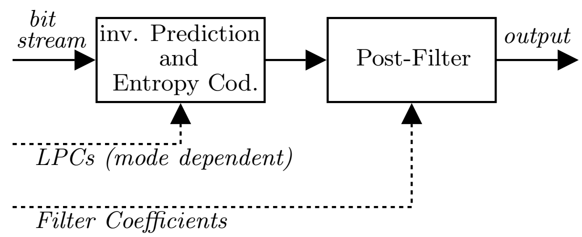

Fig. 7: ULD Decoder block diagram

The ULD decoder (Fig. 7) first reverses the entropy coding, if it was applied in the encoder. After inverse prediction, the post-filter, which is the inverse filter to the pre-filter, re-shapes the audio signal. Because the frequency response of the post-filter resembles the masking threshold, the quantization noise is shaped like the masking threshold and thus remains inaudible.

Our earlier approach to introduce error resilience strategies into the ULD system was to use regularly spaced resets with a backward adaptive predictor[3].

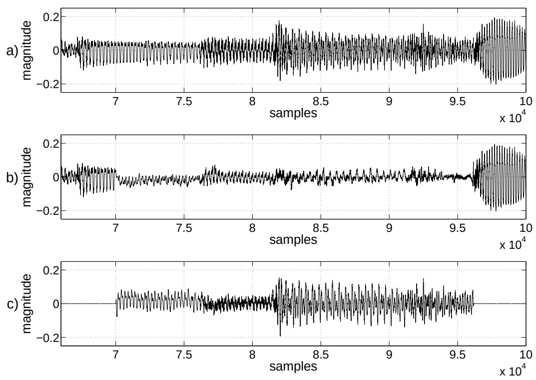

Fig. 8: Error propagation of the ULD with backward adaptive prediction after packet losses (128 samples) at position 70016 and 73856; a) decoded signal without error, b) decoded signal with packet loss, c) difference between a) and b)

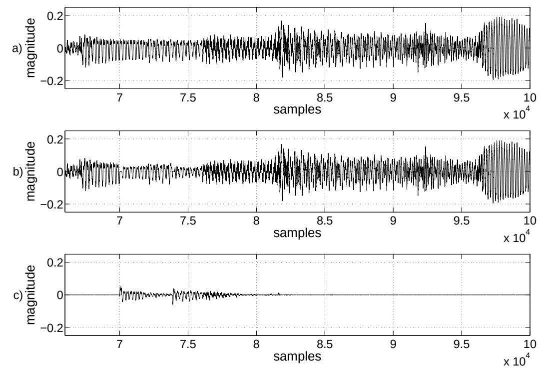

Fig. 9: Error propagation of the ULD with forward adaptive prediction after packet losses (128 samples) at position 70016 and 73856; a) decoded signal without error, b) decoded signal with packet loss, c) difference between a) and b)

The more frequent the predictor is reset, the sooner the decoder can re-synchronize after a packet loss. But frequent resets also decrease the coding gain, thus the bit rate has to be increased to maintain the same audio quality. Between the detection of a lost packet and the next reset we applied an adaptive reshaping filter to the decoded prediction residual signal as a substitute for the misadjusted inverse predictor (see Figure 8 for example music waveforms). The filter coefficients for the reshaping filter were calculated from the correctly received signal just before the error and then kept constant until the next reset instant. From then on, the actual inverse predictor was activated again.

4.3. New Approach

The new error resilience approach builds upon a new ULD structure that was first described in [13] and [14]. This method also applies psychoacoustically controlled pre- and post-filtering just as the earlier version did. But instead of calculating the Linear Predictive Coefficients (LPCs) for the closed loop prediction in a backward adaptive fashion, we now determine these in a block-wise, forward adaptive way from the pre-filtered signal. This also means we now have to code and transmit this set of LPCs every block. To save bit rate, this is done in the Line Spectral Frequency (LSF) domain. The residual signal within the closed-loop prediction is quantized uniformly with a fix number of quantization levels. The quantization step size, however, is controlled backward adaptively.

The main advantage of this new approach compared to our earlier ULD versions is that after a packet loss the bit stream can be decoded without the need to wait for a predictor reset. Due to the backward adaptive step size control and the internal states of the predictor, however, some error propagation occurs (see Figure 9 for example music waveforms), although much less than for the backward adaptive coder (Figure 8). The lost packets themselves are conceiled by an adaptive filter just like in the earlier approach.

5. LISTENING TESTS

In this paper we will compare three versions of the ULD coder with respect to their resilience to packet errors. The first and the second coder are based on the backward adaptive predictor, as descibed at the end of section 4.2. The two coder only differ in the distance between consecutive predictor resets: one second for the first coder, 0.1 seconds for the second coder. The third coder is the ULD with forward prediction, as described in section 4.3.

To evaluate the error resilience of the three coders under test, we used a two-stage approach. Because we are only interested in assessing the quality under strained network conditions, we first generated error patterns with the Soundjack/ULD combination during high network traffic. Then we simulated the bad network conditions by applying these error patterns to the bit streams of the encoded reference test set. This way we do not have to wait for eventual dropouts, and we can guarantee that all listeners are presented the same audio material to judge.

We evaluated the performance of the ULD error resilience under varying degrees of network reliability. We generated three error patterns with different severity of disturbance. The first was recorded from a connection Lübeck–Stanford which had relatively good conditions, i.e. an overall packet loss rate of about 0.1% and mainly singular packet losses. The second pattern was recorded from a connection Lübeck–Ilmenau, which showed a loss rate of about 1% and included several burst errors. In the third scenario the pattern revealed a loss rate of more than 15%, with heavy burst errors. This pattern was recorded from a DSL connection between two sites in Lübeck.

Our audio test data mainly consisted of single instrument (and voice) samples from the EBU SQAM (Sound Quality Assessment Material) CD [15]. In addition to that we used an excerpt from a Jazz Piano piece played by Keith Jarrett, a rhythm section part from the band Fantastische Vier and an acoustic guitar solo by the band Nirvana. All segments we used have a duration of about ten seconds. All items were resampled to a sampling rate of 32 kHz and a resolution of 16 bit.

The three listening tests were conducted as MUSHRA [16] tests in a silent office environment with seven experienced listeners. The MUSHRA test was implemented on a Laptop computer with STAX amplifier/headphones attached to an external DA-converter.

6. RESULTS

The results of our listening tests can be seen in Fig. 10, Fig. 11, and Fig. 12.

Fig. 10 shows the results for the case of the bad channel. Here it is obvious that for each individual item there is a statistical significant difference in the grading for the new forward adaptive version compared to the backward adaptive version with 1 s reset intervals. The significant difference can be seen in the non-overlapping confidence intervals. The overall average shows a significant difference between all 3 coder versions, with the new forward adaptive version being the best, followed by the backward adaptive version with 0.1 s reset interval, and the backward adaptive version with 1 s reset interval last. But observe that all are in the "poor" and below region, so this is a channel to avoid for NMP.

Fig. 11 shows the results for the case of the medium quality channel. Here it is interesting to see that the forward adaptive coder and the backward adaptive coder with 0.1 s reset interval show no significant difference, and are both in the "fair" region, and hence becoming usable.

Fig.12 shows the most positive case for NMP, the channel with good quality. As mentioned in Section 3.2 this positive case can be achieved by using networks with high capacity and relatively low traffic load, or by using networks with a Quality of Service (QoS) option. Here it is interesting to observe, that for the first item (Fanta4_32m) the forward adaptive coder becomes statistically indistinguishable from the original. Also in the overall average it has a rating in the "excellent" range, and is statistically significantly better than the other 2 coder versions. This case becomes suitable even for more demanding NMP sessions.

7. CONCLUSIONS

Our goal was to to obtain a system for Network Music performance over Internet or even DSL connections. We showed that compression of audio signals is important, because IP connections, and DSL connections in particular, add considerable overhead to the overall data rate. Even if a higher data rate is available, the lower bit rate of compressed audio signals decreases the jitter and hence decreases the packet loss rate at the receiver. Our listening tests showed that even in relatively good channel conditions, error resilience and error concealment of the audio coder becomes crucial. The tests further showed that the network condition has a major influence on the decoded audio quality. Hence it is important to have networks with relatively little traffic load, and it would be desirable to have a network with Quality of Service (QoS), as in the Next Generation Internet.

A recorded session of two musicians using the Soundjack/ULD approach can be found under *http://www.livemusicportal.eu* and *http://www.idmt.fraunhofer.de/schuller.html*.

8. REFERENCES

[1] A. Carôt, U. Krämer, G. Schuller, "Network Music Performance (NMP) in Narrow Band Networks," presented at the 120th AES convention, Paris, France, May 20–23, 2006

[2] A. Carôt, A. Renaud, B. Verbrugghe, "Network Music Performance (NMP) with Soundjack," 6th International Conference on New Interfaces for Musical Expression (NIME 06), Paris, France, June 4–8, 2006

[3] S. Wabnik, G. Schuller, J. Hirschfeld, U. Krämer, "Packet Loss Concealment in Predictive Audio Coding," presented at the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, October 16–19, 2005

[4] A. Carôt, A. Renaud, P. Rebelo, "Networked Music Performance: State of the Art," presented at the AES 30th International Conference on Intelligent Audio Systems, Saariselkä, Finland, March 13–15, 2007

[5] ISO/IEC 7498-1, "Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model," Reference Number ISO/IEC 7498-1:1994(E), Second Edition, corrected an reprinted, June 15, 1996

[6] J. Postel, "Transmission Control Protocol," RFC 793 (Standard), updated by RFC 3168, published by the Internet Engineering Task Force (IETF), September 1981

[7] J. Postel, "User Datagram Protocol," RFC 768 (Standard), published by the Internet Engineering Task Force (IETF), August 1980

[8] J. Postel, "Internet Protocol," RFC 791 (Standard), published by the Internet Engineering Task Force (IETF), September 1981

[9] A. S. Tanenbaum, "Computer Networks," ISBN: 0130661023, Pearson Education, 4th Edition, August 2002

[10] F. Liu, J. Kim, C.-C. J. Kuo," Adaptive delay concealment for internet voice applications with packet-based time-scale modification," Proceedings of the IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP), Volume 3, pp. 1461–1464, Salt Lake City, USA, 7-11 May 2001

[11] N. Aoki, K. Takano, I. Ohmura, H. Arai, T. Namerikawa, "Development of a VoIP system implementing a high quality packet loss concealment technique," presented on the Canadian Conference on Electrical and Computer Engineering, pp. 308–311, ISBN: 0-7803-8885-2, May 1–4, 2005

[12] R. Sinha, C. Papadopoulos, C. Kyriakakis, "Loss concealment for multi-channel streaming audio," Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video, pp. 100–109, ISBN: 1-58113-694-3, June 1–3, 2003

[13] S. Wabnik, G. Schuller, J. Hirschfeld, U. Krämer, "Different Quantisation Noise Shaping Methods for Predictive Audio Coding," presented at the IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, May 14–19, 2006

[14] S. Wabnik, G. Schuller, J. Hirschfeld, U. Krämer, "Reduced Bit Rate Ultra Low Delay Audio Coding," presented at the AES 120th convention, Paris, France, May 20-23, 2006

[15] Technical Centre of the European Broadcasting Union (EBU)," Sound Quality Assessment Material – Recordings For Subjective Tests," Tech. 3253-E, April 1988

[16] ITU-R BS.1534-1," Method for the subjective assessment of intermediate quality levels of coding system," January 2003

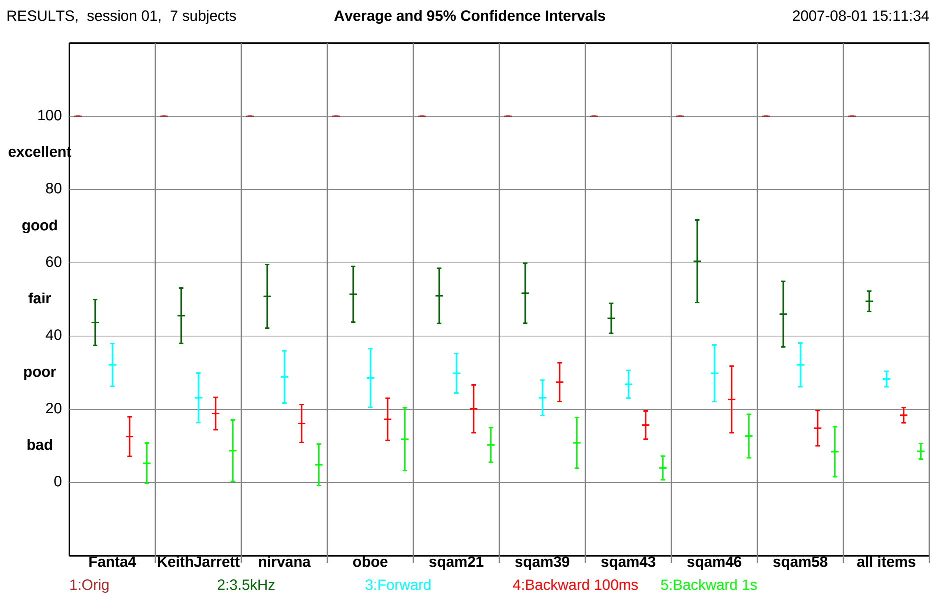

Fig. 10: Listening test results for bad channel conditions; 1: Original, 2: 3.5 kHz anchor, 3: ULD with forward prediction, 4: ULD with backward prediction and reset every 100 ms, 5: ULD with backward prediction and reset every second

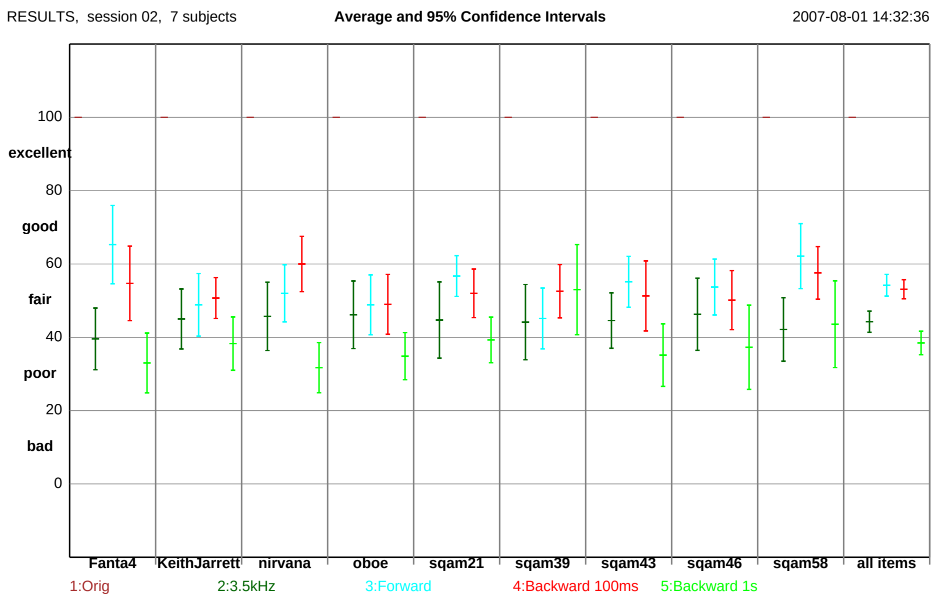

Fig. 11: Listening test results for medium channel conditions; 1: Original, 2: 3.5 kHz anchor, 3: ULD with forward prediction, 4: ULD with backward prediction and reset every 100 ms, 5: ULD with backward prediction and reset every second

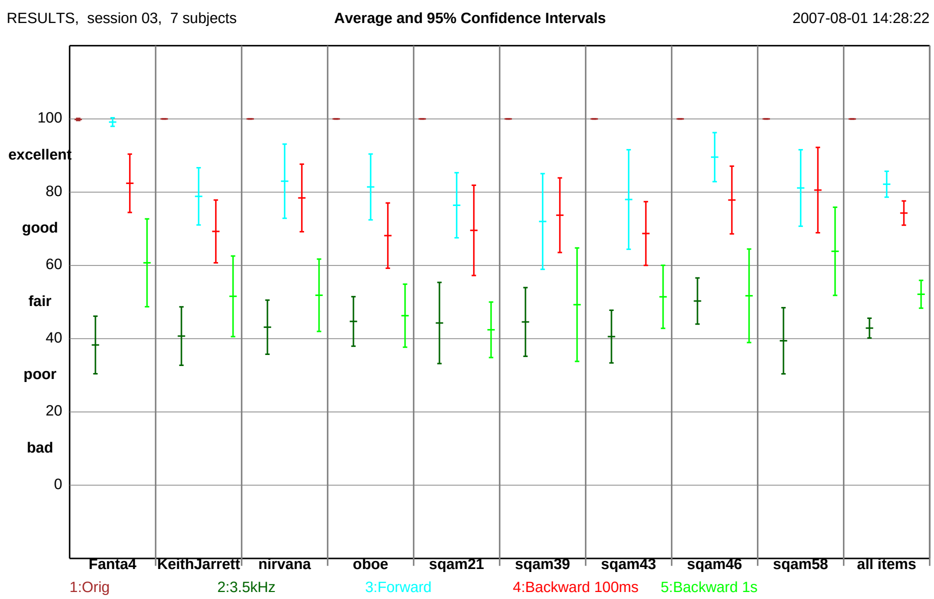

Fig. 12: Listening test results for relatively good channel conditions; 1: Original, 2: 3.5 kHz anchor, 3: ULD with forward prediction, 4: ULD with backward prediction and reset every 100 ms, 5: ULD with backward prediction and reset every second