USING WAVPACK FOR REAL-TIME AUDIO CODING IN INTERACTIVE APPLICATIONS

Zefir Kurtisi, Lars Wolf

Institut für Betriebssysteme und Rechnerverbund, Technische Universität Braunschweig, Germany

ABSTRACT

Highly interactive real-time music applications like network music performance (NMP) currently lack appropriate audio codecs that meet both demands for low delay and high audio quality. Music audio codecs introduce unacceptable processing latencies, while low-delay speech codecs provide insufficient audio quality. In this article we describe how we integrated the *WavPack* audio coding scheme into our NMP system for interactive distributed network centric music performances. Our evaluation shows that WavPack performs very well in such types of delay-critical applications. In NMP it provides high compression ratios and good audio quality without additional buffering delays.

***Index Terms*—** real-time audio compression, interactive applications, network music performance

1. INTRODUCTION

Distributed interactive real-time music performances over the network are one of the most delay sensitive applications today. Musicians can play together over the network only if the total delay is kept to those values perceived in real world environments. As shown in [1] musicians can typically tolerate a total delay of around 30 ms. In our NMP project we created a system capable of providing musicians a network centric virtual stage for interactive music performances within this strict delay bound. Since musical applications require hi-fi audio quality, the simulation of virtual stages with multi-channel surround audio results in very high audio data rates.

Therefore, audio data compression is essential for a practical utilization of NMP, but none of the audio codecs established today meet the requirements for this strict delay bound and for hi-fi audio quality. Music codecs with transparent audio quality at compression ratios below 0.1 work in the frequency domain and therefore introduce unacceptable processing delays, while real-time capable speech codecs provide insufficient audio quality. Our approach to solve this conflict was to look at audio codecs outside the mainstream. We adopted the open source audio codec WavPack into our system and found out that it is currently the best available choice for the demands of NMP.

The rest of this article is organized as follows: in section 2 we give a short overview of our NMP system, describing its basic principles and the resulting bounds and demands for appropriate audio codecs. Section 3 describes how we integrated the WavPack audio coding scheme into our NMP system. In section 4 we present performance evaluation results reflecting WavPack's suitability for our system, and then we conclude this article in section 5.

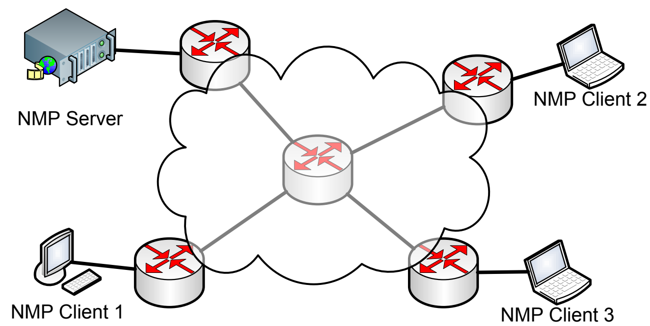

Fig. 1. Basic NMP Principle

2. NMP: A SYSTEM FOR INTERACTIVE REAL-TIME NETWORK MUSIC

Our NMP system is a working prototype application for distributed network centric interactive real-time music performance. It is based on a centralized server to which several clients are connected, as depicted in figure 1. Musicians can connect through the NMP client software to the server and join music sessions to play together. The core function of the client is to grab audio data from the sound hardware and to send audio packets to the server over an IP-network. The server synchronizes and mixes audio data from all clients in a session and returns a copy of the collective audio stream to all clients, where the audio data is played back. So far, NMP looks quite similar to existing interactive communication applications, e.g. Internet audio conferencing or VoIP. What makes NMP special and technically challenging is the unprecedented demand for both low delays and hi-fi audio quality in one single codec.

Thanks to David Bryant, lead developer and maintainer of the WavPack (`http://www.wavpack.com`) audio codec for his great work and kind support in adopting WavPack for NMP.

2.1. Delay Bound

Compared to existing on-line collaboration applications the delay bound of 30 ms for NMP is a full magnitude tighter. This prevents the adoption of established techniques currently used in interactive applications and requires specific approaches to allow the transmission of isochronous audio data over a non deterministic IP-network.

In [2] we performed a detailed delay analysis of our NMP system and realized that audio data resides in the end systems (client and server) for about half of the delay budget. This is caused mainly by the latency introduced from the packetization of audio data in computer audio hardware. Our analysis showed that an optimal setup operates with audio blocks of 128 audio samples at 48 kHz, our NMP system works with a constant packet rate of 375 $\frac{packets}{s}$ resulting in a packetization and buffering delay of t_φ=2.66 ms. This value defines the granularity of the total system delay, which has a theoretical lower bound of $5 \times t_\varphi$. This leaves about 16 ms for the network delay, which is sufficient to set up NMP sessions with network distances up to 500 km in a broadband network. Musicians between universities or within metropolitan areas can play music in a natural way with good audio quality and a total system delay below 30 ms. To maintain at least this limited application radius each extension to the prototype must not increase the delay, which is especially relevant for audio coding: a suitable audio codec must operate with block sizes of 128 samples to allow pipelined processing in the end systems without additional buffering delays.

2.2. Audio Quality and Compression

The demand for high sampling rates to reduce packetization delays in the sound hardware inherently leads to very high audio data rates. A mono audio channel sampled at 48 kHz with 16 bits per sample results in a constant data rate of 768 kbps. Considering that the server simulates virtual stages by providing client individual multi-channel surround audio, the use of audio compression is obligatory.

Today's established audio codecs can be classified into two groups. Music codecs are known through their omnipresence in everyday live as standard streaming and archiving formats for digital audio in the consumer world – MP3 is a synonym for on-line music. Like MP3 or AAC, all these codecs work in the frequency domain where a psychoacoustic model is used to remove information of less importance for human auditive perception. This leads to very high compression ratios together with very high audio quality. Good implementations of MP3 provide nearly transparent CD audio quality with 128 kbps, however the analysis involved requires a large number of audio samples and therefore introduces high buffering delays (see [3]). Delay optimized variants of these codecs (e.g. AAC-LD) reduce the coding delay down to 20 ms which is sufficient for interactive communication applications like video conferencing, but remains unsuitable for NMP.

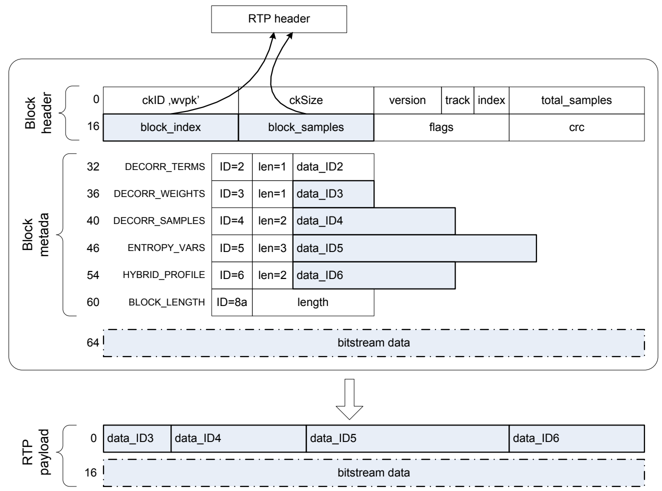

Fig. 2. WavPack header repack

The other class of audio codecs are those designated for speech or telephony communication used in DECT, GSM or VoIP. They work in the time domain and operate with small block sizes to achieve low latencies. By design they do not support wideband audio data and provide insufficient quality for music applications. Therefore they can not be used in NMP.

Due to this lack of appropriate audio codecs for NMP, in our first prototype we implemented an ADPCM compression scheme that worked in place with the block sizes given by the audio hardware and without additional buffering delays. After we realized in [4] that the provided audio quality is not acceptable for music, we evaluated alternative audio codecs that are freely available and meet our requirements best.

3. ADOPTION OF WAVPACK IN NMP

WavPack is an open audio codec written and primarily maintained by David Bryant. It is distributed under the BSD license and is intended to be completely free through employment of only well known public domain techniques and by avoiding patented algorithms. We give a brief overview on WavPack here; more detailed information on its compression algorithms can be found at the project site[1] and in [5].

WavPack supports a variety of input audio formats and meets all of our requirements for NMP. In lossless mode WavPack reduces data by prediction and entropy coding in the time domain. In typical off-line applications (like archiving of audio CDs) compression ratios of 0.3 to 0.7 can be achieved. By principle encoded bit-streams in lossless mode have a variable bit-rate that is dependent on the audio signal's entropy. This becomes problematic in scenarios where a given datarate must not be exceeded. Here the lossy coding scheme of WavPack is a key benefit for the use in NMP. The encoder can

[1] WavPack project site: `http://www.wavpack.com`

be parameterized with a maximum target bit-rate that causes the encoder to cut off all remaining less significant data from the encoded bit-stream. This allows NMP to operate outside broadband networks where it is essential to limit the bandwidth. The target bit-rate ranges down to 2.25 bpS (bits per Sample) which is sufficient to reduce the net bit-rate for a monaural 48 kHz 16 bpS audio stream from 768 to 108 kbps.

WavPack is designed to work efficiently with large block sizes. If otherwise small blocks are used as in NMP, the block header overhead impairs its coding efficiency. For a monaural audio stream WavPack uses a header of around 64 bytes per block, so that with 128 sample blocks the header overhead significantly downgrades the compression efficiency. An existing low-latency version of WavPack employs a super/sub-block encoding scheme that reduces the header overhead at the cost of adding inter-block dependencies and therefore is not applicable for NMP.

Instead, we implemented an intermediate layer on top of the WavPack codec library that repacks the encoded bit-stream to remove remaining redundant data, as depicted in figure 2. We see two different sections in the header: each audio block starts with a block header that contains information required to assure correct decoding and placement of that block, e.g. its position in the data stream, and the checksum. Following is one meta-data block for each channel encoded that includes prediction and state values required for the decoder. The encoded bit-stream data follows the header.

Since in NMP we perform sequencing in the RTP layer and bit errors are typically detected in lower layers of the network protocol stack, some fields in the block header are redundant. The remaining are either constant or not required for our application, so we can remove the block header completely and save 32 bytes in each audio packet. Furthermore, together with the codec's author, we worked out which fields in the meta-data are essential and which can be replaced by the intermediate layer on the fly. This allows us to remove the type and length fields and saves another 16 bytes per block.

With the implemented intermediate layer the sender repacks the values of the obligatory meta-data and the bit-stream returned by the encoder into the RTP payload. At the receiver we reconstruct the original block before it is passed to the decoder to maintain format compliance. In the common case of mono audio data transmission from NMP clients to the server this reduces the header overhead from 64 to 16 bytes per packet and significantly improves WavPack's compression efficiency with small block sizes.

4. EVALUATION

The most critical requirement for NMP is the ability to work with small block sizes, which is supported by WavPack natively. With the described header repacking even blocks of 128 samples can be compressed quite efficiently. Beyond this prerequisite we evaluated WavPack's performance for compression efficiency and audio quality and compared it to ADPCM.

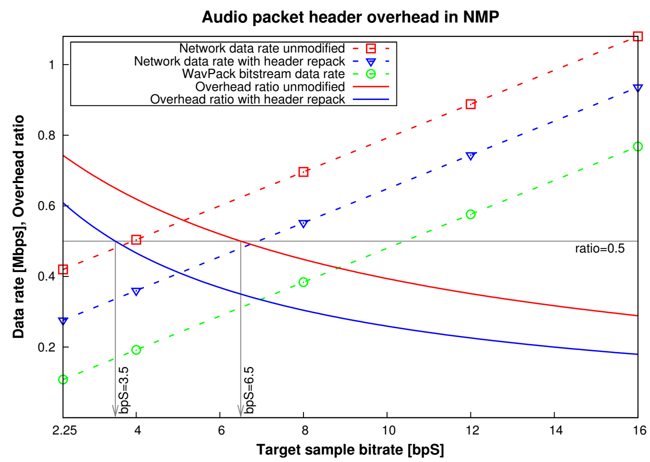

Fig. 3. Overhead ratio in WavPack-RTP streams

4.1. Data reduction

The amount of data that has to be transmitted between NMP clients and server differs from the output of the WavPack encoder. As described in section 3, even with our improvement gained by the header repacking, a block header of 16 bytes per audio packet remains. Furthermore, the employed network protocols add headers to each audio packet, resulting in an additional increase of 40 bytes (20 for IP, 8 for UDP, 12 for RTP) per network packet. With our default audio format (48 kHz, 16 bpS, mono) and a block size of 128 samples, we need to transmit 375 packets each second, resulting in an overhead data rate of 168 kbps that must be added to the compressed audio bit-stream. Without our modification we would have had to add 312 kbps.

This relation is depicted in figure 3, where the lower dotted line shows the compressed bitstream data rate returned by WavPack corresponding to the chosen target sample bitrate. The middle line is translated parallel to the y-axis by 0.168 Mbps and shows what needs to be transmitted over the network in NMP, while the upper line corresponds to the network load with unmodified WavPack headers. The overhead to payload ratio curves show how the compression efficiency gained by lower target bit-rates in WavPack is impaired by the overhead. We see that at the lowest target bit-rate of 2.25 bpS with the modified version of WavPack, more than 60% of a network packet contains header data; with the original version it is up to 75%.

The ratio becomes 50% at 3.5 bpS with the modified version and at 6.5 bpS with the original one. Using higher compression means we transmit more header than audio data. A target sample bit-rate of 4 bpS is therefore a good compromise for efficient data-reduction and a tolerable payload-to-overhead ratio, resulting in a network data rate of 360 kbps, while the maximum compression ratio of 2.25 bpS would reduce the network load to 276 kbps.

4.2. Audio quality

Now that we have seen that WavPack meets the requirements for data reduction, we evaluate the audio quality with a comparison to ADPCM, our previous codec with the same real-time capability and comparable compression ratio of 4 bpS.

We performed our evaluation with the ODG (objective difference grade) measure as proposed by the ITU for the widely accepted PEAQ [6] method. The scale for ODG ranges from 0 to -4, where 0 means imperceptible degradations and -4 implies very high distortion.

Figure 4 shows a snapshot of our results for an audio track of 400 seconds duration that includes most common music types. It shows the ODG results for the ADPCM codec at 4 bpS and that of WavPack with target bit-rates of 4, 3, and 2.25 bpS. We see that most of the time WavPack even in the highest compression mode performs better than ADPCM. ADPCM drops below -3.0 several times, while WavPack-2.25 always stays above -2.5. With a target bitrate of 3 bpS WavPack is already fully above -1.0, corresponding to very good audio quality. Lowering compression ratio to 4 bpS allows WavPack to produce nearly perfect audio quality. It far outperforms ADPCM and provides essentially transparent audio quality.

As result we can state that WavPack not only meets our demand for real-time processing but also provides high compression ratios and maintains very high audio quality for audio formats and data rates typically used in NMP.

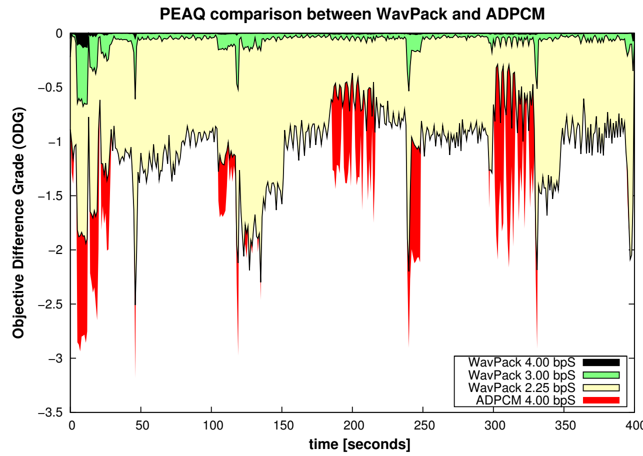

Fig. 4. Quality comparison of WavPack and ADPCM

5. CONCLUSION AND OUTLOOK

The strict demand for low delays in NMP limits the applicability of current audio codecs, that must operate with an audio block length of 2.66 ms and maintain good audio quality. Neither today's established music nor voice codecs meet these requirements. Possible alternatives like Fraunhofer's ultra low delay audio codec ULD [7] address the problem of compressing real-time audio with high quality, but even the provided minimal delay of 5.3 ms is too high and not acceptable for our application.

From today's available audio codecs we chose to adopt WavPack for our NMP system. It complies with the requirements and with minor modifications it works quite efficiently with audio block sizes of 128 samples and no additional buffer delay. The audio quality evaluation showed that WavPack is nearly transparent at 4 bpS and significantly outperforms ADPCM at all bitrates, including the highest supported compression ratio of 2.25 bpS.

In NMP by principle a large portion of the network load is caused by protocol and codec overhead. With higher compression ratios the resulting absolute data reduction of the bitstream leads to smaller relative decreases of the network load. This limits the potential for the future use of alternative audio codecs. Therefore we think that WavPack will remain the ideal audio codec for our NMP system in the long run.

6. REFERENCES

[1] N.P. Lago and F. Kon, "The quest for low latency," *Proceedings of the International Computer Music Conference*, pp. 33–36, 2004.

[2] Z. Kurtisi, X. Gu, and L. Wolf, "Enabling network-centric music performance in wide area networks," *Communications of the ACM, Special Issue on Entertainment Networking: Recreational Use of IP Networks*, vol. 49, no. 11, pp. 52–54, november 2006.

[3] M. Lutzky, G. Schuller, M. Gayer, U. Kramer, and S. Wabnik, "A guideline to audio codec delay," *AES 116th convention, Berlin, Germany*, pp. 8–11, 2004.

[4] X. Gu, M. Dick, Z. Kurtisi, U. Noyer, and L. Wolf, "Network-centric Music Performance: Practice and Experiments," *IEEE Communications Magazine*, vol. 43, no. 6, pp. 86–93, June 2005.

[5] David Salomon, *Data Compression: The Complete Reference*, Springer-Verlag, Berlin, Germany, 2007, With contributions by Giovanni Motta and David Bryant.

[6] International Telecommunication Union, "Method for Objective Measurements of Perceived Audio Quality," Rec. BS.1387, ITU-R, November 2001.

[7] U. Kramer, G. Schuller, S. Wabnik, J. Klier, and J. Hirschfeld, "Ultra Low Delay audio coding with constant bit rate," *117th AES Convention*, 2004.