“Applying Video to Low Delayed Audio Streams In Bandwidth Limited Networks”

Alexander Carôt[1], Gerald Schuller[2]

[1] *University of Applied Sciences, Anhalt, Germany*

[2] *Fraunhofer IDMT, Ilmenau, Germany*

Correspondence should be addressed to Alexander Carôt (`alexander.carot@inf.hs-anhalt.de`)

ABSTRACT

After having started as a pure Internet2 broadband service current latency optimized hard- and software perform well with conventional DSL networks, which allows this principle and the respective technology to be distributed even globally. Supporting narrow-band networks, however, leads to significant and often unforeseen problems in terms of traffic engineering especially when additionally considering conventional Internet usage on such endpoint connections. This paper first gives a general description of the problem, then it looks at the special case of video streaming and finally presents a solution solving these problems by providing an interleaved streaming scheme of low-delayed audio- and video data.

1. INTRODUCTION AND HISTORY

Due to the conventional best-effort manner [15] without any guarantee of packet delivery, the Internet was not originally developed for the purpose of sending real time traffic like audio data. Packet loss and delay variations typically lead to dropouts in the received audio stream which result in signal errors, and correspondingly, disturbing clicks and noise cracks [14]. Recent Voice-over-IP (VoIP) [13] and video conferencing services overcome these problems by applying large audio frames, large network buffers, and the principle of packet retransmissions. Consequently this results in additional latencies of several hundred milliseconds. Such delays do not represent a problematic figure in the context of voice communication: As part of the evolving globalization process and its distributed work processes realtime telecommunication on the Internet has nowadays become a widely accepted and commonly used service [3].

In the artistic world, however, these services cannot be used due to different quality and latency requirements: While phone conversations with an audio quality of 8 kHz and a latency of 400 ms can still be considered acceptable [9], a realistic musical interaction requires transparent audio quality and latencies far below 50 ms. In fact – especially when the performance speed is high – certain kinds of music cannot be performed with delays beyond 5 ms [5]. Nevertheless, a latency threshold of 25 ms generally leads to an acceptable and performable situation for the majority of musicians and music styles [3].

In order to achieve adequate realistic performance conditions one mainly needs to consider the network relevant aspects of bit rate and delay variations (generally known as “jitter”) [15]. For instance, an uncompressed one-channel 48 kHz audio stream with 16 bits/sample requires a channel having a minimum available bit rate of at least 768 kbps. The delay variations should not exceed values beyond 2 ms in order to avoid the previously mentioned large network buffers and the corresponding large delays. It is commonly known, however, that these requirements are often not met, especially in the context of DSL lines.

Starting in the year 2000 Chafe et al [7] took a different approach by consciously choosing Internet2 connections with their corresponding high bandwidth and low jitter in order to evaluate network music performances. Since then numerous distributed music sessions have successfully been performed as part of the Jacktrip project [2] between Internet2-connected locations all over the world. In that context they were able to prove that – despite its asynchronous transmission characteristics – the Internet is able to transport audio data in high quality and with low latency.

Nevertheless, it had so far not been possible to evaluate this effect for narrow band networks such as A-DSL with its low upload limitation of about 500 kbps for a 6 Mbps

DSL line. Due to that reason we developed the Soundjack project with appropriate low delay audio compression algorithms: Especially by using the ULD [12] or CELT codec [16] it is possible to transmit a high quality audio stream with a compressed payload of only 96 kbps. In that context we could show that even a conventional A-DSL endpoint connection fulfills the quality and delay requirements for distributed music performances [4]. However, it turned out that narrow band networks require explicit usage of the endpoint due to the jitter problem: While the backbone network structure with its high bandwidths of several Gbps introduces relatively low delay variations, the low upload bandwidth of typically less than one Mbps generates significant jitter with every additional packet sent in parallel to the audio stream. According to the so-called MTU (maximum transfer unit) of 1.500 bytes [15] the largest packet size is 1.500 byte and the corresponding transmission time can be calculated by equation 1, in which $\mathrm{d_t}$ represents the transmission time or delay and $\mathrm{b_c}$ represents the capacity of the link.

$$\mathrm{d_t} = \frac{1.500 \text{ bytes}}{\mathrm{b_c}} \qquad (1)$$

According to the previous upload bandwidth example of 500 kbps the corresponding delay would result in 26 ms for one single 1.500 byte packet. In turn a simultaneous audio stream would suffer from a 26 ms time gap due to the busy line. Assuming an audio packet interval of 2.7 ms this would already lead to almost 10 lost packets. The only remedy here would be the application of a jitter buffer on the receiving end. However, this adds an additional latency of at least 26 ms to the playback delay and hence represents an unacceptable solution. As a result one simply has to avoid any kind of cross traffic at the endpoint when running a distributed music session with narrow band Internet connections [3].

2. PROBLEM

There is no doubt that the domain of distributed music represents a very special application and hence can be considered acceptable having musicians consciously using their endpoint connection in the desired manner. However, one significant problem remains unsolved: Although pure audible communication without visual contact is sufficient for most musicians, an additional video stream can also be of interest – at least in order to give visual cues or further information just as it is the case in real music performance. Nevertheless, video data also represents a cross-traffic stream, which introduces the same jitter on the endpoint connection and also disrupts the low delay audio stream in the same way as stated above. As a consequence in that kind of scenario applying video is not possible with conventional technologies.

3. GOAL AND APPROACH

Our goal is to enable audio *and* video communications over the same DSL line, with still acceptable delays for our application. As described above the jitter problem occurs with every single cross traffic packet. In fact there are several approaches of reducing the packet size and in turn minimizing the jitter (e.g. by setting a lower Maximum Transfer Unit, MTU) but this implies significant drawbacks: Lower packet sizes imply a higher number of packets and in turn an increasing packet overhead [15]. This leads to an increased total bit rate, but it also does not necessarily lead to an improved situation: Decreasing the MTU by 50 % indeed results in maximal packet sizes of only 750 bytes. However, they still generate an unacceptable jitter of 13 ms on a 500 kbps network link. As a consequence we consider the avoidance of any form of cross-traffic on narrow band endpoints as the primary goal.

In order to achieve this goal we use a combination of audio- and video data into a single byte stream. Hence each stream packet contains both types of data, and they are transmitted together and do not block each other anymore. This principle generally refers to the term "interleaving" [8].

4. CONCEPT

Table 1 shows a typical parameter configuration of a distributed music session. With the previously described goal the audio packet interval of 2.7 ms, we reach a packet size of 32 bytes for the compressed audio data and a network buffer of one single packet. We use this configuration as a reference.

In terms of timing and musical interaction audio represents the reference medium in distributed music sessions [6]. Video can be a meaningful add-on but not a mandatory feature. It also does not necessarily require realtime processing, which would need significant system and network resources. Hence, applications can theoretically range from very low to very high frame and color resolution for either simple visual cueing in private music rehearsals or high quality image presentations

Samplerate	48 kHz
Framesize	128 Samples
Sample bitdepth	16 bit
Audio channels	1
Audio packet size	256 Bytes
Compressed packet size	32 Bytes
Audio packet interval	2.7 ms
Network buffersize	1

Table 1: Typical low-delay audio streaming settings

Resolution (pixel)	e.g. 320*200
Pixel bitdepth	2 - 32
Compressed packet size	codec dependent
Framerate (fps)	1 - 50
Video packet interval (ms)	20 - 1000

Table 2: Video stream settings

in public performances. In that context also the applied compression codec is of interest: With complex and efficient inter-frame compression techniques modern HD-TV codecs such as H.264 [10] deliver excellent compression rates by still maintaining very good video quality. However, due to their inter-frame processing techniques such codecs always require the analysis of a group of images, which introduces additional latency and prevents a frame-by-frame compression principle. In contrast older techniques such as JPG [11] or Motion JPEG use the frame-by-frame principle, in which each image is coded instantly and independently of one another. Assuming an equal compression ratio, however, the quality cannot compete with the previously described codec.

In any case it is finally the available Internet endpoint bandwidth which determines the upper limit regarding image resolution, framerate, bit depth. As a consequence the typical parameter settings vary significantly as shown in table 2.

So far conventional conferencing systems send both streams simultaneously but independently. In contrast our concept leaves the audio stream as described and applies a modification of the video packets: Once the sender has generated a compressed video packet we split this packet into a number of smaller chunks. These chunks are marked with a number and are added to the audio stream packets. Once the final chunk has arrived, the receiver reconstructs the original compressed video packet from the smaller chunks, from which the final image can be decoded. Maintaining a packet size below the MTU guarantees one single stream with audio packet interval T_{audio} without any additional cross-traffic packets and the resulting jitter. The only drawback is an additional video delay of one video packet. We found this acceptable in our application because the low latency of the audio stream is more important than that of the video stream. Figure 1 graphically illustrates this concept.

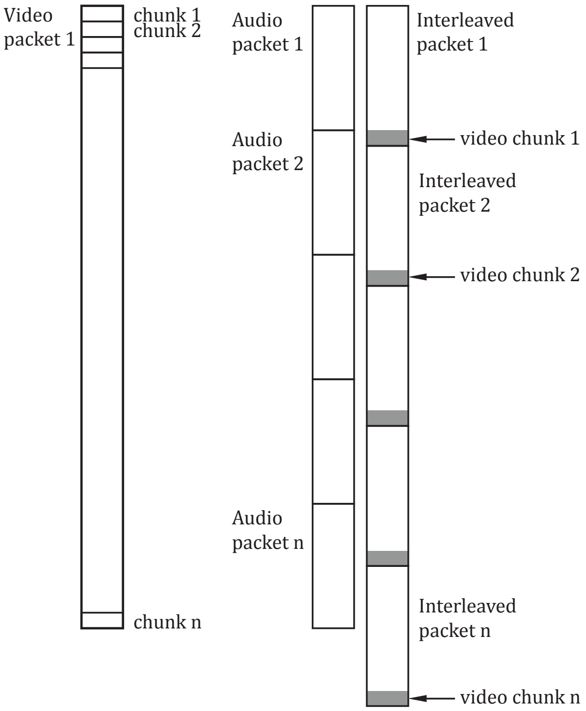

Fig. 1: Concept of interleaved audio/video transmission

The number of video chunks n depends on the audio packet interval T_{audio} and the video packet interval T_{video} with reference to $n_{\text{video}} = T_{\text{video}}/T_{\text{audio}}$. Regarding the reference audio stream setting with $T_{\text{audio}} = 2.7$ ms each frame of a four-frame-per-second video is split into 371 chunks.

5. IMPLEMENTATION

Regarding a first prototype implementation, ease of use and the lowest possible latencies we choose a common

samples/block	n_{chunks}	bytes/chunk
64	742	7
128	371	14
256	186	28
512	93	56
1024	47	112

Table 3: Resulting chunk numbers and sizes

video specification with a resolution of 320 * 200 pixels, 24 bit color resolution and the JPG compression codec, which generates packets frame-by-frame. In our test scenario this results in a variable bit rate (VBR) stream with compressed packet sizes of 3.000 bytes to 4.000 bytes. In contrast the audio stream represents a constant-bit-rate (CBR) data stream. Since the described approach assumes a CBR data stream with equal-sized packets we define a constant video packet size of 5.000 bytes and add padding bytes. Although this increases the payload, it guarantees a CBR stream. Finally, the 5.000 bytes are divided by the previously calculated number of audio chunks in order to retrieve the size of the chunk, which is then added to each processed audio packet. For the reference stream with 371 chunks this results in 14 bytes per chunk. To complete the picture table 3 shows the resulting video chunk sizes for each audio stream configuration.

6. EVALUATION

In order to provide a realistic and representative evaluation of this principle we decided to use a conventional A-DSL endpoint of 900 kbps upload, which is also used for regular music rehearsals. The download had a capacity of 10 Mbps.

We implemented a function into the software which switches between our interleaved video packet transmission and a conventional non-interleaved method with two data streams. In both cases the data was sent to a remote PC at a distance of approximately 500 km, which behaved as a network mirror that reflected every incoming packet back to the sender. In that way it was possible to artificially create and individually adjust an incoming data stream. This principle is illustrated in figure 2. The audio packets had 512 samples each. In terms of network music sessions and especially in comparison with the previously described low-delay reference settings this value is relatively high. However, we applied it in order to prove that the cross traffic problem even occurs in less delay critical settings. The network buffer was set to one single audio packet with a buffering time of 10.8 ms.

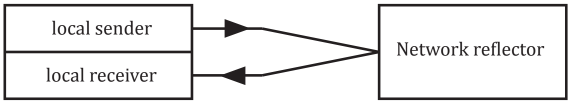

Fig. 2: Network reflector setup

The total measurement session length was 30 seconds. Every five seconds the software automatically switched between the interleaved and the non-interleaved mode, while the audio dropouts were measured on the receiving end. Figure 3 shows the packet roundtrip times for each single packet in the upper graph and the audio dropouts in the lower graph. A black spike indicates an audio dropout. In case of an uninterupted playback the graph remains white.

In the first five seconds the interleaved mode was applied. Here the average RTT was 40 ms and since the maximum jitter did not exceed four milliseconds not a single audio dropout could be measured. The delay variation was compensated by the 512-sample network buffer. However, within the next five seconds the non-interleaved mode was turned on and the RTT increased to values of between 55 ms and 70 ms in a periodic manner with every video packet sent. As a consequence the audio packets could not be delivered within the required time span, which in turn caused periodic dropouts at approximately every 250 ms. In the remaining 20 seconds the process again switched between the interleaved and non-interleaved mode and the network behavior just described could be observed.

The presented solution could not be compared with existing remote music applications such as Jacktrip [2] or eJamming [1] as they do not support video streaming. It could not be compared with competing videoconferencing systems either because none of them support low-delay audio streaming for distributed music.

7. CONCLUSION

For a delay critically audio communications scenario, like musicians playing together remotely, we showed the negative impact of network jitter on a low-delay audio

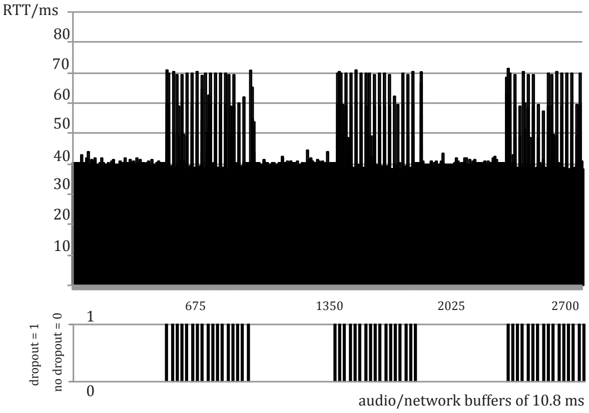

Fig. 3: Measurement results

connection by generating cross traffic on a narrow-band network. In order to overcome this problem we presented a solution which splits each video packet into a number of equally-sized smaller fragments and then adds them to the audio stream packets. Upon receiving of the final fragment the original video packet is reassembled, the content decompressed and displayed on the screen. In that way both streams are interleaved and cannot block one another anymore. In an experiment with a network mirror we showed that our approach effectively avoids packet dropouts caused by too much jitter in the transmission.

8. REFERENCES

[1] ejamming - website, August 2011. http://www.ejamming.com.

[2] Juan-Pablo Caceres and Chris Chafe. Jacktrip: Under the hood of an engine for network audio. In *Proceedings of the International Computer Music Conference 2009*, Montreal, Canada, August 2009.

[3] Alexander Carôt. *Musical Telepresence – A Comprehensive Analysis Towards New Cognitive and Technical Approaches*. PhD thesis, Institute of Telematics – University of Lübeck, Germany, 2009.

[4] Alexander Carôt, Ulrich Krämer, and Gerald Schuller. Network music performance in narrow band networks. In *Proceedings of the 120th AES convention*, Paris, France, May 2006.

[5] Alexander Carôt, Christian Werner, and Timo Fischinger. Towards a comprehensive cognitive analysis of delay-influenced rhythmical interaction. In *Proceedings of the International Computer Music Conference (ICMC) 2009*, Montreal, Canada, August 2009.

[6] Chris Chafe, Michael Gurevich, Grace Leslie, and Sean Tyan. Effect of time delay on ensemble accuracy. In *Proceedings of the International Symposium on Musical Acoustics*, Nara, Japan, March 2004.

[7] Chris Chafe, Scott Wilson, Randal Leistikow, David Chisholm, and Gary Scavone. A simplified approach to high quality music and sound over ip. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, December 2000.

[8] Douglas E. Comer. *Computernetzwerke und Internets mit Internet-Anwendungen*. Pearson Studium, third edition, 2002.

[9] International Telecommunication Union (ITU). *Recommendation H.323: Audiovisual and multimedia systems – Infrastructure of audiovisual services - Systems and terminal equipment for audiovisual services – Packet-based multimedia communications systems*, 1998.

[10] International Telecommunication Union (ITU-T). *Recommendation H.264: Advanced video coding for generic audiovisual services*, 2007.

[11] International Telegraph and Telephone Consultative Committee (CCITT). *Recommendation T.81: Information Technology – Digital Compression And Coding of Continuous-Tone Still Images – Requirements and Guidelines*, 1992.

[12] Ulrich Krämer, Jens Hirschfeld, Gerald Schuller, Stefan Wabnik, Alexander Carôt, and Christian Werner. Network music performance with ultra-low-delay audio coding under unreliable network conditions. In *Proceedings of the 123rd AES-Convention*, New York, USA, October 2007.

[13] Sivannarayana Nagireddi. *VoIP Voice and Fax Signal Processing*. Wiley, first edition, 2008.

[14] Ken C. Pohlmann. *Principles of Digital Audio*. The Mcgraw-Hill Companies, fifth edition, 2005.

[15] Andrew S. Tanenbaum. *Computer Networks*. Pearson Studium, fourth edition, 2003.

[16] Jean Marc Valin, Timothy Terriberry, Christopher Montgomery, and Gregory Maxwell. A high-quality speech and audio codec with less than 10 ms delay. *IEEE Transactions on Audio, Speech and Language Processing*, 2009.