Netjack – Remote music collaboration with electronic sequencers on the Internet

Alexander Carôt
University of Lübeck
Institute of Telematics
Ratzeburger Allee 160
23538 Lübeck,
Germany,
carot@itm.uni-luebeck.de

Torben Hohn
Deutsche Telekom AG
Laboratories
Ernst-Reuter-Platz 7
D-10587 Berlin
Germany
torbenh@gmx.de

Christian Werner
University of Lübeck
Institute of Telematics
Ratzeburger Allee 160
23538 Lübeck,
Germany,
werner@itm.uni-luebeck.de

Abstract

The JACK audio server with its ability to process audio streams of numerous applications with realtime priority, has major significance in context with audio processing on Linux driven personal computers. Although the Soundjack and the Jacktrip project already use JACK in terms of remote handmade music collaboration, there is currently no technology available, which supports the interconnection of electronic music sequencers. This paper introduces the Netjack tool, which achieves sample accurate timeline synchronization by applying the delayed feedback approach (DFA) and in turn represents the first solution towards this goal.

Keywords

music, networks, jack, realtime, latency

1 Introduction

Latency always has a major significance regarding a musical interplay. The speed of sound of about 340 m/s results in signal delays depending on the physical distance between rehearsing musicians. Hence, two musicians‘s beats can never occur in precise synchrony. However, such delay offsets represent a natural playing condition and musicians unconsciously cope with them. Nevertheless, if due to a large physical distance the delay offset exceeds a certain value, the musical interplay becomes impossible since the other musicians‘s pulses are perceived as “out of time”. This latency threshold depends on several factors such as the speed of a song, the note resolution and the musician‘s rhythmical attitude. Nevertheless, we can state, that the delay between two musicians may not exceed a value of 25 ms [2]. This corresponds to a physical distance of approximately 8.5 m.

Due to the fact that in the electronic domain signals are transmitted with speeds in extremely higher dimensions, the Soundjack [2] as well as the Jacktrip [5] software aim at achieving delay conditions in the Internet, which match such of a conventional room in order to provide remote network music performances for displaced musicians. This principle has successfully been evaluated with professional musicians displaced by more than 1000 km. However, we as well used these low-delay audio links for the interconnection of electronic music devices such as drum computers and sequencers, and in this context we discovered significant problems, which will be described in the following section.

2 Problem

Nowadays electronic music gear theoretically enables any home user to run an own home studio production. In that context modern sequencers and recording software such as “Ardour”, “Cubase” or “Logic” represent generally accepted and often applied software tools. Each of them works with a strict sample based resolution as the theoretical time reference for musical sequences. Such sequences of a multitrack recording are generally recorded subsequently and one typically expects musical events to occur on the precise beat reference. This principle differs from the previously introduced handmade music scenario, where musicians can perform with slight delays without either side noticing it: In case two displaced musicians want to run a sequence based music production together, any delay would result in unacceptable effects. Assuming a low delay audio streaming link, the slight signal delays would allow conventional musicians to perform in a convenient manner but it would result in undesired time gaps between the sample-bound tracks of the electronic devices. This effect is most obvious if two remote drum machines are supposed to be merged to a single groove: Depending on the actual amount of delay the arrival of late drum notes automatically leads to a disturbing chorus or echo effect on either side.

3 Concept

In order to prevent the effect of latency we apply the so called delayed feedback approach (DFA) [3], which was formerly used in distributed music performances, which suffered from too large delays beyond 25 ms. DFA tries to make musicians feel like playing with delays below this delay threshold by delaying one player's signal artificially: If one player mutes his own local signal and instead listens to his feedback caused by microphones and speakers of the remote side, the musical interplay happens in perfect synchronization. The disadvantage, however, lies in the fact that one player has to play in advance in order to compensate his own delayed feedback. The DFA principle is illustrated in figure 1: It takes the one-way delay (OWD) to send a signal from player A to player B. B receives the signal and can play with it. Since player B works with loudspeakers and a microphone, a mix of signal A and signal B is sent back to A. This transmission again takes the OWD. This leads to the desired synchronization of both signals but also to a playback delay of $2 \cdot \text{OWD}$ for signal A, which is equal to the roundtrip time (RTT). Since A uses headphones instead of a loudspeaker, the described feedback loop does not occur on this side. Rather than working with a real feedback loop the same effect can be reached by artificially delaying one side's signal locally with the roundtrip time.

Though DFA improves the delay situation between two musicians, it is no doubt that a delay of one's own signal typically can be considered as inconvenient and not natural. The larger the delay gets and the louder the instrument's direct noise, the more disturbing the overall playing conditions become due to the delayed signal. This is especially valid for any acoustic instrument such as a violin or a drum set. However, while investigating in DFA it already became clear to the authors that DFA can be a suitable approach for the synchronization of remote playback sound sources: In case of e.g. two DJ's turntables are connected with each other, a delay of the turntable's output would not lead to timing-problems. Unlike human beings a machine's playback behavior does not depend on an inner time or feel and hence can of course reproduce delayed sounds without loosing any kind of rhythm. Hence, DFA represents the ideal approach for the synchronization of remote music sequencers. The following section will describe our realization of this DFA based streaming system based on the open source Jack audio server technology [7; 1].

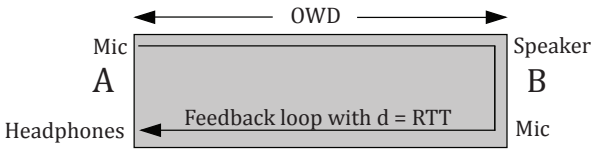

Figure 1: Delayed feedback approach (DFA)

4 Realization

Like traditional VoIP and distributed music applications the current work falls under the domain of realtime traffic on the Internet. Generally two computer's soundcards have to be connected in such a way that one's card's input is fed to the other card's output and vice verse. The program reads blocks of samples from the soundcard and packages them into UDP datagrams, which are sent to the remote destination to be played back by its respective sound device.

However, IP packets underly the effect of network jitter [9] which prevents a solid stream playback. Furthermore, packets can get lost, or reordered and hence the program must be prepared for this to happen. This is typically realized in form of a jitter buffer, which buffers a desired number of packets and possibly reorders them before the soundcard reads from it.

Each outgoing packet carries a timestamp. Upon reception, they are put into the jitterbuffer at the position corresponding to their timestamp. The draining of the jitterbuffer is driven by the local soundcard's clock. Apart from this general functionality of realtime traffic transportation our Netjack system provides further features, which have significance in terms of the approached remote music collaboration of synchronized beat devices.

4.1 Sample accurate timeline synchronization

It is well known, that due to wordclock drift two soundcards do not run in synchronization, unless they are synced via a common wordclock. Professional audio gear in a local network provides this synchronization by connecting all devices to the same wordclock. In that context one central's wordclock signal is fed via a cable connection into the external synchronization input of the respective soundcard. In a distributed sound system, however, this solution cannot be applied and leads to audio dropouts due to

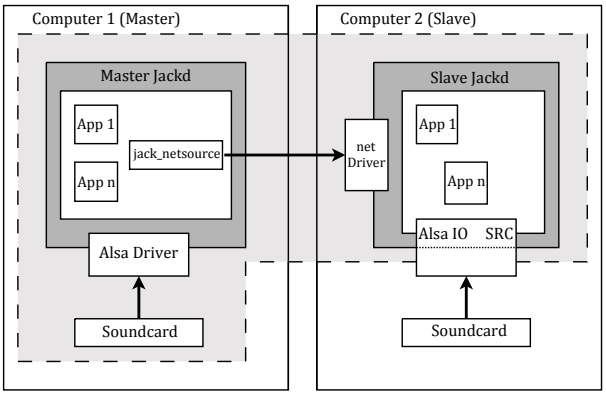

Figure 2: Principle of the netjack system

buffer under- and overruns in certain intervals depending on the amount of time drift between two remote wordclocks. In contradiction to the Jacktrip and Soundjack projects Netjack explicitly tries to prevent such dropouts and hence approaches to achieve sample accurate transport synchronization. Firstly one machine is assigned with the master role and another machine with the slave role. Once the master has established a connection to the slave it sends its wordclock via time stamps with the respective audio stream. Secondly the slave machine extracts the clock from the datastream and clocks its jack server with that speed. This way the receiving Jack slave is synchronized with the master so that way all signal processing is operating at a single clock, removing the necessity to compensate for clock drift.

In order to establish the bidirectional communication link the slave as well sends its data to the master. This, however, requires additional processing: Although the jack slave runs synchronized with the master‘s clock it now suffers from a clockdrift with the slave‘s soundcard. Thus our solution works with a second program, which applies a sample rate conversion (SRC) to the received audio buffers in order to match the slave‘s wordclock. After the SRC the slave‘s soundcard can play back the received buffers and can send its local buffers to the master, which waits with its local playback until the slave‘s first buffer has arrived. This general Netjack principle is illustrated in figure 2.

4.2 Bandwidth limitations and jitter compensation

Audio data sampled with 48 kHz and a resolution of 32 bit corresponds to data rate approximately 1.54 Mbps. This amount of data can be transmitted in local area networks (LAN), which nowadays hold bandwidth capacities of at least 10 Mbps up to several Gbps. In home consumer DSL networks, however, such capacities range in significantly lower dimensions. The upload capacity generally resides below 500 kbps and in turn requires a data reduction by encoding the respective audio stream [2]. Hence, we had to assign the requirements of remotely synchronized beat devices to an audio codec. The ideal low delay audio compression codec would exhibit the following properties:

- maximal quality
- constant coding latency
- constant compression ratio
- packet loss concealment
- float samples
- free license

Due to our experience with the Soundjack system [4] we figured that the open source CELT codec [10] currently represents the only compression technology, which meets these requirements. Regarding the actual implementation of CELT into Netjack the complex part is to make the code retain sync under packet loss conditions.

4.2.1 Packet deadline

The netjack slave has an internal deadline for packet reception, which is calibrated as long as packets are flowing. When the packet with the required sequence number has not been received within that deadline, it is considered lost. In this case the implemented CELT codec applies a packet loss concealment, in order to mask the lost data as effectively as possible.

The calibration works like this: Each received packet gets a timestamp t_r when it is received at the slave. With the reply the difference between the deadline and the receipt timestamp $t_d - t_r$ is sent back to the Master. The Master in turn subtracts this value from the difference between the timestamps of receipt t_m and consumption t_c. The result t_{late} is an approximation, of the lateness of the “reply” to a lost packet at the Master as described in equation 1.

$$t_{late} = (t_m - t_c) - (t_d - t_r) \qquad (1)$$

Currently the slave gradually adjusts the deadline t_d so that t_{late} is one eigth of the total roundtrip latency. This value is quite arbitrary,

but it works sufficiently for low as well as high roundtrip delays.

5 Proof of concept

Due to our experience in context with low delay audio streams on the Internet we assumed that with the current Netjack implementation an acceptable audio transmission could be achieved [2]. In turn we did not consider it as useful to measure audio dropouts and instead decided to perform a real online session between an A-DSL endpoint in Lübeck/Germany and an A-DSL endpoint in Berlin/Germany. Both endpoints included a wireless link. This session was performed on Tuesday, 13th of January: We ran the jack-server in Berlin, the client in Lübeck and connected them via the jack-netsource command. In turn Lübeck became the "master" and Berlin the "slave". Then both sides opened the open source sequencer "Hydrogen" [6] and connected the jack ports respectively to the application. Additionally we connected each side's system capture device to the stream in order to use this setup as a voice communication link. First, we verbally discussed the upcoming musical experiment for about 10 minutes, in which we observed a stable network connection, which casually suffered from a few audio dropouts. Depending on the actual amount these dropouts were more or less audible, however, they did not lead to an unacceptable and disturbing situation. After the agreement on specific sounds and styles we started with the composition of a musical pattern loop, which mainly consisted of various drum sounds. In this process we could clearly see that both sides ran in precise synchronization. Whenever either side added sound events at specific note values, the notes were played at precisely the same instant on the remote side. In terms of session control the master in Lübeck was in charge of starting and stopping the playback of the loop sequence. Each time the master performed these commands, the slave side followed respectively. The overall performance lasted for 30 minutes and in terms of network stability or overall quality exhibited similar results as the previous voice chat.

6 Conclusions and future work

The current implementation of our Netjack system is able to achieve a situation, in which two remote electronic musicians can perform as if they were sitting side by side in front of the same computer. Due to the precise and accurate Jack timeline synchronization on the endpoint machines in combination with audio transmission based on the delayed feedback approach (DFA) even network delays of intercontinental dimensions can be compensated effectively in such a way that the user does not take note of it. Practically the master device first sends its data to the slave machine, which adjusts its transport according to the roundtrip delay and then sends its data to the master, who does not start with its local playout process before having received it. In fact the only moment a user could notice a delay would be in the startup process – after the start button has been pressed until the first sound occurs – however, even in an international setup roundtrip delays below 200 ms have become a common value and in turn do not represent a problematic number. In terms of audio quality the implemented CELT codec can be adjusted depending on the available network capacity and achieves even at 48 kbps decent results, which allows Netjack to be used with almost any conventional A-DSL connection. Nevertheless – as usual in asynchronous networks – the transmission suffers from casual audio dropouts depending on the amount of network jitter and loss rate. As CELT already provides a packet loss concealment, these dropouts are less disturbing but still noticeable. Hence, in the future we will investigate in better and more efficient algorithms related to high quality audio concealment. Furthermore, we will improve the system in terms of a better usability: In the current implementation the slave e.g. has no opportunity to start or stop the playback and depends on the master's actions. Hence, in the future we approach to achieve a precise mirror of each performer's actions in order run the master/slave relation just as a technical background task but abandon it in terms of the musical interaction between the players. However, the general functionality of the actual Netjack implementation still suffers from a few drawbacks: Due to the complex Netjack principle with the appropriate amount of control and timing data, an actual audio packet introduces a large packet overhead, which currently almost equals the amount of audio data. In terms of efficient bandwidth capacity utilization we will approach to reduce this overhead by identifying and respectively reducing information redundancy. Moreover, the current implementation is yet not compatible with the multiprocessor Jack2 technology [8] but since both systems are

supposed to provide interoperability, our future implementation of Netjack will take this issue into account.

7 Acknowledgements

We would like to thank Paul Davis and the whole Linux audio community for providing us with exceptional support and constantly reminding us of the initial motivation of achieving high-quality audio production and collaboration tools for anyone.

References

[1] Fons Adriaensen. Using a dll to filter time. In *Proceedings of the Linux audio conference 2005*, 2005.

[2] 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.

[3] Alexander Carôt and Christian Werner. Network music performance – problems, approaches and perspectives. In *Proceedings of the "Music in the Global Village" - Conference*, Budapest,Hungary, September 2007.

[4] Alexander Carôt and Christian Werner. Prinzipien musikalischer telepräsenz. In *Tagungsband der 25. Deutschen Tonmeistertagung*, Leipzig, Germany, November 2008.

[5] C. Chafe, S. Wilson, R. Leistikow, D. Chisholm, and G. 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.

[6] Alessandro Cominu. Hydrogen website, January 2009. `http://www.hydrogen-music.org`.

[7] Paul Davis. Jack website, January 2009. `http://jackaudio.org`.

[8] Stephane Letz. Jack2 website, January 2009. `http://www.grame.fr/~letz/jackdmp.html`.

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

[10] J.M. Valin, T.B. Terriberry, C. Montgomery, and Gregory Maxwell. A high-quality speech and audio codec with less than 10 ms delay. `http://www.celt-codec.org/`.