

# A Multimedia Signal Processing Cloud Concept for Low Delay Audio and Video Streaming via the Public Internet

Christoph Kuhr\*, Alexander Carôt†

Department of Computer Sciences and Languages,  
Anhalt University of Applied Sciences, Köthen  
Email: \*christoph.kuhr@hs-anhalt.de, †alexander.carot@hs-anhalt.de

**Abstract**—A rehearsal environment for conducted orchestras via the public Internet requires a specialized server infrastructure, in order to provide minimal latencies between the musicians involved. In this document, we present a cloud computing concept for digital signal processing of audio and video data in realtime. Since 60 musicians and one conductor shall connect to the cloud, it is most important to distribute the signal processing and machine learning algorithms over multiple processing servers. The server infrastructure under investigation is built on top of an AVB network segment to be scalable and to maintain low latencies and jitter under heavy load. Latency and jitter are the most important properties of the realtime streams that are connected to the cloud, and are analyzed and discussed. The results have proven the proper design of the concept, but revealed the need for further optimization.

## I. INTRODUCTION

**S**OUNDJACK [1] is a realtime communication software that establishes up to five peer to peer connections via the public Internet. This software was designed from a musical point of view and first published in 2006 [2]. Playing live music via the public Internet is very sensitive to round trip as well as one-way latencies. Thus, the main goal of this application is the minimization of latencies and jitter, while limited by the speed of light. Participating musicians require some soft skills to tolerate the latencies none the less.

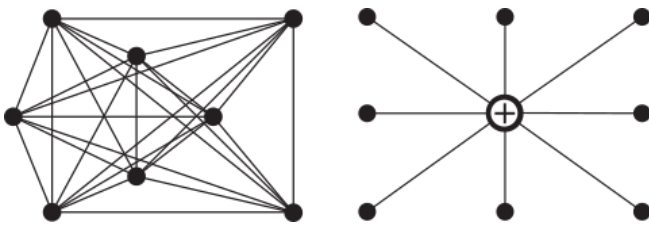


Figure 1. Left: Peer to Peer Network Topology, Right: Star Network Topology

### A. Soundjack and fast-music

The goal of the research project fast-music is to develop a rehearsal environment for conducted orchestras via the public Internet. Up to 60 musicians and one conductor, who are

randomly distributed throughout Germany, shall be able to play together live. The central node represents the multimedia signal processing cloud under investigation, which ideally is located in Frankfurt on the Main. First investigations of round trip times suggest Frankfurt on the Main as the logical center of Germany.

Further fields of research are the transmission of Wavelet based low delay live video streams and motion capturing of the conductor. The latter shall be displayed on a holographic LED cube [3] that was developed by our project partner Symonics GmbH [4].

### B. Motivation

The most important aspect for any further network and software design decision for the cloud concept is the application of digital signal processing algorithms to the audio and video streams. Examples for digital signal processing applications are audio error concealment due to UDP packetloss in the public Internet, based on a machine learning approach, or virtual room acoustics in the form of individual binaural rendered Ambisonics soundfields that simulate the musicians location inside an orchestra for a better immersion.

The second important requirement is the service time of Ethernet frames that are arriving on a serial network interface at the wide area network (WAN) side of the server cloud. In this document the network under investigation is the campus network of the university. Thus, Ethernet is also considered a WAN technology for the scope of this paper.

During the service time, no datagrams of any concurrent UDP streams can be received. Consequently any stream arriving on such a serial network interface experiences a latency, equal to accumulated latency of all streams arriving at this interface.

A scalable and extendable concept with multiple proxy servers that are connected to the same signal processing network segment is chosen over a single processing server. The Soundjack processing cloud has to be segmented accordingly. The approach that we chose is shown in fig. 2.

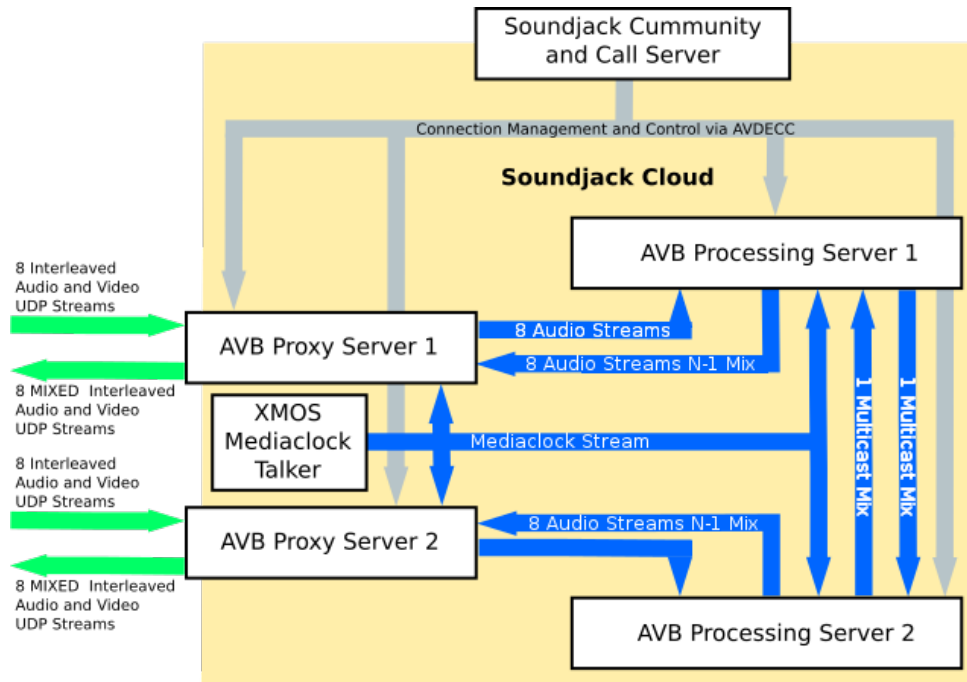


Figure 2. Soundjack Realtime Processing Cloud Concept

## II. SOUNDJACK REALTIME PROCESSING CLOUD CONCEPT

Audio Video Bridging / Time-Sensitive Networking (AVB/TSN) is a technology with the focus on audio and video streams in computer networks, that require realtime responsiveness. This technology is a set of IEEE 802.1 industry standards, which operate on OSI-Layer 2 [5]. These standards are:

- IEEE 802.1AS [6] - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks
- IEEE 802.1Qat [7] - Virtual Bridged Local Area Networks - Amendment 14: Stream Reservation Protocol (SRP)
- IEEE 802.1Qav [8] - Virtual Bridged Local Area Networks - Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams (FQTSS)
- IEEE 1722 [9] - IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks (AVTP)
- IEEE 1722.1 [10] - IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722 Based Devices (AVDECC)

These AVB standards extend the IEEE 802.1 standard family with precise synchronization, resource reservation and bandwidth shaping. Lower latencies and jitter, the avoidance of packet bursts and bandwidth shortage are addressed by these extensions, providing realtime responsiveness to a generic Ethernet computer network. These properties are used to ensure a constant bandwidth streaming with low latency and jitter inside the Soundjack realtime processing cloud. Thus, the

Soundjack client streams can be processed inside the cloud, without interfering with each other.

AVB networks require special hardware for timestamping Ethernet frames with separate transmission queues for each traffic class, i.e. AVB traffic with Stream Reservation (SR) classes A/B and generic Ethernet traffic. IEEE 802.1-2014 [11] defines the two distinct stream reservation (SR) classes A and B to differentiate audio and video traffic from other Ethernet traffic. SRP declares and registers resources on all switch ports along the path from the AVB talker to the AVB listener. This way, the resources to maintain an inter packet gap (IPG) of  $125 \mu s$  ( $250 \mu s$  for SR class B) are reserved.

The two AVB server types required for the Soundjack cloud are an AVB proxy server and an AVB processing server. The AVB proxy and processing servers are each connected to the same AVB LAN segment. The media transport in this design should not exceed a round trip time of  $4 ms$ , since AVB networks are constrained to a bounded one way latency of  $2 ms$  [9, p. 14]. Additionally, each server is connected to a non-AVB LAN segment. The Soundjack community and call server is also connected to this generic network segment and handles the session management of the different user sessions on the Internet and of the respective AVB server sessions. Since IEEE 1722.1 AVDECC traffic is not necessarily time-sensitive, it is sufficient to use a non-AVB LAN segment for command and control purposes. The Soundjack community and call server also provides community services of Soundjack.

### A. Mediaclock Server

All AVB servers subscribe to a mediaclock stream, which is supplied by an XMOS/Atterotech development board [12].

The mediaclock concept is based on the idea of the recovering from a clock stream, as it is proposed in [13].

### B. Common AVB Server Software Architecture

To meet the requirements of an AVB server software, it is not sufficient to write a multiprocessing and multithreading application in C. The hardware support for AVB requires a properly configured and optimized OS. The Linux kernel can be patched to operate in realtime mode [14], a Linux mainline kernel 4.8.6 was configured, patched with the corresponding realtime patch 4.8.6-rt5 and compiled.

The AVB talkers running on the system need to use hardware queues of the network interface to utilize the FQTS mechanism for enqueueing AVTP packets. A detailed description of the software architecture of the two different server configurations can be found in [15].

### C. AVB Proxy Server

In contrast to the public Internet where IP packets are forwarded on best effort, the Soundjack processing cloud provides a fully managed and controlled Ethernet network with AVB support. The traffic shaping property of AVB prevents the Soundjack cloud network from bursty traffic by means of a credit-based bandwidth shaper. The proxy server is used as a wave trap to break down large and erratic UDP datagrams into more and smaller, but constant AVTP packets. Thus, the stream packets can travel inside the Soundjack cloud network in a deterministic fashion, managed by the credit-based bandwidth shapers.

The AVB proxy server receives and transmits UDP streams from and to Soundjack Clients, that were assigned by the Soundjack session server. A maximum of eight streams is assigned to one AVB proxy server to keep the latency introduced by the service times low. The UDP streams received on the WAN interface are fragmented, because they need to be transmitted in the AVB LAN segment at another bitrate with a different payload. A UDP datagram of such a stream contains 256, 512 or 1024 Bytes of compressed or raw audio data. AVB implements its traffic shaping based on the idea of constant bitrate streaming. The resulting AVTP stream is sent from the AVB proxy server to the AVB processing server, which processes the eight streams and sends them back as AVTP with the same, but processed payload. Inside the Soundjack cloud a constant link capacity utilization per stream is of paramount importance to maintain the deterministic behavior of the AVB LAN segment. To achieve constant link capacity utilization, the payload of each UDP datagram needs to be properly fragmented into multiple AVTP packets. IEEE 802.1Qav [8, p. 44] requires an AVB talker to send an AVTP packet every  $125 \mu s$  for a class A stream and  $250 \mu s$  for a class B stream. This inter packet gap (IPG) is necessary to maintain a constant sample flow at 48kHz sample rate:  $6/48 kHz = 125 \mu s$ , and AVTP packet rate of  $48 kHz/6 = 8 kHz$ . A single stream may also contain multiple audio channels, thus an AVTP packet contains six samples per audio channel. Inside the Soundjack processing cloud however, all AVTP streams

contain two channels - a stereo stream. The proxy server talker instances mix the possible eight, four and two channels down to (or one channel up to) two channels per received UDP datagram, which reduces the payload size for the AVTP stream. The return stream of the Soundjack cloud has always 64 samples per UDP datagram for two audio channels - a stereo mix. The stereo mix from the samples of the UDP datagrams needs to be distributed over multiple AVTP packets. Soundjack uses a sample rate of 48 kHz exclusively and all channels are mixed down (or up) to two channels by the proxy server. The stereo mix channel count, the sample rate and the sample encoding determine the actual AVTP payload size of 48 bytes. Since AVTP packets arrive with the IPG of  $\Delta t = 125 \mu s$  at the proxy listener buffer, a constant latency is introduced by waiting for the correct amount of AVTP packets required to properly fill the UDP datagram.

Apart from the audio samples, the UDP stream also contains interleaved video data, which is described in [16].

### D. AVB Processing Server

The AVB processing server receives the audio and video streams, originating at the clients as AVTP streams with a constant packet rate of 8 kHz. It provides signal processing facilities for audio and video processing.

The JACK [17] audio server is deployed as infrastructure for the audio signal processing stage. It is a professional and open source audio server to share sample accurate audio data between different applications. A large number of signal processing applications and algorithms are available for JACK.

As off now, an audio multicast mixing application, to mix all streams that are connected to the Soundjack cloud, is deployed. A detailed description can be found in [18]. Further signal processing application, as mentioned in the introduction, are still under development.

## III. EVALUATION

A first evaluation was done with a single UDP audio stream that enters the Soundjack cloud via the WAN network interface of the AVB proxy server in the campus network. The proxy server forwards the stream as AVTP stream to the AVB processing server. After processing the audio signals, the processing server returns the AVTP stream to the AVB proxy server, which in turn constructs an UDP stream to return to the Soundjack client. The latency was measured with a scope connected to the digital-analog and analog-digital converters of the Soundjack clients audio interface.

Furthermore, the arrival timestamps of the UDP send and return stream and the different AVTP streams have been captured with the packet analyzer Wireshark. The probability density function of the difference to the previous timestamp of the AVTP streams is the bounded IPG of  $125 \mu s$  defined by AVB. In the context of the UDP streams, this probability density function gives a measure for the audio quality of the stream - whether it has high or low jitter.

#### IV. DISCUSSION

A generated sine wave with a frequency of 1  $kHz$  was transported through the entire cloud and is late by 16  $ms$  round trip time. At some point in time however, some buffer seems to underrun which could not be located yet. In this case, the latency of the return stream is stable at 316  $ms$  round trip time.

Both, the send and the return stream, show the expected IPG for the tested payload sizes. The send stream maintains an IPG of 2.666  $ms$  for 256 bytes (128mono samples/48  $kHz$ ) and the return stream maintains an IPG of 1.333  $ms$  for 256 bytes (64stereo samples/48  $kHz$ ), respectively.

An round trip time of 16  $ms$  (8  $ms$  end-to-end) for the Soundjack client streams, still violates the end-to-end delay limit of 2  $ms$  formulated in [9, p. 14]. Additional latencies that are introduced by the networks and not compensated yet are for example some minimal portion of undeterministic network behavior by the campus network, buffer synchronization latencies, packet de- and fragmentation inside the cloud. On the audio processing side we have to consider drift of the JACK audio server, because it is not phase locked to the incoming mediaclock stream. When the mediaclock is very early and the audio interface hardware interrupt is very late in relation to each other, the drift between those two interfaces can lead to a worst case latency of  $\Delta t_{max} = 1/48 kHz = 20.833 \mu s$ . This latency is much less significant than the latencies introduced by the signal processing algorithms that shall be implemented. A measurement of the latency introduced by the multicast mixing application has still to be done.

#### V. CONCLUSIONS

The Soundjack cloud prototype is not fully tested yet, but the evaluations in this paper show the proper operation of the presented concept. Analysis has shown that the AVB requirements could be mostly fulfilled. Further sources for the remaining latency, besides the actual algorithmical latency of the digital signal processing involved, have to be exposed.

#### VI. FUTURE WORK

The software has to be further optimized to reliably meet the AVB constrains.

Furthermore, the AVB server software behavior requires evaluation under heavy load with up to eight possible streams. In parallel to the creation of this paper further signal processing applications have been developed which will be integrated into the streaming process. The jitter and latency behavior under heavy load with signal processing applications in place will be evaluated in the future. Those evaluations will mainly focus on the task scheduling precision in terms of meeting the calculated task deadlines with the EBF-CBS scheduler.

Signal processing and machine learning application are under development and might be ready to be tested prior to a deployment in the public internet.

As soon as the proper networking operations of the cloud are verified, measurements will be performed in the public Internet. We will then deploy the Soundjack cloud in Frankfurt on the Main and test in a real world environment. A comparison between an IPv4 and an IPv6 deployment will be done as well.

#### VII. ACKNOWLEDGEMENTS

fast-music is part of the fast-project cluster (fast actuators sensors & transceivers), which is funded by the BMBF (Bundesministerium für Bildung und Forschung).

#### REFERENCES

- [1] (2018, Apr. 23) Soundjack - a realtime communication solution. [Online]. Available: <http://http://www.soundjack.eu>
- [2] A. Carôt, U. Krämer, and G. Schuller, "Network music performance (nmp) in narrow band networks," in *Proceedings of the 120th AES convention, Paris, France*. Audio Engineering Society, May 20–23, 2006.
- [3] A. Carôt, S. Ebeling, C. Hoene, P. Platz, and H. Loidan, "Glass panel displays with addressable leds," in *Mensch und Computer 2018 (MUC2018)*. Dresden, Germany: Gesellschaft für Informatik, Technische Universität Dresden, Sep. 2–5, 2018.
- [4] (2018, Apr. 23) Symonics gmbh. 72144 Dusslingen, Germany. [Online]. Available: <http://symonics.de>
- [5] H. Zimmermann, "Osi reference model -the iso model of architecture for open systems interconnection," in *IEEE Transactions on Communications, Vol. 28, No. 4*, Apr. 1980, pp. 425–432.
- [6] *Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE Std. 802.1AS, Mar. 2011.
- [7] *Virtual Bridged Local Area Networks - Amendment 14: Stream Reservation Protocol (SRP)*, IEEE Std. 802.1Qat-2010, Sep. 2010.
- [8] *Virtual Bridged Local Area Networks - Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, IEEE Std. 802.1Qav-2009, Jan. 2010.
- [9] *Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE Std. 1722, May 2011.
- [10] *Device Discovery, Connection Management, and Control Protocol for IEEE 1722 Based Devices*, IEEE Std. 1722.1, Aug. 2013.
- [11] *(Revision of IEEE Std 802.1Q-2011) - IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks*, IEEE Std. Std 802.1Q-2014, Dec. 2014.
- [12] (2018, Apr. 23) Xmos ltd. / attero tech inc. [Online]. Available: <http://www.atterodesign.com/cobranet-oem-products/xmos-avb-module/>
- [13] H. Weibel and S. Heinzmann, "Media clock synchronization based on ptp," in *Audio Engineering Society Conference: 44th International Conference: Audio Networking*, Nov 2011. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=16146>
- [14] J. Kacur, "Realtime kernel for audio and visual applications," in *Proceedings of the Linux Audio Conference 2010*. Wittenburg, DE: Red Hat, Apr. 2010.
- [15] C. Kuhr and A. Carôt, "Software architecture for a multiple avb listener and talker scenario," in *Proceedings of the Linux Audio Conference 2018*. Berlin, Germany: Linuxaudio.org, Jun. 7–10, 2018.
- [16] A. Carôt and G. Schuller, "Towards a telematic visual-conducting system," in *AES 44th International Conference, San Diego, USA*. Audio Engineering Society, Nov. 18–20, 2011.
- [17] (2018, Apr. 23) Jack audio connection kit. [Online]. Available: <https://jackaudio.org>
- [18] C. Kuhr, T. Hofmann, and A. Carôt, "Use case: Integration of a faust signal processing application in a livestream webservice," in *Proceedings of the 1st International Faust Conference 2018*. Mainz, Germany: Johannes Gutenberg-Universität Mainz, Jul. 17–18, 2018.