

CESNET Technical Report 11/2009

Audio Transport Implementation for UltraGrid Platform

MILOŠ LIŠKA, MARTIN BENEŠ, PETR HOLUB

Received 11.12.2009

Abstract

This document describes implementation of real-time transmissions of high quality audio for the UltraGrid platform. We have opted for standards compatible implementation of audio transmissions in accordance with RFC 3190. Also, our goal was to preserve the multi-platform character of UltraGrid and allow for future enhancements of the audio subsystem in UltraGrid. Therefore we have based the implementation on the Portaudio library.

Keywords: UltraGrid, audio, audio transmissions, PortAudio

1 Introduction

UltraGrid provides a system for low-latency uncompressed HDTV video transmissions. The original version of UltraGrid system was provided by Perkins and Gharai [3]. This version of UltraGrid has been however discontinued now. The CESNET team has modified the original UltraGrid version [1], added a number of new features and continues with its development, see the project web page¹. So far our implementation of UltraGrid lacked native audio transport implementation and users had to deal with audio transmissions separately. This uncomfortable state led us to implement support for audio transmissions into the UltraGrid system.

2 Protocol Specification

For media transport protocol UltraGrid sticks with popular Real-time Transport Protocol (RTP), which is well suited for for transmission of virtually any video or audio data. We have decided to implement the audio transmissions as out of band with respect to the video stream transmission. This approach has several advantages: (1) it is not necessary to deal with audio and video interleaving in the transmitted streams and thus the implementation is considerably simplified, (2) it is possible to send the audio stream to a completely different target machine to be processed separately which may be useful in a number of scenarios. On the other hand sending the audio stream out of band with respect to the video stream imposes increased requirements on the audio and video synchronization. This issue can be mitigated directly in UltraGrid via its capability to use time information from the RTP/RTCP packets or, e.g., synchronizing packet reflector.

¹ <https://www.sitola.cz/igrid/index.php/UltraGrid>

We have chosen to transmit 8 channels of audio encoded in 24 bit quality at 48000Hz sample rate. Such high quality might seem to be an overkill for videoconferencing scenarios. However, we aim at UltraGrid to be used also, e.g., in digital movie postprocessing industry where such a quality is a must. Moreover, bandwidth demands posed by such audio stream still represent much less than 1% of video data bandwidth. The bandwidth necessary to transmit audio in the above mentioned format is 1.2 Mbps per channel, thus 2.4 Mbps for standard stereo or up to 9.6 Mbps for full 8 channels. Compared to 1.2+ Gbps for transmission of uncompressed 1080i video and given high quality characteristics of UltraGrid it seems reasonable.

As we pursue UltraGrid to follow existing standards and recommendations, we have based processing and transmissions of the audio data on RFC 3190 [2], which describes RTP payload format for 12, 20 and 24 bit linear sampled audio. The packetization of the audio data is also based on the related RFC 1890 [4].

When processing the audio data, audio samples are represented as signed values in two's complement notation. Each audio sample is stored starting with the most significant bit. Samples are stored interleaved, starting with first sample of first channel, followed by first sample of second channel and so on. Samples for all channels belonging to a single sampling instant must be contained in the same packet. In this way 32 samples per channel (8×32 samples total) are stored in one RTP packet payload, resulting in 768 bytes of audio data per packet. The arrangement of the samples for all channels in the RTP packet payload is illustrated in Figure 1. As UltraGrid does not implement silence suppression, the marker bit in the RTP packet is always set to 0. We have set the payload type from the range of dynamically configurable payload types to be 97 (UltraGrid uses payload type 96 for RTP packets encapsulating the video).

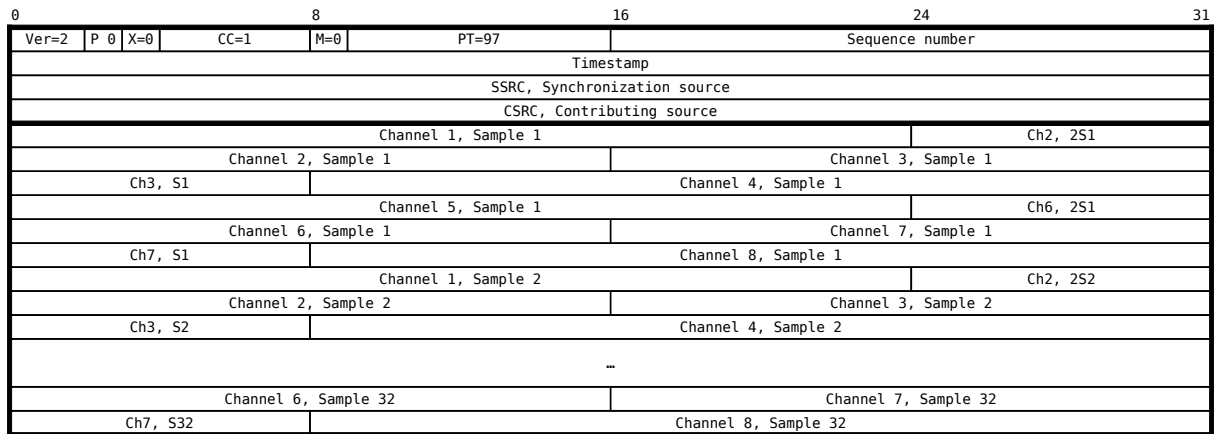


Figure 1. Illustration of audio data arrangement within a RTP packet. The fields in the RTP packet header conform to the definitions in RFC 1889 [5].

3 Portaudio Library

Because UltraGrid is a multiplatform software running on Linux and Mac OS X it was necessary to choose an audio hardware layer running on both platforms. Eventually, the portable cross-platform Audio API library Portaudio² was chosen.

Portaudio is an open source library distributed under the license compatible with MIT license, which allows anyone to use the code without restrictions. On Linux, Portaudio interfaces with OSS or ALSA³ (Advanced Linux Sound Architecture) drivers. When running on Mac OS X, Apple Core Audio⁴ architecture is used.

Portaudio library is currently distributed in two versions. The first version is marked as V18 and the second, newer version is marked as V19 with both versions being broadly used. The V19 version was chosen to be used with UltraGrid, because it provides some major improvements over V18, such as blocking API for writing to and reading off of audio device.

Portaudio library provides two basic ways to deal with audio streams. The non-blocking API and blocking API. Non-blocking API requires programmer to define call-back functions, which are later asynchronously called by Portaudio, whenever new data is required or provided.

On the other hand, blocking API defines blocking read/write functions, which means that function calls do not return until a sufficient amount of data is read or written. Using of the blocking API however requires native threads to be supported in order to interact with blocking efficiently.

Portaudio library offers a modular architecture and apart from multi-platform native audio system, also interfaces with generic audio servers, like JACK⁵ (Jack Audio Connection Kit – a system for handling real-time, low latency audio). JACK provides a promising low latency audio server which, apart from other features, allows for low-latency audio transmissions over network using JackTrip⁶ software. We also plan to eventually add support for DVS Centaurus and Centaurus II boards.

4 Audio Support Implementation in UltraGrid

The implementation is based on Portaudio library which provides access to local audio devices, and existing UltraGrid's Real-time Transport Protocol network stack, which takes care of network data transmissions. The original RTP stack had to be modified in order to provide a thread safe operations.

We have chosen the blocking Portaudio API for the UltraGrid audio support implementation. The blocking API allows for better control over threads spawned by the UltraGrid and simpler design of it's audio stack in comparison with the non-blocking API.

² <http://www.portaudio.com/>

³ http://www.alsa-project.org/main/index.php/Main_Page

⁴ <http://devworld.apple.com/audio/overview.html>

⁵ <http://jackaudio.org/>

⁶ <https://ccrma.stanford.edu/groups/soundwire/software/jacktrip/>

Audio processing runs in an extra thread, where data are being processed in a simple loop. On the sender side, once a chunk of audio data from audio interface is received, it is immediately sent to network stack, which packs the data into RTP protocol and sends it to the receiver. On the receiver side, similar simple loop running in its own thread takes care of sending audio data to the audio device, once it was received by the network stack.

5 Using UltraGrid for Audio Transmissions

Audio is enabled using command line switches `-r <device number>` and `-s <device number>`, where `-r` stands for receiver and `-s` stands for sender. Both switches take a numerical parameter, specifying which output/input audio device to use. Special value of `-1` can be used to specify a “default” device. The audio devices list is printed, when parameter `list` is specified instead of a device number. The default input device is marked using `(*i)` and the default output device using `(*o)`. Examples of available devices on Mac Pro and PC with Linux OS are depicted below. Using these parameters it is possible to run UltraGrid purely using command line, possibly running it remotely using `ssh`.

Examples of usage:

1. list all available audio outputs

```
uv -r list
```

2. list all available audio inputs

```
uv -s list
```

3. send audio data using input audio device number 2 to the IP address 10.0.0.1

```
uv -s 2 10.0.0.1
```

4. receive audio data from IP address 10.0.0.1 and play it using the default audio output device

```
uv -r -1 10.0.0.1
```

List of available devices on Mac Pro:

```
Available devices(5)
```

```
(*i) Device 0: Built-in Line Input (output channels: 0; input channels: 2)
```

```
Device 1: Built-in Digital Input (output channels: 0; input channels: 2)
```

```
Device 2: Built-in Output (output channels: 2; input channels: 0)
```

```
(*o) Device 3: Built-in Line Output (output channels: 2; input channels: 0)
```

```
Device 4: Built-in Digital Output (output channels: 2; input channels: 0)
```

List of available devices on PC with Linux OS and Alsa audio drivers:

```
Available devices(11)
Device 0: HDA NVidia: Analog (hw:0,0) (output channels: 2; input channels: 2)
Device 1: HDA NVidia: Digital (hw:0,1) (output channels: 2; input channels: 0)
Device 2: front (output channels: 2; input channels: 0)
Device 3: surround40 (output channels: 2; input channels: 0)
Device 4: surround51 (output channels: 2; input channels: 0)
Device 5: surround71 (output channels: 2; input channels: 0)
Device 6: iec958 (output channels: 2; input channels: 0)
Device 7: spdif (output channels: 2; input channels: 0)
(*i) (*o)Device 8: default (output channels: 128; input channels: 128)
Device 9: dmix (output channels: 2; input channels: 0)
Device 10: /dev/dsp (output channels: 16; input channels: 16)
```

6 Conclusions

In this report we have described the implementation of high-quality audio transmissions support in the UltraGrid system. The implementation is based on the multi-platform Portaudio library. Modular architecture of Portaudio allows for further enhancements and provides interface to number of other systems and tools to process the audio sent or received by UltraGrid. As for the future work there remains to implement disembedding of the audio captured as a part of the HD-SDI transmission using DVS Centaurus cards on Linux.

7 Acknowledgements

This work is supported by the research intent MSM6383917201.

References

- [1] HOLUB, P.; MATYSKA, L.; LIŠKA, M.; HEJTMÁNEK, L.; DENEMARK, J.; REBOK, T.; HUTANU, A.; PARUCHURI, R.; RADIL, J.; HLADKÁ, E. High-definition multimedia for multiparty low-latency interactive communication. *Future Generation Computer Systems*. 2006, vol. 22, no. 8, p. 856–861.
- [2] KOBAYASHI, K.; OGAWA, A.; CASNER, S.; BORMANN, C. *RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio*. RFC 3190⁷, IETF, January 2002.
- [3] PERKINS, C.; GHARAI, L.; LEHMAN, T.; MANKIN, A. Experiments with Delivery of HDTV over IP Networks. In *Proceedings of the 12th International Packet Video Workshop*, Pittsburgh, PA, USA, 2002. Available online⁸.
- [4] SCHULZRINNE H. *RTP Profile for Audio and Video Conferences with Minimal Control*. RFC 1890⁹, IETF, January 1996.

⁷ <http://tools.ietf.org/html/rfc3190>

⁸ <http://ultragrid.east.isi.edu/publications/pv2002.pdf>

⁹ <http://tools.ietf.org/html/rfc1890>

- [5] SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON, V.
RTP: A Transport Protocol for Real-Time Applications. RFC 1889¹⁰, IETF, January 1996.

¹⁰ <http://tools.ietf.org/html/rfc1889>