CESNET Technical Report 1/2012

4K Video and Audio Packet Format for UltraGrid 2nd Revision

PETR HOLUB, MILOŠ LIŠKA, MARTIN PULEC

Received 17. 2. 2012

Abstract

This report describes a packet format for low-latency transmissions of both uncompressed and compressed 4K video for UltraGrid platform. The goal of the format is to be generic enough to support also other types of video and audio streams, including high-definition video, 2K video and ultra-high-definition video. The packet format uses RTP headers to support legacy monitoring and analysis tools.

Keywords: packet format, multimedia, Real-time Transport Protocol, RFC 3550, RTP, audio, video, high bandwidth, extended numbering, multichannel audio, tiled video, sub-channels

Contents

1 Introduction

This memo defines a format for real-time transmission of high-definition (HD) and post-HD video streams, including both uncompressed and compressed data. The packet header structure is defined as simple as possible with fixed structure to simplify processing both in hardware and software. The format adopts RTP header structure [14] to facilitate interoperability with existing RTP monitoring and recording tools. Since RTP packet numbering is highly insufficient for high-bandwidth data, as witnessed also by other proposals such as RFC 4175 [4], we are introducing a concept of data buffer and data position within the buffer. This mechanism is also flexible enough to support both uncompressed and compressed transmissions including intra- and inter-frame compression. This is an alternative to very specific formats, such as uncompressed HD video defined in RFC 4175 [4].

2 Use Scenarios

The proposed packet structure is suitable for generic high-bandwidth real-time multimedia streaming. The reason behind our proposal is that while several payload formats have been proposed for RTP in last five years for high-bandwidth data (as further discussed in Section 7), they are mostly targeting uncompressed video and are not flexible enough to support another media types or even low-latency media with low compression ratios and therefore high bandwidth (such as DXT compression). Another important practical aspect is that high-bandwidth data is often sent in multiple channels, be it four 2K tiles for 4K video or multichannel uncompressed audio. Relative position of these channels needs to be preserved during the processing in RTP mixers. Thus the standard RTP mechanisms such as the CSRC field can not be reused for this purpose.

The scenarios targeted by this proposal include:

- 4K and Super-HD video (up to 4096×3072 or 3840×2160 respectively)
  - uncompressed 4 tiles in a single multiplexed stream
  - uncompressed 1 tile in a single stream
  - compressed 1 tile in a single stream
- 2K and HD video (up to 2048×1536 or 1920×1080 respectively)
  - uncompressed in a single stream
  - compressed in a single stream
- post-4K video formats
  - uncompressed multiple tiles in a single multiplexed stream
  - uncompressed 1 tile in a single stream
  - compressed 1 tile in a single stream
- audio (44–192 kHz sampling frequency, 16–32 b per sample) with a large number of channels (up to 1024 with the sub-stream ID defined here)
  - uncompressed in a single multiplexed stream
  - compressed in a single multiplexed stream

– multiplexed in a single stream into the video stream (e.g., HD-SDI format sent "as is")

3 Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL " in this document are to be interpreted as described in BCP 14, RFC 2119 [2] and indicate requirement levels for compliant RTP implementations.

4 Packet Header

The proposed packet header is composed of a standard RTP header and the proposed payload header, as further discussed in this section.

4.1 RTP Header

This proposal adopts an RTP packet header (RFC 3550 [14], shown also in Figure 1) for backward compatibility with existing RTP tools. Based on the RFC 3550 recommendation, the packet header is not extended and an additional payload header is defined.

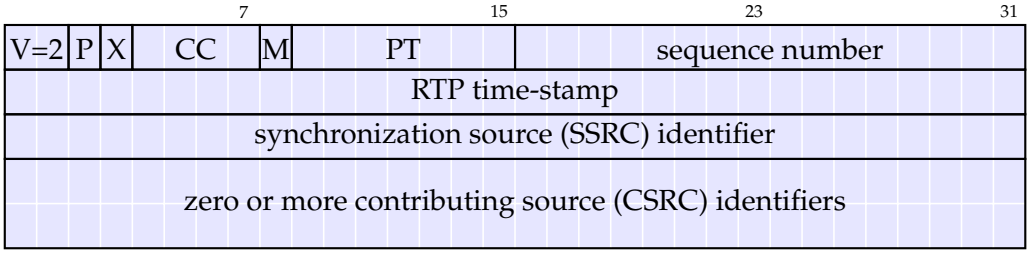

Figure 1: RTP packet header format [14]. V . . . version. P . . . padding. X . . . extension. CC . . . CSRC count. M . . . marker. PT . . . payload type.

Because of limited assignments of payload types in RFC 3551 [13], the payload type MUST be set to 20 for video and 21 for audio. Intentionally, we use these two values which are unassigned by the RFC 3551 (it is RECOMMENDED to use unassigned values even in case of overflow of the number of dynamic payload types). The actual type of data contained within the payload is specified in the payload header defined below. The RTP header MUST NOT use an extension header and the X field MUST NOT be set. All other fields MUST be set according to the RTP specification in RFC 3550. The RTP stream SHOULD be accompanied by an RTCP stream.

4.2 Video Payload Header

The proposed header (shown in Figure 2) occupies fixed size of 24 bytes/octets to facilitate its efficient processing in hardware. For high-bandwidth data, it is as-

sumed to use at least maximum standard Ethernet frame size (1500 B), or more preferably, Jumbo or Super-Jumbo frame size (up to 9000 B or even more); thus we consider packetization overhead marginal. The packet header doesn't contain any extending mechanism: if applications need it, it SHOULD be defined as a part of the payload again; this principle reflects what has been successfully used in RTP.

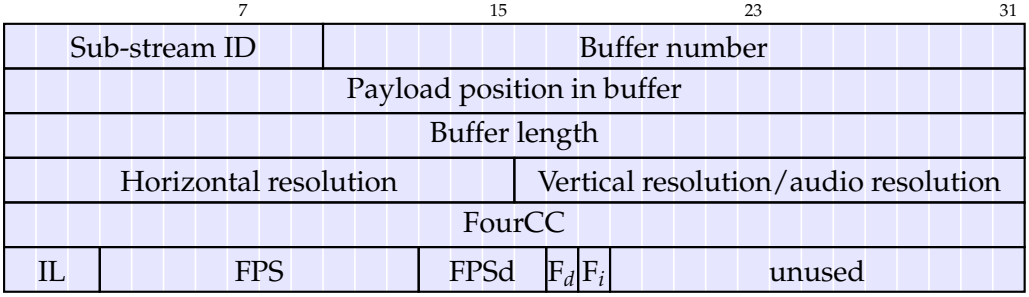

Figure 2: Video payload header.

Sub-stream ID: 10 bits

identifies substreams within a stream of related data. Such streams with multiple substreams typically represent tile positions for 4K tiled video (left-top, left-bottom, right-top, right-bottom). This ID MUST NOT be rewritten by network stream mixers (in RTP sense), unless the data is processed accordingly. Semantics of the substreams (e.g., mapping of tiles in tiled 4K video to their (x,y) position on the screen) needs to be provided by the application or using out of band signaling, for example SDP. The reason for this is that the semantics of the substreams can be quite complex and thus beyond what should be specified in a terse header of each packet.

Buffer number: 22 bits

specifies the number of the buffer to which the data belongs. Together with the position in the buffer it provides full packet sequence information and avoids problems with short RTP sequence numbers. At 120 video frames per second and one buffer per frame, it provides for 2.4 hours of continuous streaming before the numbering turns over.

Payload position in buffer: 32 bits

defines the position of the packet payload within the application buffer. It is expressed in bytes/octets.

Buffer length: 32 bits

expresses buffer length in bytes/octets. 2^{32} B should be sufficient to fit mid-term future video formats, up to 4×4 matrix of 4K video (e.g., the buffer MAY contain up to 3 uncompressed frames 16,384 × 12,288 with 3 color components and 16 b per color component, which totals to 1152 GiB per one frame).

Horizontal resolution: 16 bits

describes horizontal resolution of the video.

Vertical resolution/audio resolution: 16 bits

describes vertical resolution of the video.

FourCC: 32 bits

describes pixel format, codec and compression format using common FourCC database[1]. For video streams with embedded audio, new FourCC may need to be defined.

IL: 3 bits

specifies interlacing format of the video:

0 – progressive video,
1 – interlaced with upper field first,
2 – interlaced with lower field first,
3 – interlaced video merged into a single frame,
4 – progressive scan field.

FPS: 10 bits

specifies frame rate (frames per second) as integer 1–1024. Fractional frame rates are specified by setting FPSd, F_d, and F_i packet fields.

FPSd: 4 bits

specifies the frame rate in FPS is divided by an integer 1–16.

F_d: 1 bits

specifies, whether the frame rate is divided by 1.001 (set to 1), as common for NTSC-based frame rates, or not (set to 0).

F_i: 1 bits

specifies inversion of resulting packet rate (after computing the frame rate using FPS, FPSd, and F_d values). This is designed for special applications, that play less than one frame per second (typically at very high resolution).

The approach chosen for frame rate specification differs from both MPEG and AVI approaches. MPEG uses 4 b index table specifying one of 23.976, 24, 25, 29.97, 30, 50, 59.94, 60 frames per second rates. AVI on the other hand uses 32 b delay between consecutive frames in milliseconds. Our proposal achieves higher precision for common fractional rates and allows also specification of non-standard rates, while occupying less space compared to the AVI header approach.

4.3 Audio Payload Header

The proposed header for separate audio streams (shown in Figure 3) occupies fixed size of 20 bytes/octets to facilitate its efficient processing in hardware. Only the fields different from video payload header are described here.

[1] `http://www.fourcc.org/fourcc.php`

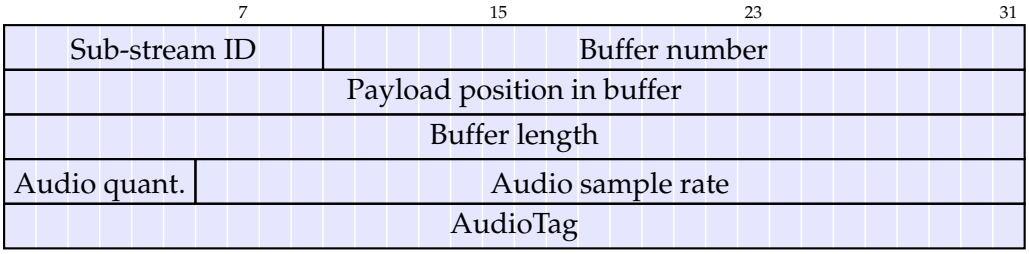

Figure 3: Audio payload header.

Sub-stream ID: 10 bits

identifies substreams within a stream of related data, typically audio channels (e.g., stereo, 5.1, 7.1). This ID MUST NOT be rewritten by network stream mixers (in RTP sense), unless the data is processed accordingly.

Audio quant.: 6 bits

describes size of audio samples. Typical values include 16/24/32, but for DCD it MAY also be 1 (i.e., delta-sigma conversion).

Audio sample rate: 26 bits

describes sampling frequency for audio signal. Frequencies up to 16,777,216 Hz are supported, which should be sufficient even for DCD conversion.

AudioTag: 32 bits

describes the audio format using Audiotag[2].

Note on PCM endianity: Audiotag does not distinguish endianity of the PCM streams. Based on common practice (as witnessed, e.g., by MPlayer), 1 SHOULD be used for little endian PCM and 9999 SHOULD be used for big endian PCM.

5 Packet Payload

The packet payload MUST be sent unmodified in the payload from buffer to buffer, complying with FourCC specification for each buffer. For simplicity and efficiency reasons (this specification is intended for high-bandwidth applications with possibly very high packet rates), payload headers and other extensions SHALL NOT be included.

6 Other Recommendations

6.1 Packet Loss Detection

Packet loss detection can be implemented based on RTP packet numbering. The intention behind using RTP packet header is to provide backward compatibility with existing RTP monitoring (namely packet loss and jitter) and QoS applications.

[2] `http://www.audiotag.org/tags.php`

For maintenance of sliding window (buffer) for reception of packets, it is possible to store pointer to the last buffer position received as a continuous block (i.e., without holes caused by packet loss or reordering). This approach is equivalent to the TCP, which specifies the last continuous byte received plus one (i.e., next byte expected).

6.2 Packet Reordering

The buffer number and the position within the buffer can be used to insert reordered packets directly onto their position in the receiving buffer.

6.3 Packet Size Recommendation

In order to minimize the number of packets transmitted per second, which is important especially for a software-based implementation, a maximum available frame size SHOULD be used.

6.4 Forward Error Correction

Because the primary focus of the proposed specification is low-latency media transmission, forward error correction (FEC) provides a useful means to minimize the effects of packet loss on the receiver without the need for data retransmission (which is not an option especially on long distance network links). FEC packet format is specified in RFC 5109 [9] primarily using XOR-based parity and can be used with the packet format proposed in this memo. However, as the data is not restructured for the network transmission, only a single level (level 0) of protection is supported. Also, high packet rate for high bit rate streams with smaller packet size may result in larger reordering where RTP numbering range is not sufficient to identify matching packets. If this becomes the case in practice, the FEC structure will need to be extended. There is also an extension specific for Reed-Solomon codes [3].

7 Related Work

7.1 Related Standards

RFC 3550 – *RTP: A Transport Protocol for Real-Time Applications* [14].

RFC 3550 provides description of the Real-time Transport Protocol and its usage for real-time multimedia streams transmissions. We opt for adopting of the RTP protocol especially because of its data transport being augmented by a control protocol (RTCP) to allow monitoring of the data delivery, and to provide minimal control and identification functionality.

On the other hand the RTP protocol as it was accepted by The Internet Society in 2003 is deficient for our purposes at least in two aspects. The RTP sequence numbering is highly insufficient for high-bandwidth uncompressed video transmissions. The default 16 b sequence number overruns in 1.2 s with

9000 B large payload of 3840×2160, 8 b 4:2:2 YUV video at 30 fps. Also the description of transmitted media provided by the payload type in RTP is insufficient. The RTP payload types assignment have already been closed by IANA (see [13], Section 3), although it barely describes especially uncompressed and compressed video payloads used in contemporary high-end video transmission applications (see Section 7.2).

RFC 3551 – *RTP Profile for Audio and Video Conferences with Minimal Control* [13].

RFC 3551 defines content-specific format descriptions and encoding rules for audio formats (DVI4, G722, G723, G726-40, G726-32, G726-24, G726-16, G728, G729, G729D and G729E, GSM, GSM-EFR, L8, L16, LPC, MPA, PCMA, PCMU, QCELP, RED, VDVI) and video formats (CelB, JPEG, H261, H263, H263-1998, MPV, MP2T, nv) with minimal control, i.e. no negotiation of transfer parameters. However, while minimum control paradigm applies also for UltraGrid (and also iHDTV, MVTP-4K and iVisto), the limited space of payload types and closed registration by IANA does not allow us to use static payload types effectively. Because static payload types are desirable for our application, we use a unassigned values of 20 and 21 and further use FourCC in the payload header defined in our specification. Use of unassigned values is RECOMMENDED by the RFC 3551 also for dynamic types if the dynamic range 96–127 is not sufficient.

RFC 3190 – *RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio* [7]

RFC 3190 extends the definition of L16 audio format and encoding rules provided in RFC 3551 for high-quality audio. Although UltraGrid adhered to the recommendations provided in this RFC for high-quality audio transmissions over RTP (see [10] for details), the RFC does not facilitate transmissions of compressed audio content at all.

RFC 3497 – *RTP Payload Format for Society of Motion Picture and Television Engineers (SMPTE) 292M Video* [5].

RFC 3497 solves the insufficient RTP sequence numbering by adding additional 16 bits for the sequence number to the payload header. Hence, 32 b is available for the sequence numbering similarly to our scheme. The packetization of the video data into the RTP payload is performed over individual scan lines of the video. The design of the payload format in RFC 3497 is however tailored specifically for SMPTE 292 video and is limited by the 12 b line number header field to represent only up to 4094 scan lines of video which in case of SMPTE 292 inherently includes line blanking and ancillary data.

RFC 4175 – *RTP Payload Format for Uncompressed Video* [4].

Payload format for a range of high-definition video formats such as SMPTE 274M or SMPTE 296M is described in RFC 4175. RFC 4175 solves the 16 b sequence number deficiency in the same way as RFC 3497. The packetization of the video data into the RTP payload is again performed over individual scan lines of the video. Data belonging to a particular scan line within a frame of a video is described by 15 b line number and 16 b offset within the line

in question. Hence, the payload format is sufficient to support even future video formats up to 4×4 matrix of 4K video with 3 color components and 10 b per color component. However, the scan line based packetization of the video data renders the packet format unsuitable for compressed video transmissions.

RFC 4421 – *RTP Payload Format for Uncompressed Video: Additional Colour Sampling Modes* [12].

RFC 4421 is not directly related to our needs and to RTP payload format for video transmissions in general as it only adds RGB color sampling modes to the video transmissions signalling defined in [4].

RFC 5109 – *RTP Payload Format for Generic Forward Error Correction* [9].

RFC 5109 [9] specifies concept of FEC packets for RTP, primarily using using XOR parity. Data is split into media packets and FEC packets. For media data structured into levels according to their importance, uneven level of protection (ULP) is available.

An extension specific to Reed-Solomon codes is available as a *RTP Payload Format for Reed Solomon FEC* draft [3].

Specific packet formats for compressed media There is a number of RFC-based standards for compressed media beyond RFC 3551. The purpose of the format-specific packetization is to allow for specific properties, such as increased resilience and improved reconstruction of the data in case of packet loss/reordering. This is however achieved at the loss of generality of the format. The following list is not complete and is intended only to give pointers to further reading:

- *RFC 2038 – RTP Payload Format for MPEG1/MPEG2 Video* [6].
- *RFC 2435 – RTP Payload Format for JPEG-compressed Video* [1].
- *RFC 3189 – RTP Payload Format for DV (IEC 61834) Video* [8].
- *RFC 3984 – RTP Payload Format for H.264 Video* [15].

7.2 Applications

While the primary application the packet format has been designed for is UltraGrid, there are other applications and hardware device with similar functionality that could benefit from it as well.

UltraGrid is a software application designed for low-latency HD video streaming over IP networks. Started by Gharai and Perkins in 2002[3], there has been multiple clones developed by CESNET[4], KISTI[5], and i2CAT[6]. Recent releases

[3] `http://csperkins.org/research/ultragrid/`

[4] `http://ultragrid.sitola.cz/`

[5] `www.gloriad-kr.org/hdtv/`

[6] `http://wiki.i2cat.net/doku.php/i2cat:public:clusters:audiovisual:uhdgroup:ultragrid`

feature also support for 2K and 4K video and compression, while still focusing on low-latency high-bandwidth media transmission. UltraGrid also implements data compatibility mode for iHDTV transport.

MVTP-4K is an HD-SDI over IP hardware converter based on FPGA, aimed at very low-latency 4K video transmissions developed by CESNET[7].

iHDTV is a software implementation of HD video over IP networks created by ResearchChannel and University of Washington consortium[8] and later made open-source[9]. It features a custom data transmission format based on UDP streaming and a very lightweight header structure [11].

NTT i-Visto is a commercial hardware implementation of HD-SDI over IP by NTT[10]. It uses proprietary UDP-based packetization and was promised to deliver iHDTV compatibility.

NTT JPEG2000 for 4K video is commercial hardware implementation of JPEG2000 compressed 4K video over IP for both real-time transmission and recording/playback applications by NTT[11]. Uses packet format specified as a part of the product documentation.

Acknowledgment

This project has been supported by a research intent MŠM 6383917201 and infrastructure grant LM2010005.

References

[1] Berc, L. M.; Fenner, W. C.; Frederick, R.; aj.: RTP Payload Format for JPEG-compressed Video. RFC 2435, IETF, Říjen 1998.
URL http://tools.ietf.org/search/rfc2435

[2] Bradner, S.: Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, IETF, Březen 1997.
URL http://tools.ietf.org/search/bcp14

[3] Galanos, S.; Peck, O.; Roca, V.: RTP Payload Format for Reed Solomon FEC. IETF draft draft-galanos-fecframe-rtp-reedsolomon-02, IETF, Duben 2010.
URL http://zinfandel.levkowetz.com/html/draft-galanos-fecframe-rtp-reedsolomon-02

[4] Gharai, L.; Perkins, C.: RTP Payload Format for Uncompressed Video. RFC 4175, IETF, Září 2005.
URL http://tools.ietf.org/search/rfc4175

[7] http://www.ces.net/project/qosip/hw/mvtp-4k.pdf
[8] http://www.washington.edu/ihdtv/
[9] http://ihdtv.sourceforge.net/
[10] http://www.i-visto.com/
[11] http://www.ntt-at.com/products_e/jpeg2000/

[5] Gharai, L.; Perkins, C.; Goncher, G.; aj.: RTP Payload Format for Society of Motion Picture and Television Engineers (SMPTE) 292M Video. RFC 3497, IETF, Březen 2003.
URL http://tools.ietf.org/search/rfc3497

[6] Hoffman, D.; Goyal, V.; Fernando, G.: RTP Payload Format for MPEG1/MPEG2 Video. RFC 2038, IETF, Říjen 1996.
URL http://tools.ietf.org/search/rfc2038

[7] Kobayashi, K.; Ogawa, A.; Casner, S. L.; aj.: RTP Payload Format for 12-bit DAT Audio and 20- and 24-bit Linear Sampled Audio. RFC 3190, IETF, Leden 2002.
URL http://tools.ietf.org/search/rfc3190

[8] Kobayashi, K.; Ogawa, A.; Casner, S. L.; aj.: RTP Payload Format for DV (IEC 61834) Video. RFC 3189, IETF, Leden 2002.
URL http://tools.ietf.org/search/rfc3189

[9] Li, A. H.: RTP Payload Format for Generic Forward Error Correction. RFC 5109, IETF, Prosinec 2007.
URL http://tools.ietf.org/search/rfc5109

[10] Liška, M.; Beneš, M.; Holub, P.: Audio Transport Implementation for UltraGrid Platform. Technická Zpráva 11, CESNET z.s.p.o., Prosinec 2009.
URL http://www.cesnet.cz/doc/techzpravy/2009/audio-transport-ultragrid/

[11] Liška, M.; Beneš, M.; Holub, P.: iHDTV Protocol Implementation for UltraGrid. Technická Zpráva 12, CESNET z.s.p.o., Prosinec 2009.
URL http://www.cesnet.cz/doc/techzpravy/2009/ihdtv-implementation-ultragrid/

[12] Perkins, C.: RTP Payload Format for Uncompressed Video: Additional Colour Sampling Modes. RFC 4421, IETF, Únor 2006.
URL http://tools.ietf.org/search/rfc4421

[13] Schulzrinne, H.; Casner, S. L.: RTP: Profile for Audio and Video Conferences with Minimal Control. RFC 3551, IETF, Červenec 2003.
URL http://tools.ietf.org/search/rfc3551

[14] Schulzrinne, H.; Casner, S. L.; Frederick, R.; aj.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF, Červenec 2003.
URL http://tools.ietf.org/search/rfc3550

[15] Wenger, S.; Hannuksela, M. M.; Stockhammer, T.; aj.: RTP Payload Format for H.264 Video. RFC 3984, IETF, Únor 2005.
URL http://tools.ietf.org/search/rfc3984