UVGRTP 2.0: OPEN-SOURCE RTP LIBRARY FOR REAL-TIME VVC/HEVC STREAMING

Aaro Altonen, Joni Räsänen, Alexandre Mercat, and Jarno Vanne

Ultra Video Group, Tampere University, Finland
{aaro.altonen, joni.rasanen, alexandre.mercat, jarno.vanne}@tuni.fi

ABSTRACT

Real-time video transport plays a central role in various interactive and streaming media applications. This paper presents a new release of our open-source *Real-time Transport Protocol* (*RTP*) library called uvgRTP (github.com/ultravideo/uvgRTP) that is designed for economic video and audio transmission in real time. It is the first public library that comes with built-in support for modern VVC, HEVC, and AVC video codecs and Opus audio codec. It can also be tailored to diversified media formats with an easy-to-use generic API. According to our experiments, uvgRTP can stream 8K VVC video at 300 fps with an average round-trip latency of 4.9 ms over a 10 Gbit link. This cross-platform library can be run on Windows and Linux operating systems and the permissive BSD 2-Clause license makes it accessible to a broad range of commercial and academic streaming media applications.

Index Terms— Open source, Video streaming, Real-time Transport Protocol (RTP), Versatile Video Coding (VVC), High Efficiency Video Coding (HEVC), Advanced Video Coding (AVC)

1. INTRODUCTION

Our society is surrounded by a myriad of real-time media streaming applications such as video communication and content delivery, and their importance [1] is even further amplified amid the COVID-19 crisis. The popularity of these applications is based on the latest video coding and transport technologies that together can enable efficient video transfer with available network bandwidths.

The landscape of international video coding standards is dominated by the universal *Advanced Video Coding* (*AVC/H.264*) [2], well-established *High Efficiency Video Coding* (*HEVC/H.265*) [3], and emerging *Versatile Video Coding* (*VVC/H.266*) [4]. Real-time AVC/HEVC/VVC streaming can be implemented with the widespread *Real-time Transport Protocol* (*RTP*), specified in RFC 3550 [5]. RTP is built on top of *User Datagram Protocol* (*UDP*) making it ideal for real-time streaming with low latency. In addition, RFC 3550 defines the *RTP Control Protocol* (*RTCP*) for conveying information about the participants and *Quality of Service* (*QoS*) monitoring.

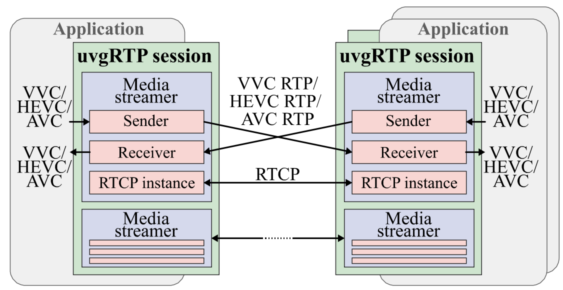

Fig. 1. Usage scenario of the proposed uvgRTP library.

Existing open-source RTP libraries are characterized in Table 1. None of the prior solutions [6]–[13] support VVC and most of them come without native support for HEVC [6]–[10]. Furthermore, LIVE555 does not conform to RFC 3550 [11], whereas GStreamer [12] and FFmpeg [13] are complete multimedia frameworks that are not optimal for high-performance applications.

This paper gives an overview of our RTP library called uvgRTP that is the first open-source library designed for real-time VVC streaming. Our recent work presented the first version of the library for HEVC [14] and this paper introduces the second version, a.k.a. uvgRTP 2.0 with built-in support for VVC, HEVC, and AVC.

2. LIBRARY ARCHITECTURE

Fig. 1 illustrates the basic workflow of uvgRTP when applied in two-way point-to-point communication between different media applications. uvgRTP has one session for each peer it exchanges media with, and each session can have multiple *media streamers* for different media such as video and audio. A sending application uses the uvgRTP sender to transfer data to the corresponding uvgRTP receiver that passes it on to a receiving application.

Fig. 2 describes the high-level architecture of our uvgRTP library with dependency relations between its components. The application interacts with the uvgRTP instance through the

Table 1. The main characteristics of the proposed uvgRTP library and other existing open-source RTP streaming libraries.

Ref.	Library	VVC [15]	HEVC [16]	AVC [17]	RTPv2	Opus [18]	Language	LoC	License	First Activity	Last Activity
[6]	libre	No	No	No	Yes	Yes	C	58k	BSD	2010	Active
[7]	PJSIP	No	No	Yes	Yes	Yes	C	360k	GPL-2.0	2003	Active
[8]	libsrtp	No	No	No	Yes	Yes	C	23k	BSD	2001	Active
[9]	JRTPLIB	No	No	No	Yes	Yes	C++	28k	MIT	1999	2020
[10]	ccRTP	No	No	No	Yes	Yes	C++	14k	GPLv2	2001	2015
[11]	LIVE555	No	Partially	Partially	No	Yes	C++	72k	LGPLv3	1996	Active
[12]	GStreamer	No	Yes	Yes	Yes	Yes	C	3062k	LGPLv2.1	2001	Active
[13]	FFmpeg	No	Yes	Yes	Yes	Yes	C	1250k	LGPLv2.1	2000	Active
	uvgRTP 2.0	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**C++**	**13k**	**BSD-2**	**2019**	**Active**

978-1-6654-3864-3/21/$31.00

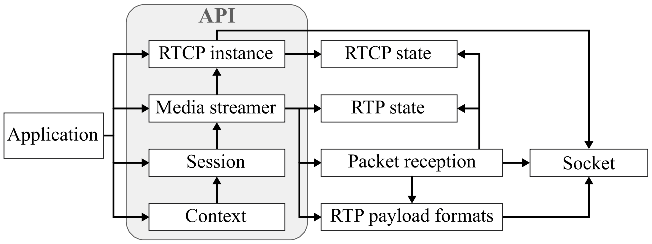

Fig. 2. Software architecture of uvgRTP.

Application Programming Interface (*API*), which consists of *context*, *session*, *media streamer*, and *RTCP instance* modules. The *context* is the top-level module, and it is used to allocate separate *sessions* for each IP address. In turn, the *session* creates the *media streamer*, which takes care of sending and/or receiving of a single RTP stream. The *RTCP instance* manages the RTCP of the corresponding RTP stream.

The *RTP state* and *RTCP state* modules uphold the current state of respective streams. The *packet reception* module receives the raw datagrams, constructs them into RTP frames, and dispatches them to various handlers. The *RTP payload format*s module creates and parses VVC [15], HEVC [16], AVC [17], and Opus [18] payload formats. It also implements a generic, easy-to-use API for handling any other RTP payload format. Finally, the *socket* module is responsible for sending and receiving RTP packets.

3. PERFORMANCE

In our experiments, uvgRTP 2.0 was benchmarked in a 10 Gbit local network between two desktop computers that were equipped with Core i7-4770 and AMD Threadripper 2990WX processors. All tests were carried out with a single 4K120p (3840×2160 pixels) raw test video that was encoded to a bit rate of 441.6 Mbit/s, 660.8 Mbit/s, and 1483.6 Mbit/s in VVC, HEVC, and AVC formats, respectively. The tests were run 100 times and the obtained results were averaged.

According to our results, uvgRTP attained a goodput of 4.6 Gbit/s over a 10 Gbit link with an average frame loss of 0.06%. This corresponds to streaming 8K VVC, HEVC, and AVC video at around 300 fps, 210 fps, and 90 fps, respectively. The average round-trip latency for VVC was 4.9 ms (14.5 ms for intra and 3.7 ms for inter frames).

4. COMPILATION AND USAGE

The source code of uvgRTP is available on GitHub at:

https://github.com/ultravideo/uvgRTP

under the BSD 2-Clause license. The repository contains a list of commented use case scenarios and build instructions for *CMake*, which can be used to generate build files, e.g., for *GNU Make* on Linux and *MinGW* or *Visual* on Windows. The compilation produces a library, which can be linked to a media application.

Using the library requires that a context is created. An application utilizes the context to allocate a session, which is used to allocate one or more *media streamers*. The *media streamer* sends media with the *push_frame* function and returns received media with the *pull_frame* function. Alternatively, media can be returned by installing a receive hook with the *install_receive_hook* function. A media codec is specified when creating the *media streamer*. The RTCP packets can be obtained from the *RTCP instance*.

5. APPLICATIONS AND INTENDED AUDIENCE

uvgRTP is designed for practically any kind of streaming media, including multifaceted audiovisual data from traditional 2D to immersive 3D spherical content. The field of applications can involve interactive video communication platforms, live streaming services, and multimedia-oriented *Internet of Things* (*IoT*). The permissive BSD license together with outstanding streaming performance makes uvgRTP ideal for commercial and exploratory research applications dealing with standardized or experimental media formats.

6. ACKNOWLEDGMENT

This paper is part of the PRYSTINE project that has received funding within the ECSEL JU in collaboration with the European Union's H2020 Framework Programme (H2020/2014-2020) and National Authorities, under grant agreement 783190.

7. REFERENCES

[1] Cisco Systems, Inc., “Cisco visual networking index: forecast and trends 2017–2022,” Dec. 2018.

[2] *Advanced Video Coding* for *Generic Audiovisual Services*, document ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC, Mar. 2009.

[3] *High Efficiency Video Coding*, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T Rec. and ISO/IEC, Apr. 2013.

[4] *Versatile Video Coding*, document ITU-T Rec. H.266 and ISO/IEC 23090-3, ITU-T and ISO/IEC, 2020.

[5] IETF RFC 3550 RTP: A Transport Protocol for Real-Time Applications, IETF, July 2003. [Online]. Available: https://tools.ietf.org/html/rfc3550

[6] libre. [Online]. Available : https://github.com/creytiv/re

[7] PJSIP. [Online]. Available: https://www.pjsip.org/

[8] libsrtp. [Online]. Available: https://github.com/cisco/libsrtp

[9] JRTPLIB. [Online]. Available: https://github.com/j0r1/JRTPLIB

[10] ccRTP. [Online]. Available: https://www.gnu.org/software/ccrtp/

[11] LIVE555. [Online]. Available: http://www.live555.com/

[12] GStreamer [Online]. Available: https://gstreamer.freedesktop.org

[13] FFmpeg. [Online]. Available: https://www.ffmpeg.org/

[14] A. Altonen, J. Räsänen, J. Laitinen, M. Viitanen, and J. Vanne, “Open-source RTP library for high-speed 4K HEVC video streaming,” *in Proc. IEEE Int. Workshop Multimedia Signal Process.*, Tampere, Finland, Sept. 2020.

[15] IETF, RTP Payload Format for Versatile Video Coding (VVC). [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-avtcore-rtp-vvc/

[16] IETF RFC 7798 RTP Payload Format for High Efficiency Video Coding (HEVC), IETF, Mar. 2016. [Online]. Available: https://tools.ietf.org/html/rfc7798

[17] IETF RFC 6184 RTP Payload Format for H.264 Video IETF, Mar. 2011. [Online]. Available: https://tools.ietf.org/html/rfc6184

[18] RTP Payload Format for the Opus Speech and Audio Codec [Online]. Available: https://tools.ietf.org/html/rfc7587