International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 07, July-2022

Implementation of RFC 7798 and Ffmpeg for Encoding and Decoding of Media Files in SoftDSP

K. Suharika
Department of ECE
RV College of Engineering Bangalore, India

Rohini S Hallikar
Department of ECE
RV College of Engineering Bangalore, India

***Abstract*—Over the last two decades, video compression has experienced several modifications. However, it appears that each new standard promises the same thing: the same visual quality at half the bitrate of the preceding one. Indeed, a 5Mbit/s HD H.265 video will have almost comparable quality as a 10Mbit/s H.264 video. It is accomplished through advancementsin both interframe and intraframe compression. The purpose of this paper is to provide an overview of high-efficiency videocodec, the new video compression standard, and explains how to packetize them in line with RFC 7798, which specifies the RTP payload format for H.265.It discusses the conversion of any videocodec to HEVC/H.265 (High-Efficiency Video Coding), a newvideo compression standard with the potential to surpass earlier standards such as H.264/Advanced Video Coding (AVC). FFmpegTool is used for the transcoding process. Following the acquisition of the HEVC video, a comprehensive technique is constructedin CPP since it takes the fewest CPU cycles when comparedto any other language. This method accepts any HEVC file as input and provides information about each header as well asthe slice content. The slices are packetized using the Real-Time Transport Protocol (RTP) Protocol in accordance with Remote Function Call (RFC) 7798, which is the RTP payload format for HEVC.**

Index Terms*—*High-Efficiency Video Codec, compression, Quality.

I. INTRODUCTION

FFmpeg is known as the Swiss Army knife of video transcoding/streaming. It is a free, open-source, and cross-platform multimedia framework that is extensively used. Ffmpeg is used by many popular and major apps or services,including YouTube, iTunes, and VLC. Since it supports a widenumber of codecs and containers, it is the most used tool for transcoding or converting audio/video from one format to another. It has a plethora of filters that may be used to edit and transform content in a variety of ways.

The video compression capability of FFmpeg is remarkable, and most streaming firms use or have used it for their produc- tion systems. JPEG, MPEG-1/2/4, H263+AAC (MPEG), The- ora (Ogg Vorbis), AVS+, VP8 (WebM), H.264/AVC, HEVC,

AV1, and other codecs are supported by FFmpeg libraries andmay be used to compress, encode, or decode movies as needed.As a result, various media files are transcoded to the appropriate format, using FFmpeg and the transcoded video isfed to the algorithm, which analyses and parses the headers.In this project any sample video is transcoded to HEVC video.

The video is first transcoded to the Hevc codec using a library named libx265 that is supported by FFmpeg. The parsing process and specifications are defined further in the methodology section, and the details of the code has been explained in the Algorithm section.

In [2] and [3], GStreamer and FFmpeg are the most popular and actively developing multimedia technologies used, which provide media streaming. Furthermore, their large range of application scenarios makes them less suited for small projects and applications that do not require considerable media pro- cessing skills.

The Design and Implementation of a Real-Time Video Stream Analysis System Based on FFmpeg was described in the paper [4], where the approach overcomes the limitations of existing stream analysis systems that only support a single data format and is more appropriate for Internet media stream analysis

The expanding volume of video traffic in telecommunications networks is discussed in this study [5], and the significance of efficient video compression methods is emphasized. A new video coding standard called High-Efficiency Video Coding (HEVC) will result in considerable bit rate reductions over its forerunners. In the HEVC standardization process, technologies including image partitioning, reference picture management, and parameter sets are categorized as “high-levelsyntax.” The interface to systems, error resilience and additional functionality are all impacted by the high-level syntax design. This article’s goal is to give a general understanding ofthe HEVC high-level syntax, which contains message headers for further enhancement information, parameter settings, im- age segmentation methods, and network abstraction layer unit headers.

In [6], the latest video coding standard produced bythe ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group which is High Efficiency VideoCoding (HEVC) has been described. The major objective is to achieve considerable improvements in compression and performance over existing standards in the region of 50 percent and bit-rate reduction for equal perceptual video quality. This article gives an overview of the HEVC standard’stechnical features and characteristics.

Following a thorough review of several articles, it was determined to move forward with this project for a numberof reasons, one of which is stated below.

Video and audio manipulation is described as a new type of media manipulation that targets digital video by combining classic video processing and video editing techniques with auxiliary artificial intelligence tools such as facial recogni- tion.It is frequently a time- consuming and error-prone opera- tion. Exploring alternative approaches in FFmpeg to solve thisproblem is the first step.

The objectives this paper addresses are mentioned below

- Initialization of FFmpeg setup on RHEL 4.6.2 to carryout Encoding, Decoding and Audio/ Video streaming
- Encode the media files with Ffmpeg utilizing multiple codecs and rate control strategy for HEVC encoding.
- Develop an algorithm to Read HEVC file and parse the headers and verify the output using pcap and Elecard HEVC Analyzer.

II. METHODOLOGY

The methodology has been explained through the flow diagram as shown in Figure 1

The code written in C++ takes a sample HEVC file as input,

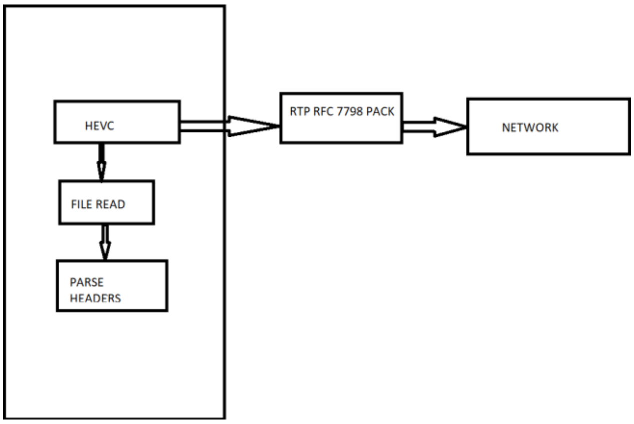

Fig. 1. Flow Diagram

after the file is read, the result is stored in a file in Hexadecimalformat.

1) Using an external tool, such as Elecard HEVC Analyzeror Hex browser, which displays each NAL unit together with its length, offset, Parameter sets such as Video Parameter Set (VPS), Sequence Parameter Set (SPS), Picture Parameter Set (PPS), etc., the Hexadecimal data is first validated byte by byte.

2) Based on the information further down, the result- ing HEVC bitstream can be divided into several cate- gories. The bitstream is a logical packet known as the NAL unit, or Network Abstraction Layer unit, that contains an ordered collection of syntactic components. There are 10 classes and 64 distinct types of network abstraction layer unit types. Understanding the structure of HEVC is crucial for parsing the headers, also known as the Network Abstraction Layer Units. The HEVC structure and the RTP payload have been described in depth below.

A. HEVC STRUCTURE

TABLE I
MAIN HEVC SYNTAX ELEMENTS

VPS	**SPS**	**PPS**	**SLICE**	**SLICE**	**SLICE**	**SLICE**

The syntactic components are arranged in a certain order as shown in the Table I in the HEVC bitstream. Each syntax element is inserted into a network abstraction layer (NAL) Unit, a logical packet. NAL Units come in 64 distinct varieties.Ten classes may be created from these.

Essential video characteristics are incorporated in VPS, SPS, and PPS. They offer a reliable method of transferring the data required for decoding. They may be kept independently or asa component of the bitstream.

The information from the encrypted video frame is includedin the Slice NAL unit. It might include the entire frame or a portion of it. Each slice may be deciphered separately, without relying on information from other slices.

B. NAL Unit Header of HEVC

NAL unit-based bitstream structure is used by HEVC. A NAL unit header is followed by a NAL unit payload in each NAL unit as shown in Figure 2 There are not many header

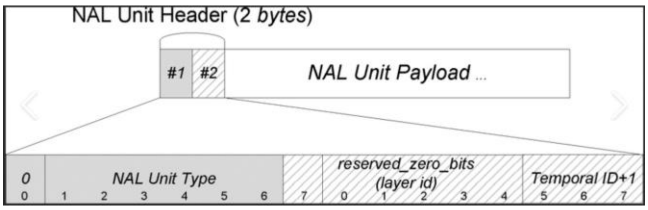

Fig. 2. NAL UNIT HEADER

structural differences between the two standards, AVC and HEVC. The forbidden zero bit must stay at zero. It's donethis way to avoid start code imitations in old MPEG-2 system settings. One bit is utilized to expand the NAL unit type'snumeric range to 64 kinds. The remainder is deemed suitable for future extensions.

The second byte of the NAL unit header comprises two syntaxelement components: reserved zero 6 bits and TID (TemporalID). One bit of it is already a component of the previously specified first byte.

C. RTP PAYLOAD FORMAT

The following RTP header information must be included in compliance with RFC7798's RTP payload format as shown inFigure 3:

1) Marking Bit (1 bit) This bit is set for the access unit's last packet transported in the current RTP stream. This

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 3. RTP PAYLOAD FORMAT

is done to ensure that the payload buffer is handled efficiently.

2) Type of Payload (7 bits) The payload type must beassigned either dynamically or through the profile used.

3) 16-bit Sequence Number (SN) This section is definedand used in accordance with RFC3550.

4) 32-bit timestamp the clock rate is set to 90kHz, and thetimestamp is set to the timestamp of the content.

5) 32-bit synchronization source (SSRC)

SSRC is used to determine the origin of RTP packets. For all components of a single bitstream, a single SSRC is employed. For each RTP stream that comprises asubset of the single bitstream's sublayers, a different synchronization source field is utilized.

III. ALGORITHM TO PARSE THE HEADERS IS AS FOLLOWS:

The basic goal is to extract some nal unit packets (e.g. based on layer id and temporal id), and no modifications to the VPS, SPS, PPS, Slice Header, and so on are required. Following the steps listed below will do that:

1) In the bitstream file, search for the pattern 0x000001, which divides up all the nal units. A 0x00 byte mayalso occur before this pattern if the nal unit after it isthe first nal unit of an access unit (access unit = all nal units for decoding a whole frame).

2) In compliance with the HEVC standard, read the nal unit header and keep/delete the nal units as appropriate. Keep the parameter settings (nal unit types 32, 33 and34 according to Table of the HEVC standard in the RFCdocumentation).

3) Assemble all of the nal units into a new file, ensuring that the 0x000001 sequence is included in between.

IV. RESULTS AND DISCUSSION

The output of FFmpeg after using the command ffmpeg -v error -y -i input.mov -vcodec libx265 transcoded.hevc generates a HEVC video output.

The RTP headers, which comply with RFC 7798, have been added to the binary file after it has been processed by the code. The output is verified using the HEX Browser whichis a tool used to analyse the HEVC file content. The output obtained is shown in the Figure 4. The result of the code matches that of the tool, indicating that the code is accurate.

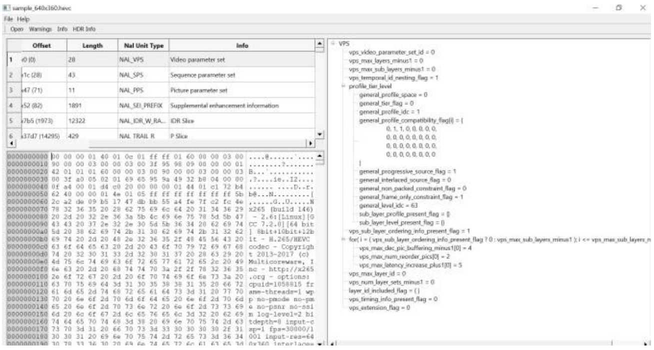

Fig. 4. HEVC Analyser Tool output

CONCLUSION

This research explained the theory and practice of encoding or transcoding media files using the FFMPEG standard. The system satisfies the need for real-time stream analysis and supports a wide range of transmission protocols, media container formats, and video/audio coding standards. Several developments in video coding technology are represented by HEVC. Its video coding layer is based on current block-based motion-compensated hybrid video coding methods, althoughit differs significantly from preceding standards in a numberof important ways. The forthcoming HEVC standard was developed and defined in collaboration with the ITU-T VCEG and ISO/IEC MPEG organizations. In comparison to the performance of earlier standards, the new design's capabilities, when properly implemented, offer around a 50 percent bit-rate savings for equivalent perceptual quality (especially for a high-resolution video).

V. FUTURE SCOPE

Future extensions of HEVC are likely to include scal- able coding, 3-D/stereo/multi-view video coding (the latter of which includes the encoding of depth maps for use with ad-vanced 3-D displays), extended-range formats with increased bit depth and improved color component sampling, and thosethat are already being investigated and prepared by the JCT- parent VC's bodies. On the other hand, the syntactic structure is quite adaptable, and other profiles are anticipated to beadded in the future. The next additions involve consider- able efforts to increase the number of supported video for- mats (including higher-fidelity chroma formats and wider bit depths. Available options include 3D Multiview video, layered coding scalability, and full- resolution chroma.

REFERENCES

[1] J. Ra¨sa¨nen, A. Altonen, A. Mercat and J. Vanne, "Open-source RTP Library for End-to-End Encrypted Real-Time Video Streaming Appli- cations," 2021 IEEE International Symposium on Multimedia (ISM), 2021, pp. 92-96, doi: 10.1109/ISM52913.2021.00023.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] X. Lei, X. Jiang, and C. Wang, “Design and implementation of a real-time video stream analysis system based on ffmpeg,” in 2013 Fourth World Congress on Soft- ware Engineering, IEEE, 2013, pp. 212–216.
[5] R. Sjoberg, Y. Chen, A. Fujibayashi, et al., “Overview of hevc high-level syntax and reference picture management,” IEEE transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1858–1870, 2012.
[6] C. Moreno, E. Crimaldi, V. K. Verma, and M. Huerta, “Study of the effect of security on quality of service on a webrtc framework for videocalls,” in 2021 In- ternational Symposium on Computer Science and Intelligent Controls (ISCSIC), IEEE, 2021, pp. 374–379.
[7] H. Facchini, S. P erez, A. C ardenas, M. Mu noz, and G. Q. Salom on, “Differenti- ated behavior of video traffic in ipv6 multicast networks for ieee 802.11 ac wire- less clients,” in 2019 IEEE CHILEAN Conference on Electrical, Electronics Engi- neering, Information and Communica- tion Technologies (CHILECON), IEEE, 2019, pp. 1–6.
[8] H. Kim, H. Lee, and H. Lim, “Performance of packet analysis between observer and wireshark,” in 2020 22nd International Conference on Ad- vanced Communication Technology (ICACT), IEEE, 2020, pp. 268–271
[9] S. Ahovainio, “Distributed hevc encoding on multi-computer systems,” 2021
[10] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, “Scalable rate control for mpeg-4 video,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 6, pp. 878–894, 2000
[11] L.-Y. Duan, Y. Lou, Y. Bai, et al., “Compact descriptors for video analysis: The emerging mpeg standard,” IEEE MultiMedia, vol. 26, no. 2, pp. 44–54, 2018.
[12] H. Zeng, Z. Zhang, and L. Shi, “Research and implementation of video codec based on ffmpeg,” in 2016 international conference on network and information systems for computers (ICNISC), IEEE, 2016, pp. 184–188
[13] Z. Li, W. Gao, F. Pan, et al., “Adaptive rate control for h. 264,” Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp. 376–406, 2006.
[14] A. Nistico, D. Markudova, M. Trevisan, M. Meo, and G. Carofiglio, “A comparative study of rtc applications,” in 2020 IEEE International Symposium on Multimedia (ISM), IEEE, 2020, pp. 1–8.
[15] C.-H. C. N.-W. Lin, “A development tool for voicexml-based interactive voice re- sponse systems”.
[16] J.-H. Shin and K.-S. Hong, “Simple and powerful interactive e-learning system us- ing vxml: Design and implementation of web and pstn linked efficient learning sys- tem,” in International Conference on Computa- tional Science and Its Applications, Springer, 2006, pp. 354–363
[17] M. Viitanen, A. Koivula, A. Lemmetti, A. Yl a-Outinen, J. Vanne, and T. D. H am al ainen, “Kvazaar: Open-source hevc/h. 265 encoder,” in Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 1179–1182.
[18] J.-R. Ohm and G. J. Sullivan, “High efficiency video coding: The next frontier in video compression [standards in a nutshell],” IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 152–158, 2012
[19] Y. Chen, Z. Wen, J. Wen, M. Tang, and P. Tao, “Efficient software h. 264/avc to hevc transcoding on distributed multicore processors,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 8, pp. 1423–1434, 2014.
[20] F. Andreasen, M. Baugher, and D. Wing, “Session description protocol (sdp) secu- rity descriptions for media streams,” Tech. Rep., 2006
[21] A. B. Johnston, SIP: understanding the session initiation protocol. Artech House, 2015
[22] H. Schulzrinne and S. Casner, “Rtp profile for audio and video confer- ences with minimal control,” Tech. Rep., 2003.
[23] S. Meng, Y. Duan, J. Sun, and Z. Guo, “Highly optimized imple-mentation of hevc decoder for general processors,” in 2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2014, pp. 1–6
[24] D. McGrew and E. Rescorla, “Datagram transport layer security (dtls) extension to establish keys for the secure real-time transport protocol (srtp),” Tech. Rep., 2010.