Noname manuscript No.
(will be inserted by the editor)

IMPLEMENTATION OF 4kUHD HEVC-CONTENT TRANSMISSION

A.Adeyemi-Ejeye · M.Alreshoodi · S.D. Walker

Received: date / Accepted: date

Abstract The Internet of things (IoT) has received a great deal of attention in recent years, and is still being approached with a wide range of views. At the same time, video data now accounts for over half of the internet traffic. With the current availability of beyond high definition, it is worth understanding the performance effects, especially for real-time applications. High Efficiency Video Coding (HEVC) aims to provide reduction in bandwidth utilisation while maintaining perceived video quality in comparison with its predecessor codecs. Its adoption aims to provide for areas such as television broadcast, multimedia streaming/storage, and mobile communications with significant improvements. Although there have been attempts at HEVC streaming, the literature/implementations offered do not take into consideration changes in the HEVC specifications. Beyond this point, it seems little research exists on real-time HEVC coded content live streaming. Our solution fills this current gap in enabling compliant and real-time networked HEVC visual applications. This is done implementing a technique for real-time HEVC encapsulation in MPEG-2 Transmission Stream (MPEG-2 TS) and HTTP Live Streaming (HLS), thereby removing the need for multi-platform clients to receive and decode HEVC streams. It is taken further by evaluating the transmission of 4k UHDTV HEVC-coded content in a typical wireless environment using both computers and mobile devices, while considering well-known factors such as obstruction, interference and other unseen factors that affect the network per-

A.O Adeyemi-Ejeye
Access Networks Laboratory
School of Computer Science and Electronic Engineering
Wivenhoe, Colchester
E-mail: aoteje@essex.ac.uk

M.Alreshoodi
mamalr@essex.ac.uk

S.D Walker
stuwal@essex.ac.uk

formance and video quality. Our results suggest that 4kUHD can be streamed at 13.5 Mb/s, and can be delivered to multiple devices without loss in perceived quality.

Keywords UHDTV · HEVC · Video Streaming · MPEG-2 TS · HLS

1 Introduction

Internet of Things and video streaming applications have generated growing interests in recent years in the computing/networking research community[1–4]. Video transmission and streaming, accounts for a large percentage of internet traffic[5], making it a bandwidth-hungry application. At the same time, significant improvements in video resolution with a major shift towards Ultra-High Definition Television (UHDTV)[6] is aimed at increasing the overall viewing experience. Currently, the UHDTV standard allows two resolutions, namely: 3840 x 2160p (4kUHD) and 7680x4320 (8kUHD), with 4kUHD video content now readily available for broadcast. With the proliferation of Ultra-High Definition (UHD) Video streaming, there is an imminent need to provide a range of transport solutions, especially for broadcast (one-to-many) of coded content. A number of studies have shown the possibilities of streaming beyond HD (BHD) video content. In [7], design requirements for real-time, long-distance uncompressed 4k streaming were proposed. The authors provided three solutions for transmitting 4k video content with the lowest bitrate being 4.59 Gb/s. The authors in [8–11] provided alternative solutions for streaming 4k video content using JPEG 2000 codec. Their solutions discussed the bi-directional of 4k video content at 60Hz using JPEG 2000 multi-layer, scalable coding and a bandwidth of up to 700Mb/s. While in [12], multiple 4k transmission systems are synchronised to produce 8k video resolution at 60Hz. Each 4k frame was compressed using JPEG 2000 at a bit rate of over 400Mb/s; therefore, for the full 8k stream, a bandwidth of over 1.6Gb/s is required. For the transmission of 4k UHD over wireless, authors in [13] discussed the transmission of uncompressed 4kUHD using one of the 60GHz standards (IEEE 802.15.3c), with a maximum transmission distance of 1m. For compressed 4kUHD, the authors in [14] discussed the live streaming of 4kUHD pre-encoded video, while in [15], a solution was proposed for the live streaming of real-time encoded content. Both solutions were at a frame rate of 24Hz and data rate of 20Mb/s. The emergence of the High Efficiency Video Coding standard (HEVC)[16,17] provides approximately 35.4%[18] increase in compression over its predecessor H.264/AVC[19], while maintaining the same level of perceived visual quality and aims to address the bandwidth issue. It appears that very little literature is available that shows or demonstrates the use of compliant broadcast standards for real-time streaming of HEVC. Although Schierl et al [20] discussed the possibilities of integrating HEVC with MPEG-2 TS. While a later version of GPAC[21] has enabled the multiplexing of HEVC coded streams to MPEG-2 TS, its implementation is essentially limited, as it does not consider live streaming of real-time encoded content. In [22] a framework for streaming and evaluating

HEVC content in a loss-prone network using pseudo-RTP was developed. The authors used an obsolescent HEVC Network Abstraction Layer (NAL) header (draft 6[23]), making it no longer compliant with the use of current HEVC coded data. Our major contribution is the design and implementation of a solution for HEVC streaming that realises real-time content transportation. Effective testing of HEVC performance under a wide range of network conditions is then possible and enables interoperability irrespective of hardware and operating system differences (especially at the client side). In particular, a comprehensive solution is defined, which allows for the easy integration of live HEVC encoding for streaming, based on current standardisation efforts. Testing is performed using hardware in a typical office environment that includes the use of peer-to-peer/one-hop scenarios and also the effects of interference and obstruction. Experimental results also provide benchmark performance indicators for 4kUHDTV videos. The rest of our paper is organised as follows. Section 2 deals with related work on the development of HEVC and HEVC streaming solutions; Section 3 describes the proposed streaming solution, while the test environment and implementation are discussed in Section 4. Section 5 provides the extensive results and discussion of our experiments, and section 6 provides conclusion and future work.

2 Related Work

This section describes existing work on the development of HEVC, HEVC streaming solutions and MPEG-2 TS. All features of HEVC and MPEG-2 TS are taken from the I.T.U's HEVC specifications [17] and recommendations in [24]respectively.

2.1 High Efficiency Video Coding (HEVC)

HEVC offers reduction of video bandwidth while maintaining the same quality as compared to its predecessor codec H.264/AVC, while both coding standards share common features such as the use of both, a video coding layer (VCL) and network abstraction layer (NAL). At the inception of the standardisation of HEVC, it was projected to improve compression over its predecessor codec by at least 50%. However, with experiments performed in [18] suggests that HEVC Main profile (MP) bit-rate savings vary between two major scenarios (interaction and entertainment) with regards to coding efficiency. Interaction applications show an average of 40% in bit-rate savings, while entertainment applications show an average bit-rate savings of 35.4% based on objective video quality evaluations, and 49.3% average bit-rate savings for perceived video quality; both in comparison to the H264/AVC High profile (HP) using high-definition (HD) content. This was achieved using the maximum coding unit size for luma permitted in HEVC MP of 64 x 64 in the VCL, and is also beneficial for higher resolution videos as well as video sequences with sparse contents. At the NAL of HEVC, the same concept as in H.264[19] applies; though with modifications. In previous HEVC NAL header specifications, the forbidden field was 1bit while its **nal-unit-type** was 6bits; this was the NAL header used in the framework developed by the authors in [22] and is no longer in force. Amendments in Draft 8[25], show the use of **nal-ref-id** field and the removal of reserved bits field, which was replaced with **nuh-layer-id** (6 bits) and the **nuh-temporal-id-plus1**(3 bit temporal level indicator); these changes are still in force in Draft 10[25].

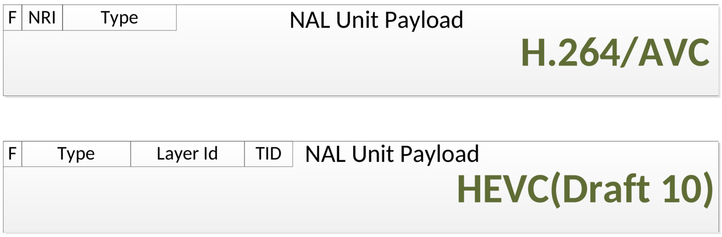


Fig. 1: Comparison between H.264/AVC and current HEVC NAL units

Fig. 1. shows the comparison between the NAL units of H.264/AVC and the current HEVC standard. Similar to H.264 NAL units, HEVC NAL units also provide an extension header, under certain circumstances. The extension header which consists of a **nuh-layer-id** field anticipated for scalable and 3D video coding and a **nuh-temporal-id-plus1** field.

2.2 MPEG-2 Transmission Stream

MPEG-2 systems[24], provide two layers of packetisation for any of its transport streams. The first layer of packetisation produces the packetised elementary stream (PES), which is obtained by the encapsulation of coded video, audio, and data elementary streams (ES) or bitstreams. For video (and audio), the encapsulation is done by the sequential separation of the elementary streams into access units. Each PES packet contains data from only one elementary stream; therefore, an audio stream cannot share the same PES with a video stream. The second layer of encapsulation produces the transport streams (TS) which are used for transmission. These streams have fixed length subdivisions of the PES packets (as data payload) with its additional header information. A TS packet is 188 bytes in length. An illustration of a TS packet multiplexing process is seen in fig. 2.

The header is normally 4-byte long and begins with a synchronisation byte of 0x47 followed by an optional adaptation field and more information on the other flags is in the MPEG-2 systems specifications. Schierl et al [20] describe the possibilities and advantages of integrating HEVC with MPEG-2 TS using the standard target decoder receiver model(STD) without any form

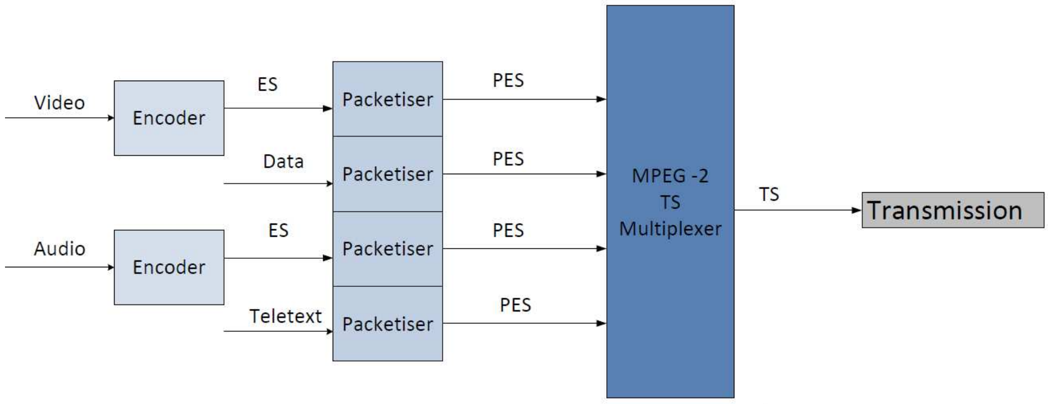


Fig. 2: MPEG-2 TS Multiplexing and Transporting

of implementation, and GPACs[21] real-time implementation of MPEG-2 TS streaming for pre-encoded content. This can be seen in the experimental implementation described by the authors in [26] for 4k scalable HEVC (SHVC) video transmission over UDP. Although their work provides insights into the streaming of 4k content, it is limited as only offline MPEG2-TS encapsulated data can be streamed in real time. This work bridges the gap by demonstrating a formal realisation strategy for delivery of HEVC video content with up-to-date specifications using MPEG-2 TS encapsulation, and transmitting over user datagram protocol (UDP). The choice of using UDP only transportation (UDP/IP/Ethernet)is as a result of the 8 bytes extra overhead in comparison with a real-time transport protocol (RTP/UDP/IP/Ethernet)[27]. This work deals with the 13-bit Packet Identifier (PID) which is used to uniquely identify the HEVC PES (from its corresponding ES). Since MPEG-2 TS streaming solution is widely available, the well-known open source FFMPEG[28] implementation of this standard is extended to recognise and encapsulate HEVC streams as recommended in [29]. The choice of FFMPEG is based on its wide acceptance in the broadcast industry for live encoding and streaming.

2.3 HTTP Live Streaming (HLS)

HLS provides a similar solution to MPEG-DASH by breaking down an overall stream into segments and transported over HTTP. The protocol enables both, the transportation of video and audio, for playback on a wide-range of devices such as phones, tablets, desktop computers, and more recently, smart TVs. It also supports live and on-demand transmission, adaptive bit-rate streaming and media encryption. The streaming architecture for HLS consists of three parts:

Server : This is responsible for providing the input stream to the distributor. It generates the encoded video/audio data, encodes it and encapsulates it, into the suitable format.

Distribution : This is responsible for responding to client requests and delivery of the media content, and its associated resources. All of this can be achieved using a standard web server.

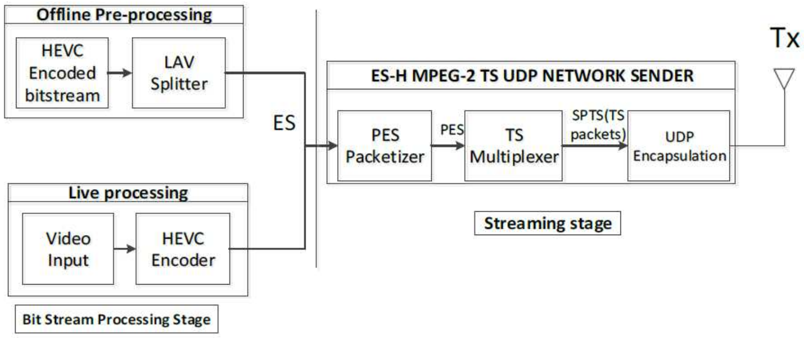
Client : The responsibility of requesting, downloading, reassembling, decoding and display of the appropriate content lies with the client.

In a typical scenario, an audio-video input is taken by their respective encoders, the encoded data is then encapsulated into MPEG-2 transport streams[24]. This is then segmented into a series of short media files by the segmenter. The segmenter also creates and maintains an index file that contains the list of media files, and its uniform resource locator (URL) is then published on the web server. The client can then request and read the media files in the order presented, decode, and display them.

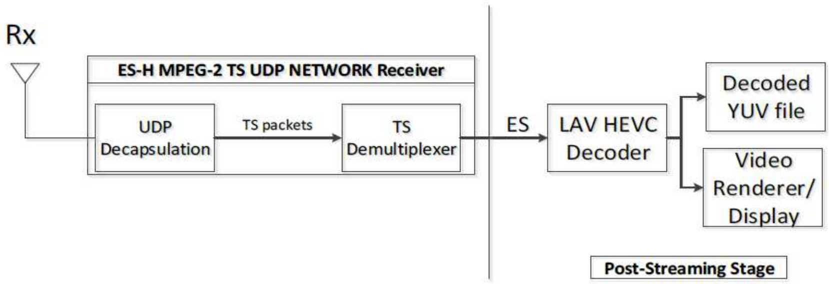
While the transmission of UHD H.264/AVC[19] coded content has been documented in [14, 15] and in [30] which discusses a multi-platform adaptive bitrate broadcast solution for 4kUHD streaming using Real Time Messaging Protocol (RTMP), HTTP Live Streaming (HLS) protocol and HTTP Dynamic Streaming (HDS) protocol. However, the H.264 standard limits its maximum input resolution for encoding to 4096x2160. With the standardisation of High Efficiency Video Coding standard (HEVC)[16, 17], the input video resolution for encoding can now be up to 8k video resolution. HEVC also provides approximately 35.4%[18] increase in compression, while providing the same level of visual quality in comparison with its predecessor codec H.264/AVC. With regards to HTTP transmission, the MPEG-DASH (Dynamic Adaptive Streaming over HTTP) protocol[31] has been considered for the transmission of both HEVC[32] (terrestrial and broadcast) and 3D-HEVC [33], and its visual quality evaluated in previous studies [34–36]. MPEG-DASH functions by breaking the content into sequences of HTTP-based file segments, with each segment containing a short duration of content for playback. Although MPEG-DASH aims to solve the non-standardisation problem of HTTP transport of video and audio, it seems it is still at the early stages of adoption and poses a challenge to the deployment of 4kUHD over-the-top(OTT) content[37]. At the same time, the well established non-standardised HTTP Live Streaming[38] (HLS) protocol offers similar solutions to MPEG-DASH.

3 Design of the Proposed Framework

The proposed framework for HEVC encoded streams (ES-H), illustrated in fig. 3, is designed to enable system benchmarking for real-time HEVC streaming and evaluation (objective and subjective), using a broadcast and widely used standard. It also provides the flexibility of experimental evaluation under a wide range of network conditions and hardware preferences. ES-H is an experimental streaming environment, where HEVC elementary streams can be converted into PES interleaved time stamped streams and then transmitted over a network, all in real-time. ES-H, unlike any other framework, takes into consideration the use of realistic systems and the problems associated with them. Firstly, there is a bit stream processing stage where the video is encoded either live or off-line pre-processing; in both cases the ES is provided. In streaming stage, the server-side operation processes MPEG-2 TS encapsulation and transmits over UDP, while the client-side operation receives the packets and processes the MPEG-2 TS decapsulation to reproduce the distorted ES. The post-streaming stage decodes the received HEVC ES and either renders the decoded video or saves it to file for evaluation.



(a) Tx



(b) Rx

Fig. 3: Proposed ES-H Framework

3.1 Bit Stream processing stage

The bit stream processing stage consists of two steps: encoding and the extraction of elementary streams. The first step is split into two categories, where video can either be encoded offline (offline pre-processing) or can be encoded in real-time (live encoding). For both categories, the FFMPEG co-compiled version of LibX265 [39] was used.

In the offline pre-processing category, a raw video sequence is pre-encoded, using specified encoding parameters. In modern computers, the file source reads data without any prior knowledge of the data. This presents a problem, as it can provide an incorrect format for the streaming. The authors, therefore, adopts the use of the open-source LAV[40] HEVC media demultiplexer module, which was modified by redefining media subtype and using the macro definition. The media demultiplexer module was used to identify the HEVC elementary streams, its video properties, such as frame rate and sends the video data to the streaming module.

3.2 Streaming Stage

The streaming stage consists of four steps: PES packetisation, TS multiplexing, transmission and reception. In other to transmit the HEVC content using MPEG-2 TS, the packet ID (PID) for it needs to be defined according to current standards using an integrated platform, that enables both live encoding and pre-encoded content for real-time transmission. In this proposed framework, the HEVC-coded stream is transmitted with the stream ID 0x24. To begin with, the HEVC elementary streams (ES) are firstly packetised into PES streams and then sent to the TS multiplexer for encapsulation in TS packets. In the TS multiplexer, the PID of the PES packets are compared to the entries in the program map table (PMT) which contains information about the program (elementary stream) for identification (as illustrated in fig. 4.). Since the HEVC PES format was not available to FFMPEG, its libraries were extended by defining its PES format and PMTs descriptor flags were defined, enabling the detection and encapsulation of HEVC coded streams into TS packets. In addition, the single program transmission stream (SPTS) was adopted since the focus was only on video transmission. Seven MPEG-2-TS packets were then streamed over UDP due to the maximum transmission unit (MTU) available. This innovative solution has been verified and is now available as a patch in the FFMPEG repository [41,42].

3.3 Post-processing stage

For the post-processing stage, the STD as implemented by FFMPEG was used to receive the MPEG2-TS streams, de-multiplex them based on its PID and feed the HEVC bit-streams to the HEVC decoder. This enabled validation of

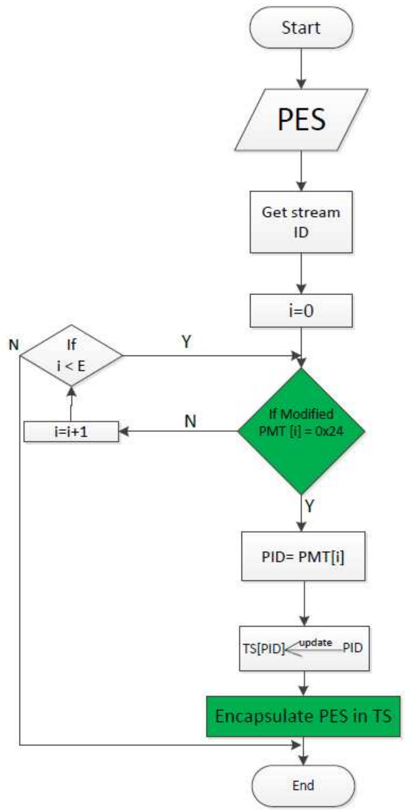


Fig. 4: TS Multiplexing and Transportation

the HEVC MPEG2-TS stream format and also ascertained the flexibility and interoperability of the ES-H MPEG-2 TS sender. The decoder can either decode the frames into a YUV or can decode them for display using its rendering applications.

4 Implementation

4.1 Hardware Implementation

The proposed ES-H system has been fully tested on in a realistic environment with commercial, off-the-shelf (COTS) computer hardware. The environment used was a typical office space where performance factors are uncontrollable. All software components are written in either C++ (Directshow filters) or C (amended MPEG-2 TS multiplexer). Directshow filter compatibility is currently limited to operating systems[43]; however, the amended MPEG-2 TS multiplexer can be implemented on other platforms with very few or no modifi-

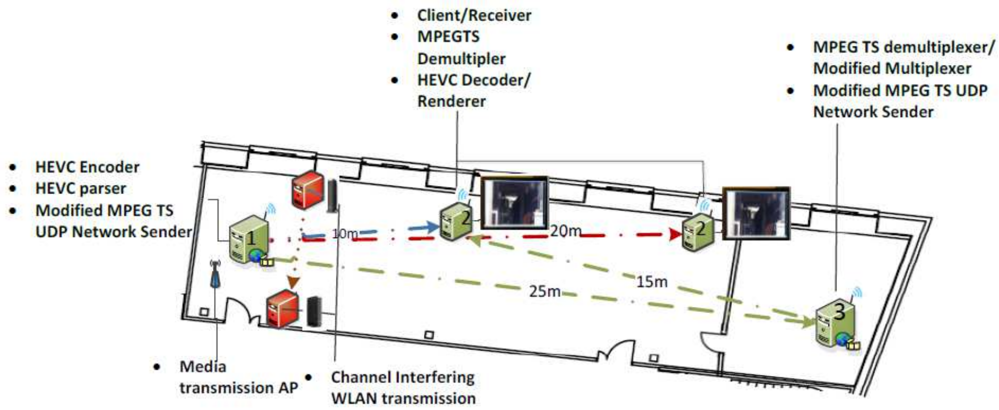


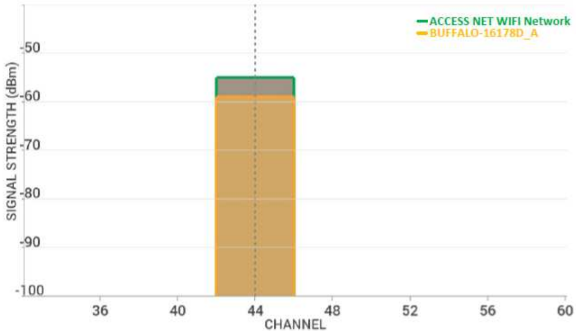
Fig. 5: Implementation Design

cations. Fig. 5, shows the design implementation for testing the HEVC MPEG-2 TS transmission. Typical computer configurations for nodes 1 and 3 were used, while node 2 is a high-powered machine which enabled 4kUHD video display. For 10m and 20m, the connection involved increasing the range between server (node 1) and client (node 2), while in the one-scenario, a 30ms buffer was added in node 3 to reduce packet loss, as without it severe packet loss was noticed. Since objective video quality metrics were used, the decoder was configured to work in real-time, and saved decoded YUV files for evaluation; therefore it decodes as the packets are being received. An 802.11ac Wi-Fi (Access net WIFI Network) AP[44] operating within the 5GHz frequency was used, while connections to the AP were enabled using USB 2.0 802.11ac dongles[45]. To introduce interference, another AP (Buffalo AC1300 [46])was used and it functioned within the same channel (Channel 44). To verify the evidence of interference, an openly available Wifi monitoring tool[47] was used for observation. The results are presented in fig. 6. Before each transmission, an initial data-rate measurement was taken using [48].

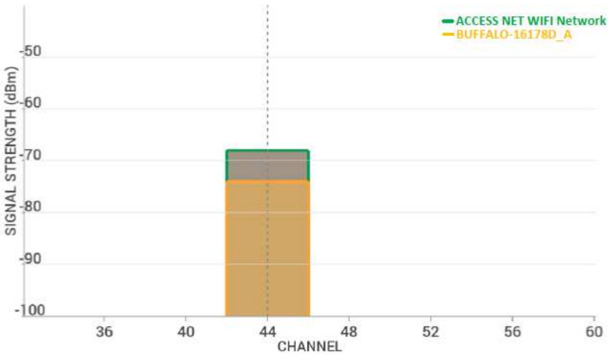
The effect of observed interference in fig. 6 can be seen when correlated with the available bandwidth in fig. 7. Since the maximum transmission unit (MTU) for wireless local area network (WLAN) is approximately 1500 bytes and MPEG-2 TS packet size is 188 bytes, the UDP protocol was set to carry seven MPEG-2 TS packets of 1316 bytes.

4.2 Video Samples and Configuration

Since reduction of bandwidth usage for BHD is one of the major issues HEVC hopes to address, as already discussed in section 2.1, four test sequences varying in motion and scene complexity were used. These test sequences were sourced from [49](Foreman, News and Coast) and [50] (Sintel 4k). Table 1 shows the video sequences classification, based on their spatial information



(a) 10 m



(b) 20 m

Fig. 6: Observation of direct interference at (a) 10 m (b) 20 m

(SI) and temporal information (TI) indices on the luminance component of each content, as indicated in [51] to determine the level of motion. The parameters for the 4k UHD encoding can be seen in table. 2.

Table 1: Video Sequences SI and TI

Video Sequences	SI	TI
Sintel	16.3897	72.2639
Foreman	19.7101	38.2870
Coast	10.8370	16.9183
News	17.5219	21.2441

The choice of bitrate is based on the proposed average bit-rate saving of 35.4% in comparison to H.264/AVC, which has shown the possibility of compressing 4kUHD to 20Mb/s ABR rate-control.

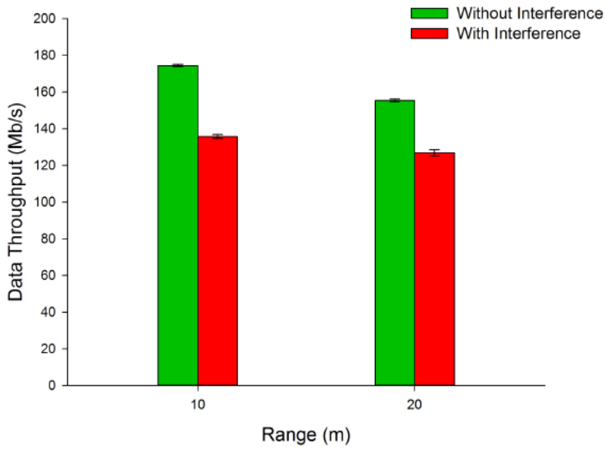


Fig. 7: Available Throughput at 10 and 20 metres respectively

Table 2: 4kUHD Video HEVC encoding parameters

Parameter	**Value**
Profile	Main
Rate Control	Average Bit-Rate
Coding Tree Block	64 x 64
Group of Pictures (GOP)	25(25Hz) 30(30Hz) 60(60Hz)
Frame Rate	25Hz 30Hz 60Hz
Bit-rate	13Mb/s (25Hz) 15Mb/s (30Hz) 20Mb/s (60Hz)

4.3 4kUHD HLS End-to-End Video Streaming

To provide a scalable solution, the MPEG2-TS encapsulated video streams were received by a conventional web server and segmented with a segment length of 10 seconds, a playlist entry size of 10, and a maximum 10 segment files to wrap on disk, to generate video fragments and its index file (.m3u8). For the video encoding, two scenarios were considered. The first, using pre-encoded video content and the second, live encoding from a video capture device using the following parameters in table 3, while the full implementation can be seen in fig. 8.

Table 3: HEVC encoding parameters

Parameter	**Value**
64 x 64 Profile	Main
Rate Control	Average Bit-Rate
Frame Rate	25fps
Group of Pictures (GOP)	25
Bit-rate	10Mb/s

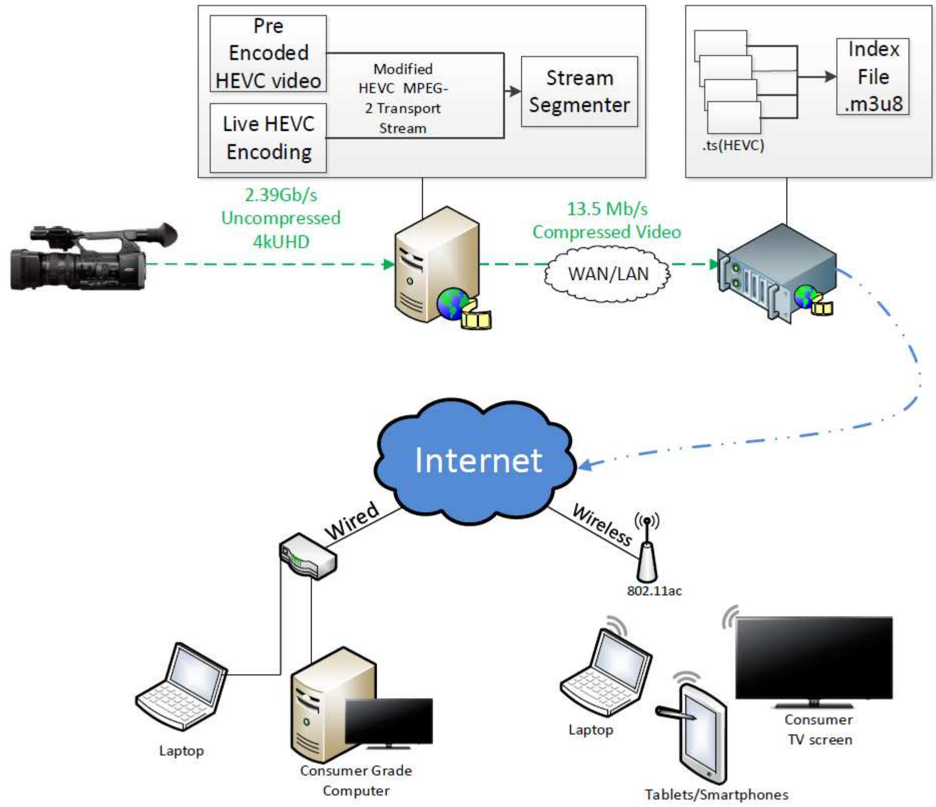


Fig. 8: HLS Implementation Design

In the pre-encoded scenario, test sequences previously discussed in section 4.2 were reused. While in the live encoding scenario, video was captured using a SONY AX1 4kUHD[52] camera and BlackMagic Decklink capture card[53] at 25Hz and was piped to FFMPEG using BlackMagic devices tools[54]. A 16-core processing server was used to retrieve the video and encode. The maximum coding tree unit was also varied for the inter-prediction process to investigate its impact on the computational load.

5 EXPERIMENTAL RESULTS

All results shown in this section are based on average values obtained during the experiments; each experiment was conducted ten times and during office hours. Initial bandwidth measurements were taken for both scenarios. As seen in fig. 7, the average throughput measured shows expectation of minimal packet loss since the data rate of the coded content is only 13 Mb/s. It is also assumed that the total throughput for the one-hop scenario will be less than the first two since there are more than two devices within the same network. Evaluation of the proposed framework with different video sequences varied, based on motion and scene complexity. The metrics used were structural similarity index metric (SSIM), network end-to-end delay (using wireshark[55]) and decoder frame drops. The figures also show the use of error bars based on the standard deviation of uncertainty. This section is split into two subsections.

5.1 4k UHD Streaming

5.1.1 Peer-to-Peer Streaming (P-to-P)

In fig. 9, the video quality of each sequence transmitted at 25Hz, 30Hz and 60Hz is shown. It can be seen that with all video sequences there is a drop in quality, as either the frame rate or distance is increased. While in the cases of 25 and 30Hz, the video quality measurements are acceptable with the lowest mean SSIM value being 0.8864 for the 20m 30Hz sintel stream (with a standard deviation of 0.0158), which is still considered to be fair quality based on its equivalent mean opinion score 3 as recommended in [56]. It was noticed that the major reason for the huge decline in quality for the 60 Hz sequences was due to the decoding complexity presented by the encoded sequences at that frame rate. This can be correlated with results shown in fig. 11, where the inability of the decoder to receive and decode frames at the appropriate rate, due to computational overheads, led to them being dropped/discarded to make way for other frames being received, since decoding was performed in real-time.

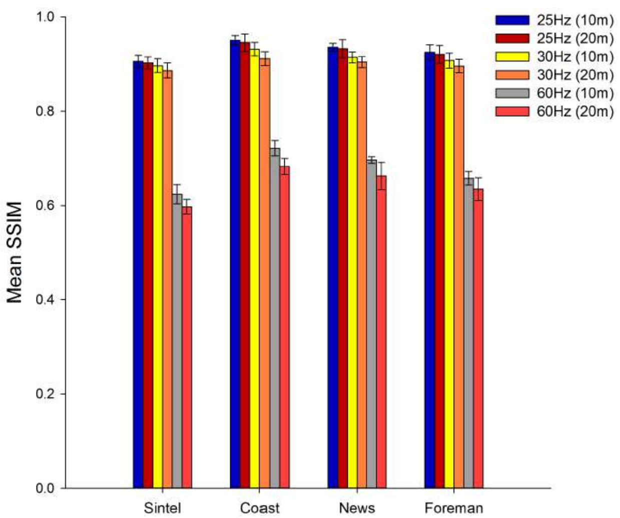


Fig. 9: Video Quality Assessment for P-to-P Experiments

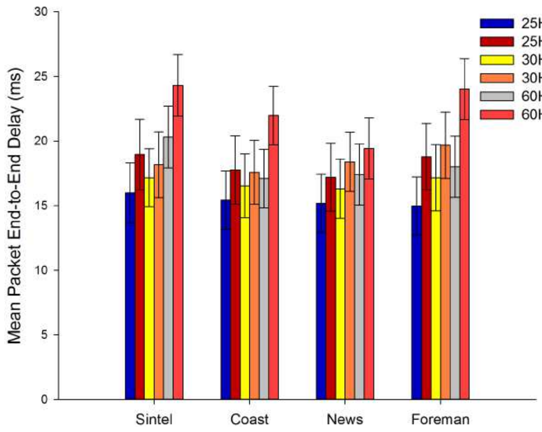


Fig. 10: Average Delay for P-to-P Experiments

Fig. 10, shows the performance in terms of end-to-end delay. It can be seen

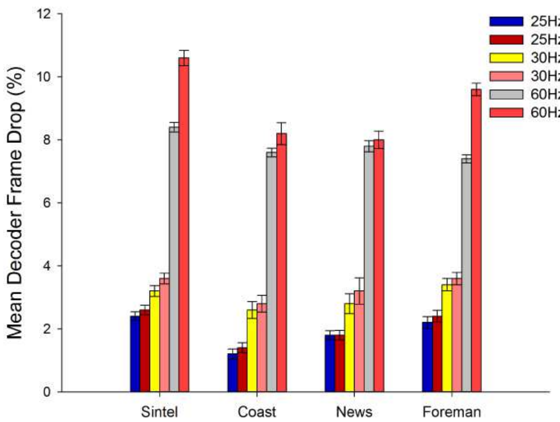


Fig. 11: Average Frame loss for P-to-P Experiments

that there is an increase in end-to-end delay as both frame rate and distance that varied. This can be attributed to the increased time spent by the packets in the outbound queue.

5.1.2 One-Hop Streaming (O-H)

In fig. 12, the video quality of each sequence transmitted at 25Hz, 30Hz, and 60Hz is shown. It can be seen that with all video sequences there is a drop in quality, as either the frame rate or distance is increased. The drop in video quality is caused by the same issue raised in subsection 5.1.1, where the

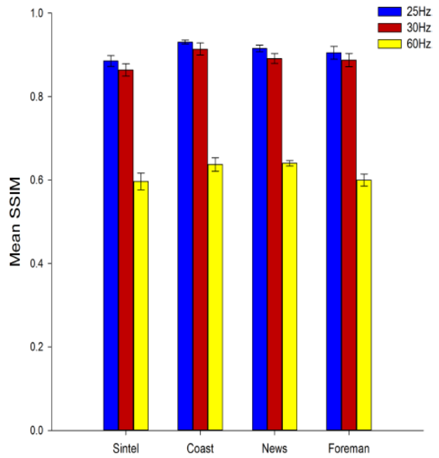


Fig. 12: Video Quality Assesment for O-H Experiments

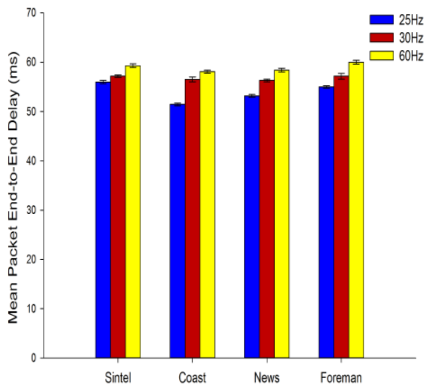


Fig. 13: Average Delay for O-H Experiments

decoder could not decode received frames at the appropriate rate, and therefore opted for dropping the frames instead. This can also be correlated with fig. 14.

Fig. 13, shows the performance in terms of end-to-end delay. It can be seen that there is an increase in end-to-end delay as both frame rate and distance are varied. In comparison with fig. 11, it can be seen that the end-to-end delay is higher. This can be attributed to the introduced 30 ms virtual buffer at the inbound queue of Node 3 (fig. 5). The reason for this was in a typical computer, there is only one queue which serves both inbound and outbound transmission for WLAN transmissions; therefore, the creation of this virtual buffer was to avoid any competition between inbound and outbound packets.

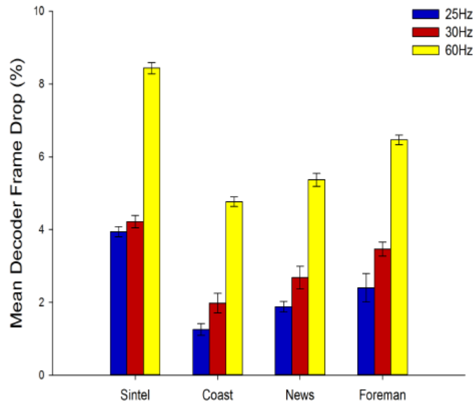


Fig. 14: Average Frame loss for O-H Experiments

5.2 4kUHD HLS based streaming

5.2.1 Pre-Encoded Video Streaming

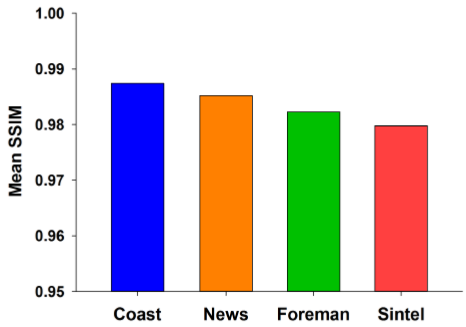


Fig. 15: Video Quality Assessment

All results shown in this section are average values obtained during the experiments; each experiment was conducted ten times and during office hours. Evaluation of the proposed solution was done with different video sequences that varied in motion and scene complexity. To determine the video quality, the video was decoded and stored in a consumer grade computer, while the SSIM metric was used to determine the video quality. In Fig. 15, the video quality of each of the sequences transmitted is shown. Based on the recommendations in [56], the video sequences all fall in to the equivalent mean opinion score is 4, which is considered to be good quality.

5.2.2 Live-Encoded Video Streaming

The results in table. 4, suggests that the processing power needed for HEVC encoding would be much higher. Furthermore, it can be estimated that HEVC real-time encoding for live transmission on a 64-core processor, using a maximum CTU of 64 x 64 will only enable a frame rate of up to 16Hz, while 32 x 32 and 16 x 16 with enable up to 32Hz and 48Hz respectively, when such a system is used as a dedicated encoder.

Table 4: Coding Tree Unit(CTU) vs Achievable Average Frame rate

CTU	Average (Hz)
64 x 64	4
32 x 32	8
16 x 16	12

6 Conclusion and Future Work

This paper has presented a novel design and implementation framework for HEVC streaming using MPEG-2 TS, based on the most recent specifications. This framework also provided significant insights into streaming 4k UHDTV video HEVC encoded content at a low bitrate of 13.5 Mb/s.

In addition, it provides insights into another method for HEVC transmission over HTTP using a well known protocol; based on our solution. This has been tested and initial results suggest the good video quality. It also discusses the challenges of transmitting UHDTV HEVC live-encoded content, which would require either a high-powered system or more investigations into its coding algorithm.

By using hardware in a typical wireless environment and introducing interference while taking into account performance reduction factors such as obstruction, multiple connected devices, we have considered what is practically obtainable. From the results presented, it can be seen that although channel interference does have an impact on the 802.11ac network bandwidth performance, its effect does not affect the video quality adversely, as the available bandwidth still enables the transmission of HEVC-coded 4k UHDTV video. Our results also point out that current software implementations for HEVC decoders struggle to decode 4kUHD content at 60Hz.

This paper presents insights into the realistic deployment of HEVC video content for broadcasting at low bitrates due to its flexibility, and therefore, provides a favourable solution to IoT applications. Our future work will focus on the development and evaluation of live encoders and also the optimisation of decoders to enable high refresh rates and improve the overall experience in transmitting UHD HEVC-coded content. We also intend to consider the effect of video bitrate on power consumption, especially for hand-held devices.

References

1. A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, p. 25, ACM, 2011.
2. L. Plissonneau and E. Biersack, "A longitudinal view of http video streaming performance," in *Proceedings of the 3rd Multimedia Systems Conference*, pp. 203–214, ACM, 2012.
3. V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, "Vivisecting youtube: An active measurement study," in *INFOCOM, 2012 Proceedings IEEE*, pp. 2521–2525, IEEE, 2012.
4. J. Guo, E. W. Wong, S. Chan, P. Taylor, M. Zukerman, and K.-S. Tang, "Performance analysis of resource selection schemes for a large scale video-on-demand system," *Multimedia, IEEE Transactions on*, vol. 10, no. 1, pp. 153–159, 2008.
5. I. Cisco, "Cisco visual networking index: Forecast and methodology, 2011–2016," *CISCO White paper*, vol. 2016, 2011.
6. E. Nakasu, "Super hi-vision on the horizon: a future tv system that conveys an enhanced sense of reality and presence," *Consumer Electronics Magazine, IEEE*, vol. 1, no. 2, pp. 36–42, 2012.
7. J. Halák, M. Krsek, S. Ubik, P. Žejdl, and F. Nevřela, "Real-time long-distance transfer of uncompressed 4k video for remote collaboration," *Future Generation Computer Systems*, vol. 27, no. 7, pp. 886–892, 2011.
8. T. Margolis, S. Brown, T. Cornish, H. Bracho, M. Stanton, T. C. M. de Brito Carvalho, F. F. Redigolo, F. C. De Castro, K. Kaneko, J. De Almeida, *et al.*, "Tri-continental premiere of 4k feature movie via network streaming at file 2009," *Future Generation Computer Systems*, vol. 27, no. 7, pp. 924–934, 2011.
9. T. Shimizu, D. Shirai, H. Takahashi, T. Murooka, K. Obana, Y. Tonomura, T. Inoue, T. Yamaguchi, T. Fujii, N. Ohta, *et al.*, "International real-time streaming of 4k digital cinema," *Future Generation Computer Systems*, vol. 22, no. 8, pp. 929–939, 2006.
10. D. Shirai, T. Yamaguchi, T. Shimizu, T. Murooka, and T. Fujii, "4k shd real-time video streaming system with jpeg 2000 parallel codec," in *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pp. 1855–1858, IEEE, 2006.
11. T. Fujii, M. Kitamura, T. Murooka, D. Shirai, and A. Takahara, "4k & 2k multi-resolution video communication with 60 fps over ip networks using jpeg2000," in *Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium on*, pp. 167–170, IEEE, 2009.
12. M. Kitamura, D. Shirai, K. Kaneko, T. Murooka, T. Sawabe, T. Fujii, and A. Takahara, "Beyond 4k: 8k 60p live video streaming to multiple sites," *Future Generation Computer Systems*, vol. 27, no. 7, pp. 952–959, 2011.
13. A. O. Ejeye and S. D. Walker, "Uncompressed quad-1080p wireless video streaming," in *2012 4th Computer Science and Electronic Engineering Conference (CEEC)*, 2012.
14. A. O. Adeyemi-Ejeye and S. D. Walker, "Ultra-high definition wireless video transmission using h. 264 over 802.11 n wlan: Challenges and performance evaluation," in *Telecommunications (ConTEL), 2013 12th International Conference on*, pp. 109–114, IEEE, 2013.
15. A. O. Adeyemi-Ejeye and S. Walker, "4kuhd h264 wireless live video streaming using cuda," *Journal of Electrical and Computer Engineering*, vol. 2014, p. 2, 2014.
16. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.
17. ITUT, "High efficiency video coding(10/2014)," *ITU-T H*, vol. 265, 2014.
18. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standardsincluding high efficiency video coding (hevc)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012.
19. ITUT, "Advanced video coding for generic audiovisual services(itu-t rec. h. 264— iso/iec 14496-10:2011 avc)," *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050*, 2011.

20. T. Schierl, M. M. Hannuksela, Y.-K. Wang, and S. Wenger, "System layer integration of high efficiency video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1871–1884, 2012.
21. J. Le Feuvre, C. Concolato, and J.-C. Moissinac, "Gpac: open source multimedia framework," in *Proceedings of the 15th international conference on Multimedia*, pp. 1009–1012, ACM, 2007.
22. J. Nightingale, Q. Wang, and C. Grecos, "Hevstream: a framework for streaming and evaluation of high efficiency video coding (hevc) content in loss-prone networks," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 404–412, 2012.
23. H. H. E. V. Coding, "text specification draft 6," *JCTVC-H1003*, 2011.
24. ITU-T/ISO/IEC, "H.222.0 : Information technology - generic coding of moving pictures and associated audio information," *ITU-T/ISO/IEC*, vol. 265, 2012.
25. B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (hevc) text specification draft 8," *JCTVC-J1003, July*, 2012.
26. W. Hamidouche, G. Cocherel, J. Le Feuvre, M. Raulet, and O. Deforges, "4k real time video streaming with shvc decoder and gpac player," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*, pp. 1–2, IEEE, 2014.
27. B. Bing, *3D and HD broadband video networking.* Artech House, 2010.
28. F. Bellard, M. Niedermayer, *et al.*, "Ffmpeg," *http://ffmpeg. org*, 2012.
29. ITU-T/ISO/IEC, "H.222.0 : Recommendation h.222.0 (10/14)," *ITU-T/ISO/IEC*, vol. 265, 2014.
30. L. Clift, A. O. Adeyemi-Ejeye, G. Koczian, S. D. Walker, and A. Clark, "Delivering live 4k broadcasting using today's technology," *International Broadcast Convention*, 2014.
31. ISO/IEC/23009-1:2014, "Information technology-dynamic adaptive streaming over http (dash)-part 1: Media presentation description and segment formats," *ISO/IEC MPEG, Tech. Rep*, 2012.
32. 3GPPTS26.247-V11.2.0, "Technical speci cation - 3rd generation partnership project; technical speci cation group services and system aspects," *Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic AdaptiveStreaming over HTTP (3GP-DASH)*, 2013.
33. B. Oztas, M. T. Pourazad, P. Nasiopoulos, I. Sodagar, and V. Leung, "A rate adaptation approach for streaming multiview plus depth content," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 1006–1010, IEEE, 2014.
34. Q. Wang, J. Nightingale, J. M. Alcaraz-Calero, C. Luo, Z. Pervez, X. Wang, and C. Grecos, "H. 265 video streaming in mobile cloud networks," *Resource Management of Mobile Cloud Computing Networks and Environments*, p. 238, 2015.
35. I. Irondi, Q. Wang, and C. Grecos, "Subjective evaluation of h. 265/hevc based dynamic adaptive video streaming over http (hevc-dash)," in *IS&T/SPIE Electronic Imaging*, pp. 94000B–94000B, International Society for Optics and Photonics, 2015.
36. I. Irondi, Q. Wang, and C. Grecos, "Empirical evaluation of h. 265/hevc-based dynamic adaptive video streaming over http (hevc-dash)," in *SPIE Photonics Europe*, pp. 91390L–91390L, International Society for Optics and Photonics, 2014.
37. J. Ozer, "Dash vs. hls: Request for comments," 2014.
38. R. Pantos and W. May, "Http live streaming, ietf draft," 2010.
39. D. Buitenhuis, "Libx265 encoder," *http://anzwix.com/a/Libav/AddLibx265Encoder*, 2014.
40. H. Leppkes, "Lav filters 0.62," *https://github.com/Nevcairiel/LAVFilters/releases/tag/0.62*, 2014.
41. A. Adeyemi-Ejeye, "Hevc mpeg ts definition in mpeg ts header," *https://github.com/FFmpeg/FFmpeg/pull/60/files*, 2014.
42. A. Adeyemi-Ejeye, "Hevc mpeg ts: Allowing hevc video using ts stream type 0x24," *https://github.com/olugbone/FFmpeg/commit/071069b29109ab726daac9e4*, 2014.
43. Microsoft, "Microsoft windows operating system," *http://windows.microsoft.com/en-us/windows/home*, 2014.
44. Apple, "Airport extreme 802.11ac," *https://www.apple.com/uk/airport-extreme/specs/*, 2014.
45. Netgear, "A6200 802.11ac wifi usb adapter," 2013.

46. N. Matsuura, "Wireless lan device and controlling method thereof," May 13 2014. US Patent 8,724,476.
47. ManageEngine, "Free wifi monitor plus android app," *http://www.manageengine.com/wifi-monitor-plus-android-app/free-wi -monitor-plus-android-index.html*, 2014.
48. Totusoft, "Lan speed test (lite) v1.3," 2013.
49. Elemental, "4k test sequences," 2013.
50. Blender-Foundation, "Sintel 4k," 2011.
51. T. Installations and L. Line, "Subjective video quality assessment methods for multimedia applications," *Networks*, vol. 910, p. 37, 1999.
52. Sony, "Sony fdr-ax1 4k professional handycam," 2011.
53. BlackMagic-Design, "Decklink 4k extreme," 2006.
54. BlackMagic-Design, "Blackmagic devices tools," 2011.
55. G. Combs *et al.*, "Wireshark," *Web page: http://www. wireshark. org/last modified*, pp. 12–02, 2007.
56. T. Zinner, O. Hohlfeld, O. Abboud, and T. Hoßfeld, "Impact of frame rate and resolution on objective qoe metrics," in *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pp. 29–34, IEEE, 2010.