System Layer Integration of High Efficiency Video Coding

Thomas Schierl, *Member, IEEE,* Miska M. Hannuksela, *Member, IEEE,* Ye-Kui Wang, and
Stephan Wenger, *Senior Member, IEEE*

Abstract—This paper describes the integration of High Efficiency Video Coding (HEVC) into end-to-end multimedia systems, formats, and protocols such as Real-time transport Protocol, the transport stream of the MPEG-2 standard suite, and dynamic adaptive streaming over the Hypertext Transport Protocol. This paper gives a brief overview of the high-level syntax of HEVC and the relation to the Advanced Video Coding standard (H.264/AVC). A section on HEVC error resilience concludes the HEVC overview. Furthermore, this paper describes applications of video transport and delivery such as broadcast, television over the Internet Protocol, Internet streaming, video conversation, and storage as provided by the different system layers.

Index Terms—Broadcast, DASH, High Efficiency Video Coding (HEVC), HTTP streaming, Internet streaming, JCT-VC, media access, MPEG-2 TS, network transport, Real-time transport Protocol (RTP), storage.

I. INTRODUCTION

THE EMERGING High Efficiency Video Coding (HEVC) standard [1], also known as ISO/IEC MPEG part 2 video and anticipated as ITU-T Rec. H.265, is expected to provide the next big step in video coding efficiency. It is generally believed to reduce the bitrate compared to the H.264/AVC [3] High Profile by 50% at comparable subjective quality [36]. The HEVC standard targets at a wide range of video applications, including high-bitrate entertainment, Internet streaming as well as video conferencing. Similar to H.264/AVC, HEVC is a hybrid video codec based on predictive transform coding and an entropy coding step. As H.264/AVC, also HEVC defines a network abstraction layer (NAL) [2], which provides the primary interface to system level technologies.

In this paper, we highlight the integration of HEVC into system layer technologies, such as Real-time transport Protocol (RTP) [4], MPEG-2 Systems [5], ISO Base Media File Format [6], and MPEG-DASH [7].
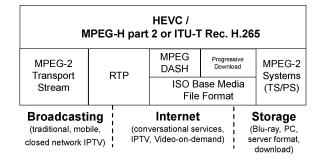
Fig. 1. High level overview of HEVC system layers.

The system interface of HEVC is an important component in the media access chain, and a prerequisite for its successful deployment in various consumer electronics applications. We consider four broad categories of systems: broadcast delivery over digital television and Internet Protocol Television (IPTV) channels including video-on-demand (VoD), streaming over the Internet using the Hypertext Transport Protocol (HTTP), real-time communications such as video conferencing, and store-forward delivery based on a file or a physical memory delivered to the end user (such as: camcorded videos or Blu-ray Disks). To this end, different system technologies have been standardized. These technologies can be summarized as follows.

1) RTP [4], the Internet engineering task force (IETF) protocol for real-time transport over the Internet protocol (IP) is deployed in IPTV as well as in conversational applications such as video conferencing or video chat (see Section IV).
2) MPEG-2 Systems [5] are used by Blu-ray disk storage as well as for transmission of digital TV channels (see Section V).
3) The ISO Base Media File Format [6] and MPEG-DASH [7], for progressive download in VoD applications and HTTP Streaming over the Internet or download applications (see Sections VI and VII).

MPEG is currently developing MPEG Media Transport (MMT) [8], which will be published as part 1 (Systems) of the MPEG-H standard suite. As MMT at this point in time does not specifically address HEVC, we refrain from a discussion of the technologies specified therein.

An overview of the different system technologies and their relation to the four broad application categories (store/forward,

TABLE I

SOME CHARACTERISTICS AND REQUIREMENTS OF VIDEO APPLICATIONS

	Broadcast	Streaming	Video telephone	Camcording and playback	Ultra-low delay
End-to-end latency	No strict requirements	Reasonable for transmission latency	Low ($<150\,$ms)	No strict requirements	Ultra-low ($<40\,$ms)
Decode-output delay	Reasonable for channel switching	Reasonable for startup delay	Very low ($<40\,$ms)	Reasonable for startup latency	Ultra-low ($<<16\,$ms)
Random access	Frequent access points needed for channel switching	Relatively frequent access points needed for seeking	Intra coded pictures only needed for endpoints joining a multiparty call and recovery after severe transmission errors	Relatively frequent access points needed for seeking	At startup of session, in multi-party sessions, required for late joiners
Trick modes	Yes (for providing trick modes of recorded broadcasts)	Yes	Not relevant	Yes	Not relevant
Bitrate adaptation	Not relevant	Yes, for real-time delivery over best-effort networks. Realized through stream switching or scalable coding.	Yes, to match the transmitted bitrate to downlink throughput. In multiparty calls, realized through transcoding or scalable coding. In point-to-point calls, realized through bitrate control in encoding.	Not relevant	Not relevant/depends on use case: for remote applications over best-effort Internet, bitrate adaptation is required.
Error robustness	Forward error correction in the protocol stack. Forward error control and detection in video coding and error concealment in video decoding.	Congestion avoidance through bitrate adaptation. Robust packet scheduling. Retransmission.	Forward error control in video coding and error concealment in video decoding. Feedback-based encoding for preventing error propagation.	Not relevant	Not relevant/depends on use case: for remote applications over best-effort Internet, error robustness is required.

broadcast, Internet streaming and conversational applications) is given in Fig. 1.

This paper is organized as follows. In Section II, we briefly characterize the protocol stacks commonly used in video applications. Section III recapitulates the high-level syntax of HEVC. The remainder of this paper provides details of the integration of HEVC into various protocol hierarchies, such as RTP (see Section IV), MPEG-2 transport stream (TS) (see Section V), ISO Base Media File Format (see Section VI), and MPEG DASH (see Section VII).

II. VIDEO APPLICATIONS AND SYSTEM LAYER OVERVIEW

Video applications may be categorized, for example, into: broadcast (television), streaming, video conferencing, and store-and-forward types of applications such as camcording and file playback. In broadcast, the same stream is (potentially) transmitted to a large number of users, which usually do not have a feedback channel to the encoder or transmitter. Streaming applications over IP may be further categorized according to whether a unicast or multicast connection is used, whether live encoding is performed or pre-encoded content is provided, whether a feedback channel is available, and which type of network topology is used, e.g., if proxy servers or media gateways are involved. Both broadcast and streaming typically allow some amount of latency as long as the initial startup time remains reasonable. In contrast, the end-to-end latency in video conferencing must be kept as small as possible. Due to the low end-to-end latency requirement, the mechanisms for robustness against transmission errors in video

telephone applications are typically somewhat different (often source-coding based) compared to those in broadcast and streaming applications (which are often system layer based). In camcording, the captured video sequence is encoded and encapsulated to a multimedia file together with audio and some metadata. When the multimedia file is played back, the audio and video bitstreams are extracted from the file and provided to the respective decoders.

The application families introduced in the previous paragraph have varying characteristics, which impose diverse requirements for video encoding and decoding. Table I lists some of the key differentiating factors between the mentioned applications that have to be taken into account during the video codec design to facilitate applications. Subsequently in this paper, we refer to the presented factors to highlight the key differences between HEVC and previous codec designs from a systems integration perspective. Besides the more traditional use cases of broadcast, streaming, video telephony, and camcorder/playback, the use case referred to as ultra-low delay has been identified as required for live TV contribution, automotive, or remote applications such as gaming. We anticipate that such use cases will become increasingly important with today's Internet access technologies such as LTE and fiber to the home.

The mentioned application families can be realized through different formats and protocols. For example, either the RTP or the HTTP [10] may be used for streaming. While the applications differ in their requirements and characteristics, there are commonalities between their protocol stacks. Some of the most commonly used protocol stacks are presented in

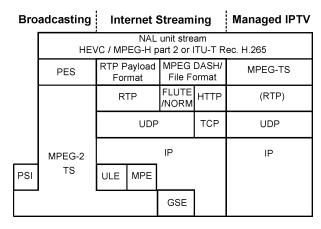| Broadcasting	Internet Streaming			Managed IPTV
NAL unit stream HEVC / MPEG-H part 2 or ITU-T Rec. H.265				
PES	RTP Payload Format	MPEG DASH/ File Format		MPEG-TS
	RTP	FLUTE /NORM	HTTP	(RTP)
	UDP		TCP	UDP
MPEG-2 TS	IP			IP
PSI	ULE	MPE		
		GSE		

Fig. 2. HEVC system layer stacks.

Fig. 2. Here, we summarize in addition to Fig. 1 the different delivery formats used in digital video broadcasting (DVB) [11] digital TV protocol stacks, IETF Internet real-time streaming stack as well as video-on-demand Internet delivery.

In the left part of the figure, the traditional digital broadcast approach is shown as used in DVB—S/S2, C/C2, and T/T2 systems via satellite, cable, and terrestrial transmission, respectively. The middle part of the figure shows the IP-based streaming which is typically used in the conversational and Internet streaming scenarios. A mixture of the two aforementioned approaches, broadcast and IP-based streaming, is shown in the right part of the figure and is used in DVB's managed or closed IPTV system specifications. Since all three approaches include MPEG-2 TS and/or the RTP protocol, we discuss these two protocols in more detail in the following sections. The file format as discussed in Section VI is used for file delivery but even more for MPEG DASH over HTTP (in Section VII), IP broadcast channels using file delivery over unidirectional transport (FLUTE) [12], as well as storage and recording. FLUTE is the selected protocol in 3GPP for delivery of DASH-formatted media over multicast/broadcast channels [34].

III. HEVC OVERVIEW

In this section, we provide an introduction to HEVC and its relevant features for system layer integration. The section includes the following:

1) introduction of the new codec features of HEVC in Section III-A;
2) motivation for extended high-level syntax relative to H.264/AVC shortcomings in high-level syntax in Section III-B;
3) important HEVC high-level syntax features in Section III-C;
4) discussion of HEVC error resilience in Section III-D;
5) summary of high-level parallelization techniques of HEVC in Section III-E.

A. HEVC Design Overview

HEVC [1] and H.264/AVC [3] share a similar hybrid video codec design. Conceptually, both technologies include a video coding layer (VCL) and an NAL. The VCL includes all low-level signal processing, including block partitioning, inter and intra prediction, transform-based coding, entropy coding, loop filtering, and so on. The NAL encapsulates coded data and associated information into NAL units, a format that facilitates video transmission and application systems.

As H.264/AVC, an HEVC bitstream consists of a number of access units, each including coded data associated with a picture that has a distinct capturing or presentation time. Each access unit is divided into NAL units, including one or more VCL NAL units (i.e., coded slice NAL units) and zero or more non-VCL NAL units, e.g., parameter set NAL units or supplemental enhancement information (SEI) NAL units. Each NAL unit includes an NAL unit header and an NAL unit payload. Information in the NAL unit header can be (conveniently) accessed by media gateways, also known as media aware network elements (MANEs), for intelligent, media-aware operations on the stream, such as stream adaptation.

An important difference of HEVC compared to H.264/AVC is the coding structure within a picture. In HEVC, each picture is divided into coding tree units (CTUs, also referred to as treeblocks) of up to 64×64 luma samples. CTUs can be recursively split into smaller coding units (CUs) using a generic quad-tree segmentation structure. CUs can be further split into prediction units (PUs) used for intra and inter prediction and transform units (TUs) defined for transform and quantization. HEVC includes integer transforms for a number of TU sizes.

Predictively coded pictures can include uni-predicted and bi-predicted slices. The flexibility in creating picture coding structures is enhanced compared to H.264/AVC. The VCL in an HEVC encoder generates, and in an HEVC decoder consumes, syntax structures known as slices. One purpose is to adapt to the maximum transmission unit (MTU) sizes commonly found in IP networks, irrespective of the size of a coded picture. Picture segmentation is achieved through slices. Additionally, a new slice concept has been specified as the dependent slice fragment, which allows for fragmentation of slices at treeblock boundaries. In contrast to regular slices, i.e. independent slice fragments, dependent slice fragments, do not break coding dependencies, and their use, therefore, incurs only a very limited coding efficiency penalty. In an error-free environment, an encoder can choose to fragment a coded picture in potentially many small units and provide them to the network transmission stack before having finished encoding the remainder of the picture, and without incurring the penalties of broken in-picture prediction. One use-case of dependent slice is reduction of end-to-end delay for ultra-low delay applications.

For ultra-low delay operations, HEVC furthermore includes a novel concept known as decoding units, which describes parts of the picture as subpictures. HEVC's hypothetical reference decoder (HRD) model can include the description and timing of decoding units.

As in H.264/AVC, in-loop deblocking filtering is applied to the reconstructed picture. HEVC also includes a new in-loop filter that may be applied after the deblocking filtering: sample adaptive offset (SAO).

Another important difference of HEVC compared to H.264/AVC is the availability of coding tools that are specifically designed to enable processing on high-level parallel architectures. Slices, as in both HEVC and H.264/AVC, may be used for parallel-processing purpose. Besides slices, two new high-level parallel processing tools, namely Tiles and Wavefront Parallel Processing (WPP) are available. In combination with dependent slices, WPP also supports ultra-low delay applications.

B. High-Level or Systems Shortcomings of H.264/AVC

Operational experience with H.264/AVC has shown a number of shortcomings of that standard and/or its system layer interface. Specifically, H.264/AVC does not efficiently support open groups of pictures (GOPs, introduced below), and its support for temporal scalability has certain shortcomings.

In H.264/AVC, a coded video sequence is defined as one instantaneous decoding refresh (IDR) access unit followed by non-IDR access units up to the next IDR access unit. Random access to a bitstream is guaranteed only at IDR pictures. However, an H.264/AVC encoder can also code certain pictures as intrapictures starting an open GOP by limiting pictures following such an intrapicture in output order not to reference pictures before the intrapicture (in decoding order), thereby creating a random access point. However, a H.264/AVC decoder has no mechanism (outside a recovery point SEI message) to learn the existence of such entry points and is not required by the standard to be able to start decoding from an intrapicture starting an open GOP. Furthermore, the use of IDR pictures for providing random access points leads, in many cases, to less efficient coding compared to that achieved with open GOPs (e.g., 6%, as reported in [13]). Consequently, in practice many systems requiring periodic random access capability ended up to having suboptimal compression efficiency for H.264/AVC bitstreams due to relatively frequent use of IDR access units.

Use cases for hierarchical temporal scalability were identified during the design of H.264/AVC [14] but the full compression efficiency potential of hierarchical temporal scalability was not identified until later [15], [16]. Consequently, the hierarchical temporal scalability for H.264/AVC was originally designed by assuming fairly shallow hierarchies and was realized through the sliding window reference picture marking mechanism [17]. In H.264/AVC, deep temporal scalability hierarchies, e.g., typically more than four temporal levels, require the use of long-term reference pictures, adaptive reference picture marking, and reference picture list modification. Adaptive reference picture marking is somewhat vulnerable to picture losses [2], and hence the use of decoded reference picture marking SEI messages is recommended in error-prone environments. Furthermore, H.264/AVC includes no identification of the temporal level in NAL unit header or slice header, as the subbitstream extraction for temporal scalability was not considered a normative feature of the standard at the time of developing the specification.

When the scalable extension, referred to as Scalable Video Coding (SVC) [3], of H.264/AVC was designed, the temporal scalability was regarded as one of the scalability types inher-

ently supported. Thus, an indication of the temporal level, the temporal_id syntax element, was introduced in the NAL unit header extension for SVC. However, the NAL units specified for H.264/AVC do not contain the temporal_id syntax element. Thus, the process to extract a subset of the bitstream, e.g., for bitrate adaptation remained complex, as some NAL units had to be parsed beyond the NAL unit header to determine whether they must be removed from or kept in the bitstream subset.

In conclusion, H.264/AVC and SVC enable hierarchical temporal scalability to the same extent as HEVC. However, the use of deep temporal hierarchies in H.264/AVC and SVC requires quite sophisticated encoding and error robustness. Furthermore, H.264/AVC does not provide any directly accessible indication of the temporal layering of the bitstream.

C. High-Level Syntax

In this section, we review those high-level syntax aspects of HEVC that differ significantly from H.264/AVC, starting with the mechanisms addressing the shortcomings identified in the previous section, followed by other changes.

In HEVC, open-GOP random access points are directly signaled in the NAL unit header, instead of relying on recovery point SEI messages as in H.264/AVC. A picture that is an open-GOP random access point is signaled by a distinct NAL unit type, and the picture is named a clean random access (CRA) picture or a broken-link access (BLA) picture.

IDR pictures, CRA pictures, and BLA pictures are collectively referred to as random access point (RAP) pictures. A conforming bitstream may start with a RAP picture of any type (in H.264/AVC, a conforming bitstream has to start with an IDR picture only). Therefore, an HEVC-based system can rely on decoders supporting bitstreams starting with CRA or BLA pictures and thereby open GOPs, leveraging the improved coding efficiency of open GOPs.

BLA pictures are typically originated from bitstream splicing wherein a CRA picture at the splicing point is changed to be a BLA picture. HEVC specifies one type of CRA pictures and three types of BLA picture. A CRA picture may have associated random access skipped leading (RASL) pictures or random access decodable leading (RADL) pictures or both. Similarly, the first type of BLA pictures may have associated RASL pictures or RADL pictures or both. The second type of BLA pictures does not have associated RASL pictures but may have associated RADL pictures. The third type of BLA pictures does not have associated RASL or RADL pictures.

When random access is performed starting from a CRA or BLA picture, the associated RASL pictures, if any, should be skipped as an inter prediction reference picture may not be available in the bitstream, while the associated RADL pictures, if any, are correctly decodable. RASL pictures and RADL pictures are leading pictures, which precede the associated RAP picture in output order but follow it in decoding order.

Furthermore, HEVC specifies two types of IDR pictures, one without associated RADL pictures, and one that may have associated RADL pictures. Therefore, there are altogether six types of RAP pictures specified in HEVC. These different types of RAP pictures can be straightforwardly mapped to the different types of stream access points (SAPs) specified
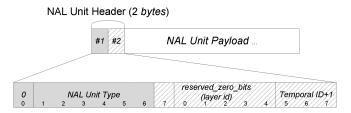
Fig. 3. HEVC NAL unit interface.

in [6] for use with DASH [7], see Section VII for more details.

The support for temporal scalability in HEVC has been improved compared to the mechanisms in H.264/AVC. Supported are both temporal layer signaling and stream adaptation through subbitstream extraction. All NAL units use the same two-octet-long NAL unit header as shown in Fig. 3, which avoids parsing dependencies with profile information in MANEs for temporal layer access. Further layer information is jointly provided by NAL unit header and the video parameter set (VPS) as described below. The NAL unit header includes a three-bit temporal_id_plus1 field, which indicates the temporal sublayer of the NAL unit. Using this field, MANEs can easily identify the temporal layer of any NAL unit, and use this information for stream adaptation.

Other high-level syntax aspects of HEVC that we consider significant enough to mention here are as follows.

The two-byte NAL unit header includes a 6-bit reserved_zero_bit filed, which is intended to be used by extensions such as a future scalable and 3-D video extension for indicating the layer_id. These 6 bits will carry spatial/SNR layer or view type identification information further specified by the VPS, as shown below. As this field is always present in the NAL unit header, one significant shortcoming of SVC, the lack of this information in an H.264/AVC-compliant base layer, can be avoided, although this field will be ignored by the base HEVC codec.

Temporal sublayer access (TSA) pictures were introduced to indicate temporal layer switching points, so to avoid inefficient signaling mechanism through an SEI message used in SVC. TSA pictures are also signaled by a distinct NAL unit type. Furthermore, step-wise temporal sublayer access (STSA) can be signaled.

In addition to sequence parameter set (SPS) and picture parameter set (PPS), both inherited from H.264/AVC, a so-called VPS is introduced in HEVC. The VPS was included to address identified shortcomings of the H.264/AVC scalable and multilayer extensions. In those extensions, the only data structure that provides an overview of a layering structure (or its 3-D equivalent) is available in the so-called scalability information SEI message. For HEVC, it was felt that an SEI message is not the best possible option to convey such critical information as layer dependencies or layer-specific profile and level information. Accordingly, the VPS includes such information, in the form of fixed length binary code words that allow straightforward processing by network elements. It is anticipated that with the advent of a scalable or 3-D extension to HEVC, the VPS will be populated with most, if not all, of the information the scalability information SEI

message used to provide. For the base HEVC codec, the VPS serves as general higher layer interface to an HEVC bitstream providing general information about profiles, maximum levels and similar information.

The reference picture buffer management has been redesigned for improved error resilience. Instead of relying on the decode-state-based sliding window or adaptive reference picture marking mechanism to maintain the status of reference pictures in the decoded picture buffer, for each picture a direct signaling mechanism named reference picture set (RPS) is applied. With this mechanism, the decoder does not need the reference pictures status of a previous picture in decoding order to derive which reference pictures are to be kept in the decoded picture buffer for inter prediction reference purpose. As part of this mechanism, the processes for generating and handling of nonexisting pictures that were needed for temporal scalability support in H.264/AVC become unnecessary in HEVC and are therefore not present in the HEVC standard.

The reference picture list construction (RPLC) process has also been improved. In HEVC, RPLC is based on RPS signaling, the reference picture list initialization process is unified for P slices and B slices, and no reordering of pictures according to the output order of pictures is needed in the reference picture list initialization process. When a different reference picture list than the initial one is needed, the final list is directly signaled, instead of being modified from the initial list in H.264/AVC.

Some features related to error resilience or stream adaptation as well as other features of H.264/AVC, such as slice groups (i.e., flexible macroblock order—FMO), redundant slices, arbitrary slice order (ASO), data partitioning, and SP/SI pictures, are not included in HEVC, due to their rare deployment in real-world applications.

D. Error Resilience of HEVC

In many applications, error robustness means outside source (de)coding should be used as indicated in Table I. However, error resilience supported by the video codec is always an important feature, especially if the system layer uses unreliable transport as typically in video telephony and video conferencing scenarios. In this section, we review the error resilience characteristics of HEVC.

The essential tools, such as slices, are unchanged from H.264/AVC. Slicing allows to produces VCL NAL units fitting into a network packet, while having almost no coding dependencies to other slices of the same picture, thus the decoding process may be resilient to a loss of a slice. Many other error resilience tools of H.264/AVC, such as FMO, ASO, redundant slices, data partitioning, and SP/SI picture as mentioned before, have been removed due to their rare usage. H.264/AVC included a few SEI messages for encoder-assisted error detection and concealment, out of which the scene information SEI message has been retained in HEVC. Among other things, the message assists decoders in detecting scene cuts and gradual scene changes and hence selecting the type of the error concealment method accordingly if slice losses have occurred [18].

Due to highly tuned coding tools and prediction mechanisms, error concealment may cause unpredictable impacts in HEVC [19]. The use of error concealment should be carefully considered in implementations and is a topic for further research works. While error concealment might be a riskier approach to take with HEVC than with some other codecs, HEVC provides a good toolset for coping with transmission errors and a basic level of error resilience even in basic standard-conforming decoders as explained in the following paragraphs.

Similarly to earlier video coding standards, the HEVC specification consists of the syntax and the semantics of the HEVC NAL units and bitstream as well as the decoding process of error-free bitstreams. Loss resilience has been conventionally considered as an implementation-specific feature. However, as the possibilities of the decoder to cope with transmission errors depends on the way the encoder used the tools affecting error robustness, a basic level of loss resilience can only be achieved with mandatory syntax elements and decoding processes—a principle which was considered during the HEVC standardization process particularly when it comes to handling of reference pictures as explained in the next paragraph.

The H.264/AVC design includes decoded reference picture marking according to specific memory management control operation (MMCO) commands included in the bitstream. A correct decoding operation requires decoders to maintain a state machine of reference picture marking according to MMCO commands, and a loss of even a single MMCO command can lead to unpredictable impacts. While well-designed H.264/AVC codec implementations are able to tackle this vulnerability gracefully through the use of error robustness features provided by H.264/AVC, it is not for granted that both the encoder and the decoder in a system support these features. Hence, in HEVC both encoders and decoders mandatorily apply the RPS feature for decoded reference picture marking. Consequently, HEVC decoders are always able to detect reference picture losses reliably.

Temporal scalability is better supported thanks to the NAL unit header signaling and may be used to limit temporal error propagation. For example, if a slice was lost in a picture having temporal_id equal to 2, all pictures having temporal_id equal to 0 and 1 can still be correctly decoded. Temporal scalability also gives a way to avoid error concealment as follows. If a slice was lost in picture having temporal_id equal to M, where $M > 0$, the decoder can choose to skip decoding of subsequent pictures having temporal_id equal to or greater than M until the next RAP picture. The decoder may also increase the number of decoded and displayed temporal layers at each TSA or STSA picture. A TSA picture enables up-switching, at the TSA picture, to the sub-layer containing the TSA picture or any higher sub-layer, from the immediately lower sub-layer. An STSA picture enables up-switching, at the STSA picture, to the sub-layer containing the STSA picture, from the immediately lower sub-layer.

HEVC includes two novel SEI messages, which can help in error resilience. First, the decoded picture hash SEI message contains a checksum derived from the decoded samples of the associated picture, hence enabling error detection. Second, the structure of pictures (SOP) description SEI message describes the inter prediction and temporal structure of the bitstream. An SOP is defined as one or more coded pictures consecutive in decoding order, in which the first coded picture in decoding order is a reference picture having temporal_id equal to 0 and no coded picture except potentially the first-coded picture in decoding order is an IDR or CRA picture. It is usually advantageous to identify repetitive prediction hierarchies in the bitstream and specify SOPs accordingly. The SOP description SEI message resides in the first access unit of an SOP and describes the temporal structure and the inter prediction hierarchy of the SOP comprehensively. The encoder can generate the SOP description SEI messages, whenever it uses a regular inter prediction pattern. MANEs, such as multipoint control units (MCUs) in video conferencing, can use SOP description to conclude if a full picture has been lost in the uplink and send feedback to the encoder faster than any decoder-based loss detection would be able to. Furthermore, SOP description SEI messages enable MANEs and decoders to conclude the temporal propagation of a slice loss and hence determine the best error handling strategy. For example, error concealment may be used if there is no or very short temporal error propagation, while another error handling strategy could be chosen if the temporal error propagation is severe.

E. Parallelization Tools

While the decoder complexity increase of HEVC over H.264/AVC is reportedly comparatively modest, the encoder complexity is substantially higher, as many additional coding modes have been added. To address this complexity issue, for the first time in any video compression standard, tools have been included that specifically address high-level parallelization. These tools are known as Tiles and Wavefront Parallel Processing (WPP), and both have been included in the HEVC Main profile. Although the tools have been justified mostly as facilitators for encoder parallelization, both can also be used for parallel implementations of a decoder, as long as the decoder can rely on a bitstream that is known to include a sufficient number of Tiles or WPP. Therefore, additional signaling is required, as discussed in Section IV-B3 and Section V-C. Wavefronts and Tiles may have different use cases.

When Tiles are used, the picture is divided into rectangular groups of treeblocks separated by vertical and horizontal boundaries. These boundaries break all dependencies so that a tile can be processed independently, but some filtering (such as deblocking and SAO) may be applied afterward to control boundary artifacts. Therefore, Tiles address hardware architectures in which the bandwidth between the processors or cores executing the encoder/decoder is restricted, as well as software-based architectures where the cores or processors are difficult to synchronize on a treeblock level without incurring delay or processor underutilization. Since Tiles allow boundary filtering to cross tile boundaries, Tiles may also require intercore communication depending on the actual implementation. Furthermore, tiles may also be applied in environments where MTU size matching is important (i.e., in error prone IP environments), since they are in principal

robust to losses due to the breaking of coding dependencies at Tile boundaries similar to slice boundaries. Therefore, an example use case for Tiles may be video conferencing over lossy IP channels.

In WPP, rows of treeblocks may be encoded or decoded in parallel while preserving all coding dependencies. Since a treeblock being processed requires the left, top-left, top, and top-right treeblocks to be available in order for predictions to operate correctly, a shift of two treeblocks is enforced between consecutive rows of treeblocks processed in parallel. Therefore, WPP requires, compared to Tiles in the noncross border filtering mode, additional intercore communication. Typically intercore communication is not a burden for today's multicore processor architectures and WPP is therefore suited for software and hardware implementations. Especially, implementations of WPP are straightforward, since WPP does not affect the ability to perform single step processing, i.e., entropy coding, predictive coding as well as in-loop filtering can be applied in a single processing step. An example use case for WPP may be high-quality streaming over robust channels. In combination with dependent slices, this tool can be also used in ultra-low delay applications. In WPP, a row of treeblocks is referred to as a WPP substream, which can be decoded by a specific processing unit.

For more details, we refer to [35] in this issue.

IV. RTP INTEGRATION

A. RTP Overview

The RTP [4], together with its profiles, extensions, and payload format specifications, form the IETF's framework for real-time media transport. Despite its name, RTP is an application layer protocol in the ISO/OSI sense. The RTP specification includes the definition of two protocols: the RTP itself and the real-time control protocol (RTCP).

While RTP can be used over a number of different transport solutions, it is most frequently used over an unreliable datagram service known as UDP, which, in turn, is conveyed over an IP. RTP facilitates resequencing and loss detection through sequence numbering of RTP packets, synchronization between multiple RTP flows through the RTP timestamp and RTCP Sender Reports, and identification of the payload type. For more details and tutorial level information about RTP, we refer to [20].

For the mapping of complex media data types, for example, video codec data, RTP relies on helper specifications known as RTP payload formats. An RTP payload format, as a minimum, specifies packetization rules of the payload into RTP packets: things like mapping of RTP header fields to events in the media bitstream, points on which the media stream can be broken up into packets, and so on. However, more complex payloads often require specific header information not available in the media codec data itself, which can be conveyed in a payload header that is also specified in the RTP payload format. Newer RTP payload format specifications further include sections related to the signaling of the media in one of the IETF's preferred signaling protocol solutions, and a security considerations section.
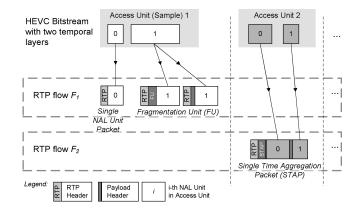

Fig. 4. HEVC over RTP with two temporal layers using session multiplexing.

As all nongeneric aspects of an RTP media transport for a given media codec, such as HEVC, are specified in the RTP payload format specification, the remainder of this section describes the first drafts of such a specification [21].

B. RTP Payload Format for HEVC

1) *Media Transport:* The RTP payload format for HEVC [21] reuses many of the concepts of the RTP payload specification for H.264/AVC, namely RFC 3984 [22] and its successor RFC 6184 [23]. We expect the reader to be somewhat familiar with at least one of these specifications.

For the media transport, the payload format [21] offers the following options.

1) A single NAL unit can be conveyed in an RTP packet. The NAL unit header co-serves as the RTP payload header.
2) Multiple NAL units with identical time stamps (i.e., belonging to the same picture/access unit can be aggregated into a single RTP packet, and their boundaries are identified through an aggregation payload header.
3) An NAL unit can be fragmented into two or more fragments, each of which is transferred in its own RTP packet. The boundaries of the fragments are identified through a fragmentation header.

The options mentioned above are a subset of those of RFC 3984 and RFC 6184. The support for aggregation packets that cover multiple pictures has been removed. The reason for this step is twofold: first, the multitime aggregation packet type (MTAP) of RFC 3984 has seen little, if any, practical use. Second, HEVC is generally believed to be employed with considerably larger picture sizes (at least on average) than H.264/AVC. Even considering the better coding efficiency of HEVC when compared with H.264/AVC, a typical HEVC coded picture is likely approaching the common MTU size seen on the Internet (of approximately 1500 octets).

The HEVC payload format offers two modes of operation, non-interleaved and interleaved, both inherited from RFC 3984 and RFC 6184. In the non-interleaved mode, both the packet transmission order and the NAL unit encapsulation order in an aggregation packet follow the decoding order of NAL units. The non-interleaved mode suits for example

low-delay conversational applications. In the interleaved mode, the transmission order of NAL units may differ from their decoding order, and consequently the receiver is required to deinterleave received NAL units back to their decoding order prior to passing them to a decoder. The interleaved mode may be used for example in streaming applications, where its primary use case is robust packet scheduling that facilitates continuous playback even when the network throughput fluctuates unexpectedly. All the above-mentioned packet types can be used with both the modes.

An example for transport of an HEVC temporal scalable bitstream in two different RTP sessions, also known as session multiplexing, is shown in Fig. 4. The shown bitstream contains two temporal layers, where every second access unit belongs to the second temporal layer. The usage of the Single NAL unit, Single Time Aggregation and Fragmentation Unit packets of the non-interleaved packetization mode is shown in the figure. Session multiplexing will be also a feature to differentiate between layers or views in the scalable or 3-D extensions of HEVC.

2) Signaling Support Similar to RFC 3984: At the time of writing, the signaling support of the HEVC payload specification draft [21] is not fully developed yet; insofar, the information provided in this section should be verified against the final specification.

We expect that the signaling support will follow roughly the outline of what is available in RFC 3984 [22]. RFC 3984 allows signaling basic codec parameters, such as profile and level, maximum values for bitrate, framerate, buffer parameters, and signaling of payload-internal parameters.

Equivalent to RFC 3984, there will be support for the out-of-band transport of parameter sets in both declarative and offer-answer signaling contexts. Signaling conveyed parameter sets can co-serve as a precise description of the operation point an encoder is able to produce and/or a decoder is able to consume. This allows for a finer granularity of operation point selection than merely signaling the profile and level—a feature that has seen real-world use in an RFC 3984 context. The new VPS will have a special role in SDP signaling, since this single parameter set allows for accessing relevant stream information for high level applications. Therefore, it should be straightforward, possibly in addition to the BASE64 encoded version used for other parameter sets, to also place a text representation of the VPS content (or a subset thereof) into the SDP, which makes it directly accessible to SDP parsers.

3) Support for Parallelization Tools: HEVC includes support for high-level parallelization in the form of Tiles and WPP coding tools.

Tiles and WPP allow a parallel encoder or decoder to designate its multiple cores, processors, or other independent processing units, to be assigned to certain spatial parts of the picture. In the following, we focus on decoder-level parallelization. The issues with signaling can probably be best explained using an example.

Assume an encoder has created a bitstream of 1080p60 resolution that includes four equally sized Tiles. In declarative use, the bitstream would be announced at a certain level x.y depending on the maximum bitrate. Assume further, a decod-

ing system is incapable of decoding level x.y using a single core, but capable of decoding if using four cores. (Given the complexity of HEVC decoding process and today's processor architectures, this is not an unreasonable assumption). Without having information in the signaling available indicating that there are four Tiles or WPP that allow splitting the decoding/reconstruction load onto four cores, the decoder would have to indicate that it is incapable of decoding the bitstream.

Two aspects have to be specified in the RTP payload format to support such a scenario for Tiles. First, the payload format requires a parameter, in the form of a single integer, of the maximum number of cores/processors the bitstream is tailored to support through Tiles. Second, while there are restrictions on the Tile layout in HEVC to prevent evil bitstreams, the encoder is still fairly flexible in deciding its Tile layout. Accordingly, without further information, it is not known that all Tiles available in a bitstream are of (at least approximately) the same size. The RTP payload format, on the other hand, announces parallel decoding capability not in the unit of Tiles per picture, but cores required for processing Tiles. Restrictions are included in the payload format requiring that the spatial area assigned through Tiles to a given core cannot be more than a certain percentage of the whole picture's spatial area divided by the number of cores. Accordingly, the encoder is still free in its Tile layout, but the encoding system has to signal the number of cores based on approximately similar spatial areas for which each core is responsible for decoding. Preference was given to such an approach over relying on a receiver/decoder to determine its capability from the Tile layout as available in an SPS. One key reason has been that the content of an SPS can change during the lifetime of a negotiated session, as sequences in the video coding sense can be quite short: a few hundred milliseconds to a few seconds in most random-access cases. Decoder capabilities, on the other hand, do not change.

For using WPP in a scenario as discussed above, just signaling of the activation of the tool is sufficient to let a parallel decoder determine whether a bitstream with a higher level than the maximum supported level without using parallelization is decodable. The reason is that WPP has a higher scalability in parallelization as Tiles, i.e. one core per treeblock row can be applied. The minimum number of supported cores can be directly derived from the picture height assuming the maximum treeblock size being 64×64. Therefore, the RTP payload format will contain in the SDP parameters the indication of the activation of WPP in a bitstream. In this context, HEVC foresees a parameter indicating a spatial segmentation of a picture in the video usability information (VUI).

Furthermore, the interleaved packetization mode of the HEVC payload format may be further used to transmit parallel partitions of a parallelized HEVC bitstream, such as Tiles or WPP substreams, as they get ready from the different processing cores. This may be of advantage if ultra-low delay processing is required in very large pictures, such as 8 k while not missing the encoder and decoder high-level parallelization feature, where multiple network packets are expected per parallel partition.

V. MPEG-2 TS INTEGRATION

MPEG-2 TS [5] is used globally for digital television broadcast and optical disk storage. It defines the multiplexing and synchronization of compressed video and audio data. The standard is capable of carrying several different formats that support media services. This section describes the core mechanisms for delivery of HEVC coded media using MPEG-2 TS in a wide variety of applications including cable delivery, terrestrial broadcast, packaged media, and closed-network IPTV.

A. MPEG-2 Systems Overview

MPEG-2 systems standard is a joint specification of ISO and ITU-T. This specification addresses the combination of one or more (compressed) elementary streams of video and audio, as well as other information/data, into one or more streams which are suitable for storage or transmission. In addition, the specification provides information to enable synchronized decoding of the multimedia information over a wide range of retrieval or reception conditions.

Two stream formats are specified: the transport stream (TS) and the program stream (PS). In the following, we limit the discussion to the TS only, since this is the commonly used format for the applications discussed above. TS is based on a packet-oriented multiplex that makes it suitable for broadcast, as well as storage purposes.

In the basic multiplexing approach for a single video and a single audio elementary stream, a TS is produced. The video and audio components (in MPEG-2: elementary stream) are coded using any of the codecs specified in MPEG (as these have to conform to the byte-stream definitions of MPEG-2 systems), besides others. The elementary stream data are packetized with a header containing information about the carried media, i.e., in the HEVC case a complete access unit or one or more NAL units, to form a packetized elementary stream (PES) packet. PES header contains timing information such as a common clock base and timing to indicate when the bitstream data can be decoded and presented. The PES header is comparable to the function of the RTP and the RTP Payload header as discussed in Section IV.

The PESs packets are fragmented to fixed length transport stream packets of 188 bytes. The TS combines the fragmented PES data, where each PES is associated with a packet identifier (PID) in the TS packet header, of one or more programs with one or more independent time bases into a single multiplexed stream. The PID in the TS packet header is used, beside other purposes, for identifying a PES, for associating PESs such as the video component to a program, as well as for demultiplexing of media components. The transport stream is designed for use in environments where errors are likely, such as transmission over lossy channels. Such errors may manifest as bit value errors or packet loss.

B. MPEG-2 STD

The unique method of synchronization in MPEG-2 systems is based on the system target decoder model (STD). This is a common model used by encoding equipment and decoders for interoperability. Certain TS packets contain clock information or a time-line which is referred to as the program clock reference (PCR). The PCR is used for synchronizing senders and receivers and thus allow for avoiding clock skew, which may cause buffer overflows/underflows. Each media component such as video or audio includes a time value that indicates when a complete picture or audio frame is ready for decoding with respect to the timing ruled by the PCR. This is called the decoding time stamp (DTS) and it is common practice to include this information for each presentable sample of the multimedia component while if not present, it is derived to be equal to the presentation time stamp (PTS). Video codecs use the function of picture reordering between transmission input and decoding output order and hence the packets include both the PTS, which refers to the time value for displaying a picture, and DTS for video components. Audio codecs do not use reordering, as a result the PTS is equal to the DTS and therefore audio packets only include the PTS.

Video and audio coding specifications use the term access unit to define the coded bits of a full picture or audio frame. The STD model is a buffer management model. The STD specifies the buffer sizes to hold coded media data and the time when an access unit can be removed from the buffer and decoded as well as any reordering before display (for video components). Therefore, it is very similar to the HRD model of H.264/AVC [3] as well as HEVC [1] and preserves the constraints given by those HRD models on picture timing, buffer size and buffer handling. The STD controls similar to the HRD the elementary stream buffer, equivalent to the coded picture buffer in the HRD, as well as the decoded picture buffer. For audio, the STD uses the buffer size specified in the respective codec. In addition, MPEG Systems specifies a buffer size for the metadata used to signal multimedia components. More details can be found in [5].

C. HEVC Integration Into MPEG-2 TS

Even though HEVC specifies a unitary access unit structure consisting of all picture-related NAL units, the transport may allow different organization as follows.

1) Transport of NAL units of an access unit within one PES associated with one PID.
2) Transport of NAL units of temporal layers or layers/views (of HEVC extensions) of a single access unit each in a separate PES of the same program.

The principle of distributing data of different HEVC access units is inherited from the MPEG-2 Systems extensions for SVC [32] and MVC [33]. Fig. 5 shows how NAL units associated with different temporal IDs or layer IDs of the same access unit (with the same DTS) are distributed across multiple PESs.

The Amendment of the MPEG-2 Systems standard [5] for HEVC [24] supports the transport of one or more temporal layers in different substreams of an elementary stream. The amendment for HEVC further describes the encapsulation of NAL units into PES packets

The approach of separating substreams to PESs and by that to different PIDs in turn requires extensions to the conventional
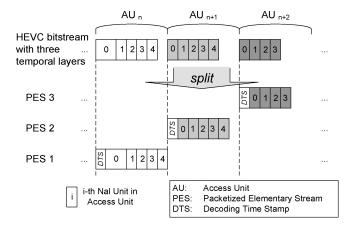
Fig. 5. Encapsulation of HEVC bitstream into multiple PESs with temporal scalability.

single stream STD model described above. The intended STD extensions for HEVC are shown in Fig. 6. In the following, the relevant information that needs to be managed by the encoders and decoders implementing the HEVC transport scheme is described.

An HEVC descriptor describes basic characteristics of the elementary stream, such as the profile and the level to which the elementary stream conforms. Furthermore, the descriptor indicates the temporal layers included within a PES. The contained temporal layers are indicated, when a temporal-subset of an HEVC bitstream is included in a PES on a certain PID and the specific profile and level information of the temporal-subset is further indicated. The descriptor may further include information about used parallelization tools within the bitstream similar as described in Section IV-B3), depending on the ongoing standardization process of the amendment to MPEG-2 TS for HEVC.

Upon arrival in the receiver, the STD demultiplexes the incoming TS packets based on the PID. After some buffering and processing of the TS and PES headers (TS/PES processing in the figure), the elementary stream data are collected in a dedicated part of the elementary stream buffer, also known as the coded picture buffer in HEVC [1]. Based on the decoding timestamp, NAL units of the different ES buffers are collected and delivered to the HEVC decoder. At the point of writing, the amendment for HEVC was not fully specified, therefore the information in this section should be verified against the final standard.

VI. ISO BASE MEDIA FILE FORMAT INTEGRATION

The ISO Base Media File Format (ISOBMFF) [6] is used as the basis for many codec encapsulation formats, such as the AVC File Format [25], as well as for many multimedia container formats, such as the MPEG-4 File Format, the 3GPP File Format (3GP) [27], and the DVB File Format [28], [29]. In addition to continuous media, such as audio and video, static media, such as images, as well as metadata can be stored in a file conforming to ISOBMFF. Files structured according to the ISOBMFF may be used for many purposes, including local media file playback, progressive downloading of a remote

file, segments for Dynamic Adaptive Streaming over HTTP (DASH) [7], [30] (see Section VII), containers for content to be streamed and its packetization instructions, and recording of received real-time media streams.

A box is the elementary syntax element in the ISOBMFF, including a four-character type, the byte count of the box, and the payload. An ISOBMFF file consists of a sequence of boxes, and boxes may contain other boxes. A Movie box ("moov") contains the metadata for the continuous media streams present in the file, each one represented in the file as a track. The metadata for a track is enclosed in a Track box ("trak"), while the media content of a track is either enclosed in a media data box ("mdat") or directly in a separate file. The media content for tracks consists of a sequence of samples, such as audio or video access units. The ISOBMFF specifies the following types of tracks: a media track, which contains an elementary media stream, a hint track, which either includes media transmission instructions or represents a received packet stream, and a timed metadata track, which comprises time-synchronized metadata.

Although originally designed for storage, the ISOBMFF has proven to be very valuable for streaming, e.g., for progressive download or DASH. For streaming purposes, the movie fragments defined in ISOBMFF can be used. In Fig. 7, a fragmented ISOBMFF file is shown with two tracks, e.g., associated with video and audio. After reception of the "moov" box any movie fragment "moof" with its associated media data can be decoded, provided that it contains a random access.

The metadata for each track includes a list of sample description entries, each providing the coding or encapsulation format used in the track and the initialization data needed for processing that format. Each sample is associated with one of the sample description entries of the track.

The ISOBMFF enables specifying sample-specific metadata with various mechanisms. Specific boxes within the Sample Table box ("stbl") have been standardized to respond to common needs. For example, a Sync Sample box ("stss") is used to list the random access samples of the track. The sample grouping mechanism enables mapping of samples according to a four-character grouping type into groups of samples sharing the same property specified as a sample group description entry in the file. Several grouping types have been specified in the ISOBMFF.

The draft HEVC file format [31] has many design aspects similar to the AVC file format [25], as both HEVC and H.264/AVC bitstream consist of a sequence of access units and a sequence of NAL units. Some of the most fundamental design decisions for the HEVC file format are reviewed in the following paragraphs, with a specific focus to differences compared to the AVC file format.

A fundamental property of the parameter set feature of H.264/AVC and HEVC is that parameter sets are allowed to be carried in-band within the video bitstream or out-of-band. However, the current AVC file format allows parameter sets to be present only out-of-band (i.e., not being a part of media samples), either in a separate parameter set track or in sample description entries for an H.264/AVC track within the "moov" box. In the next version of the AVC file format, two new
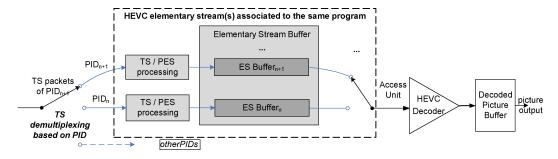
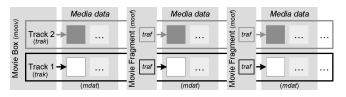Fig. 6. STD receiver operation for HEVC with multi-PES.

Fig. 7. ISOBMFF with movie fragments.

Fig. 8. DASH system architecture.

sample entries ("avc3" and "avc4") will be included, which allow in-band carriage of parameter sets.

Using the current AVC file format not supporting the new sample entries, the file encapsulator had to extract those parameter sets from the bitstream prior to encapsulating the bitstream to the file, and create either parameter set track or sample description entries based on the extracted parameter sets. A player reading an AVC file with a parameter set track has to parse both the AVC video track and its parameter set track synchronously. The parameter set track has been rarely used in H.264/AVC file generator and parser implementations.

In order to simplify the HEVC file format and to allow easier file generation with in-band parameter sets, the HEVC file format includes two modes for storage of parameter sets. In the first mode, which corresponds to the out-of-band mechanism, identified by the use of "hvc1" sample entry type, parameter sets are stored only within sample description entries. The additional out-of-band mechanism, which considers a separate parameter set track, is not included in the HEVC file format. In the second mode, identified by the use of "hev1" sample entry type, parameter sets can be stored in sample description entries and/or in-band within the samples themselves. All parameter sets required for decoding a sample at or subsequent to a random access point are included either in the referred sample description entry or are present in-band at or subsequent to that random access point before they are referred to.

Similarly to the AVC file format, it was considered that any pieces of data that may appear as in-band in the HEVC bitstream and as file format metadata should be avoided and should only appear as file format metadata to avoid conflicts and enable easier file modification. Consequently, all picture timing information is provided as file format metadata, for example as decoding times, composition times, and edit lists, and in-band timing information does not need to be stored in the file and should be ignored by file parsers.

The presence of temporal_id for all access units including the BLA and CRA picture types is utilized in the HEVC file format a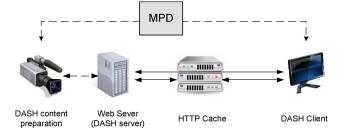s follows. A sample group type for temporal scalabil-ity was defined in the HEVC file format, where each sample group description entry corresponds to one temporal_id value, thus a file format level mapping of samples to temporal levels can be done with this sample grouping. In addition to the mapping, the sample group description entries provide information on the profile and level to which the temporal subset of the bitstream conforms to, the maximum and average bitrate, an indication if the temporal level represents a constant picture rate, and the average or constant frame rate of the temporal subset of the bitstream. The temporal scalability sample grouping provides an easy-to-use index of access units for temporal subsets of an HEVC bitstream, hence enabling for example fast-forward play operation to be implemented through decoding a bitstream subset. As the HEVC standard specifies a normative decoding process for bitstreams starting with a IDR, BLA, or CRA picture, they qualify as random access points for all standard-compliant decoders. Thus, it was decided to include IDR, BLA, and CRA samples in the Sync Sample box of the HEVC File Format.

VII. DASH INTEGRATION

Dynamic Adaptive Streaming over HTTP (DASH) [7] is a standard for HTTP (adaptive) streaming applications. It mainly specifies the format of Media Presentation Description (MPD), also known as manifest, and the media segment format. See for an overview of DASH in [30]. Fig. 8 shows the DASH system architecture including content preparation, the HTTP server, HTTP caching infrastructure and the DASH client, where the MPD describes the media available on the server. This lets the DASH client to autonomously download the media version at the media time it is interested in.

A typical procedure for DASH based HTTP streaming includes the following steps.

1) A client obtains the MPD of a streaming content, e.g., a movie. The MPD includes information on different alternative representations, e.g., bit rate, video resolution, frame rate, audio language, of the streaming content, as well as the URLs of the HTTP resources (the initialization segment and the media segments).

2) Based on information in the MPD and the client's local information, e.g., network bandwidth, decoding/display capabilities and user preference, the client requests the desired representation(s), one segment (or a part thereof) at a time.

3) When the client detects a network bandwidth change, it requests segments of a different representation with a better-matching bitrate, ideally starting from a segment that starts with a random access point.

During an HTTP streaming session, to respond to the user request to seek backward to a past position or forward to a future position, the client requests past or future segments starting from a segment that is close to the desired position and that ideally starts with a random access point. The user may also request to fast-forward the content, which may be realized by requesting data sufficiently for decoding only the intra coded video pictures or only a temporal subset of the video stream.

The latest ISOBMFF specification specifies six types of Stream Access Points (SAPs) for use with DASH. The first two SAP types (types 1 and 2), correspond to IDR pictures in H.264/AVC and HEVC. The third SAP type (type 3) corresponds to open-GOP random access points hence BLA or CRA pictures in HEVC.

HEVC's BLA/CRA support, both signaling through the BLA/CRA picture NAL unit type and conforming bitstreams being possible to start with BLA or CRA pictures, makes HEVC more friendly than H.264/AVC to all DASH operations based on intra coding or random access points, e.g., seeking, stream adaptation, and intrapictures-based fast-forward trick mode. As mentioned earlier, random accessibility provisioning based on BLA/CRA pictures also provides better compression efficiency for the video bitstream.

The temporal scalability signaling of HEVC also provides a superiority of HEVC over H.264/AVC for use with DASH, as both temporal subsetting-based stream adaptation and fast-forward trick mode (e.g., through the subrepresentation feature of DASH) become more convenient.

DASH support two types of media segment formats, one based on ISOBMFF, and the other based on MPEG-2 TS. For DASH contents with H.264/AVC video based on ISOBMFF, the AVC file format is used for encapsulation of H.264/AVC video streams. For future DASH contents with HEVC video based on ISOBMFF, the HEVC file format will be used. One shortcoming of the AVC file format design relates to DASH-based live streaming of ISOBMFF encapsulated H.264/AVC video. Due to lacking of parameter set track implementations of the AVC file format, storing of parameter sets in sample entries becomes the only option in many cases. However, since sample entries must be included in the movie box ("moov") per ISOBMFF, all parameter sets must be composed at the beginning of the streaming "session" and no new parameter

sets may be generated afterwards. This would unavoidably result in a sub-optimal coding of the video sequences during live streaming. With the flexible design for storage of parameter sets in the HEVC file format and the additional sample entries for AVC file format as discussed in Section VI, new parameter sets may be generated whenever needed and stored in-band with the video samples, and the above problem faced with H.264/AVC is solved. DASH will also allow for carrying HEVC packetized following the MPEG-2 TS format as presented in Section V.

VIII. Conclusion

This paper gives a first insight into the system layer integration of the HEVC standard into system layers RTP, MEPG-2 TS, ISO File Format, and HTTP Streaming using DASH. Beside the HEVC integration aspects into those system standards, this paper gives an overview of today's system layers for video transport and delivery over Broadcast, Internet and using storage devices. HEVC supports by default temporal scalability, new parallelization techniques and further new High Level Syntax elements. The use of those features for system layer integration is described in detail, while giving for each discussed system layer a basic introduction.

References

[1] B. Bross, W. J. Han, J. Ohm, G. Sullivan, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 6*, document JCTVC-J1003, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 10th Meeting, Stockholm, Sweden, Jul. 2012.

[2] R. Sjöberg, Y. Chen, A. Fujibayashi, M. M. Hannuksela, J. Samuelsson, T. K. Tan, Y.-K. Wang, and S. Wenger, "Overview of HEVC high-level syntax and reference picture management," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1857–1869, Dec. 2012.

[3] ITU-T and ISO/IEC JTC 1, *Advanced Video Coding for Generic Audio-visual Services*, document ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2010.

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, Eds. (2003, Jul.). *RTP: A Transport Protocol for Real-Time Applications, IETF STD 0064, RFC 3550* [Online]. Available: http://tools.ietf.org/html/rfc3550

[5] ITU-T Rec H.222.0 (05/2006), *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Systems*, document ISO/IEC 13818-1:2007 information technology—generic coding of moving pictures and associated audio information (MPEG-2)—Part 1: Systems, 2006.

[6] ISO/IEC JTC1/SC29/WG11, "Information technology—Coding of Audio-Visual Objects—Part 12: ISO base media file format," ISO/IEC 14496-12:2008 (3rd ed.), 2008.

[7] ISO/IEC JTC1/SC29/WG11, "Information technology—Dynamic Adaptive Streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats," ISO/IEC 23009-1:2012, 2012.

[8] K. Park and G. Fernando, *Working Draft of MPEG Media Transport*, document ISO/IEC JTC1/SC29/WG11/N12531, Feb. 2012, San Jose, CA.

[9] M. Luby, T. Stockhammer, and M. Watson, "IPTV Systems, Standards and Architectures: Part II—Application Layer FEC In IPTV Services," *IEEE Commun. Magazine*, vol. 46, no. 5, pp. 94–101, May 2008.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee (Eds.), (Jun. 1999). *Hypertext Transfer Protocol—HTTP/1.1,* document IETF RFC 2616 [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt

[11] U. Reimers, *DVB (Digital Video Broadcasting)*, 2nd ed., Berlin, Germany, Springer Verlag, Sep. 2004.

[12] T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh (Eds.). (Oct. 2004). *FLUTE—File Delivery Over Unidirectional Transport,* document IETF RFC3926 [Online]. Available: http://tools.ietf.org/html/rfc3926

[13] A. Fujibayashi and T. K. Tan, *Random Access Support for HEVC*, document JCTVC-D234, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th Meeting: Daegu, Korea, 20–28 Jan. 2011.

[14] M. M. Hannuksela, *Enhanced Concept of GOP*, Joint Video Team document JVT-B042, Jan. 2002 [Online]. Available: http://wftp3.itu.int/av-arch/jvt-site/2002_01_Geneva/

[15] D. Tian, M. M. Hannuksela, and M. Gabbouj, "Sub-sequence video coding for improved temporal scalability," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 6074–6077.

[16] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2006, pp. 1929–1932.

[17] Q. Shen, Y.-K. Wang, M. M. Hannuksela, H. Li, and Y. Wang, "Buffer requirement analysis and reference picture marking for temporal Scalable Video Coding," in *Proc. Int. Packet Video Workshop*, Nov. 2007, pp. 91–97.

[18] Y.-K. Wang, M. M. Hannuksela, K. Caglar, and M. Gabbouj, "Improved error concealment using scene information," in *Proc. Int. Workshop VLBV03*, LNCS. Sep. 2003, pp. 283–289.

[19] Y. Ye and E.-S. Ryu, *On Adaptation Parameter Signalling*, document JCT-VC, H0132, Feb. 2012.

[20] C. Perkins, *RTP: Audio and Vudei for the Internet*. Reading, MA: Addison-Wesley, 2003

[21] T. Schierl, S. Wenger, Y.-K. Wang, and M. M. Hannuksela (Eds.). (Feb. 2012). "RTP payload format for High Efficiency Video Coding," Internet Engineering Task Force (IETF), Audio Video Transport Group (AVT) [Online]. Available: http://tools.ietf.org/html/draft-schierl-payload-rtp-h265

[22] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer (Eds.), (Feb. 2005.). "RTP payload format for H.264 Video," IETF RFC 3984, Internet Engineering Task Force (IETF), Audio Video Transport (AVT) group [Online]. Available: http://tools.ietf.org/html/rfc3984

[23] Y.-K. Wang, R. Even, T. Kristensen, and R. Jesup (Eds.). (May 2011). "RTP payload format for H.264 Video," IETF RFC 6184, Internet Engineering Task Force (IETF), Audio Video Transport (avt) group [Online]. Available: http://tools.ietf.org/html/rfc6184

[24] T. Schierl, K. Gruenberg, and Sam Narasimhan, *Working Draft 1.0 for Transport of HEVC Over MPEG-2 Systems*, document ISO/IEC SC29/WG11, MPEG99/N12466, Feb. 2012.

[25] ISO/IEC JTC1/SC29/WG11, *Information Technology—Coding of Audio-Visual Objects—Part 15: Advanced Video Coding (AVC) File Format*, document ISO/IEC 14496-15:2010(E), Jan. 2010.

[26] ISO/IEC JTC1/SC29/WG11, *Information Technology—Coding of Audio-Visual Objects—Part 14: MP4 file Format*, document ISO/IEC 14496-14:2003, 2003.

[27] 3rd Generation Partnership Project; "Technical Specification Group Services and System Aspects; Transparent End-to-End Packet Switched Streaming Service (PSS); 3GPP file format (3GP)," ETSI TS 26.224, Rel. 10, Nov. 2011.

[28] ETSI TS 102 833 v1.2.1, "File format specification for the storage and playback of DVB services," Mar. 2011.

[29] M. M. Hannuksela, S. Döhla, and K. Murray, "The DVB file format," *IEEE Signal Process. Mag.*, vol. 29, no. 2, pp. 148–153, Mar. 2012.

[30] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[31] D. Singer, *Study of ISO/IEC 14496-15:2010/PDAM 2, AMENDMENT 2: Carriage of HEVC*, document ISO/IEC JYT1/SC29 WG11, Jul. 2012.

[32] T. Schierl, B. Berthelot, and T. Wiegand (Eds.), *ISO/IEC 13818-1:2007/AMD3—Transport of SVC Video Over ITU-T Rec H.222.0*, |document ISO/IEC 13818-1, Oct. 2009.

[33] T. Schierl, K. Grüneberg, S. Narasimhan, and A. Vetro, Eds., *ISO/IEC 13818-1:2007/AMD4—Transport of Multiview Video Over ITU-T Rec H.222.0 |Document ISO/IEC 13818-1*, ISO/IEC JTC1/SC29/WG11, London, U.K., Sep. 2009.

[34] 3GPP, ETSI TS 26.346, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (Release 11), V11.1.0, Jun. 2012.

[35] C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel scalability and efficiency of HEVC parallelization approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1826–1837, Dec. 2012.

[36] P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, "Subjective quality evaluation of the upcoming HEVC video compression standard," in *Proc. SPIE Optics Photon.*, vol. 8499. Aug. 2012.

Thomas Schierl (S'05–M'07) received the Diplom-Ingenieur degree in computer engineering and the Doktor der Ingenieurwissenschaften (Dr.-Ing.) degree in electrical engineering and computer science from the Berlin University of Technology, Berlin, Germany, in 2003 and 2010, respectively.

He has been with the Fraunhofer Institute for Telecommunications (HHI), Berlin, Germany, since 2004. Since 2010, he has been the Head of the Multimedia Communications Group, Image Processing Department, Fraunhofer HHI. He is a co-author of various IETF RFCs, the IETF RTP Payload Format for H.264 Scalable Video Coding (SVC), and for HEVC. In 2007, he visited the Image, Video, and Multimedia Systems Group of Prof. B. Girod at Stanford University, Stanford, CA, for different research activities. His current research interests include mobile media streaming and content delivery.

Dr. Schierl is a co-editor of the MPEG Standard on Transport of H.264 SVC, H.264 MVC, and HEVC over MPEG-2 transport stream in the ISO/IEC MPEG group. He is also a co-editor of the AVC File Format.

Miska M. Hannuksela (M'03) received the Master of Science degree in engineering and the Doctor of Science degree in technology from the Tampere University of Technology, Tampere, Finland, in 1997 and 2010, respectively.

He has been with Nokia since 1996 in different roles, including a Research Manager/Leader in the areas of video and image compression, end-to-end multimedia systems, as well as sensor signal processing and context extraction. He is currently a Distinguished Scientist of Multimedia Technologies with the Nokia Research Center, Tampere. He has published around 100 journal and conference papers and hundreds of standardization contributions in JCT-VC, JVT, MPEG, 3GPP, and DVB. He holds patents from more than 60 patent families. His current research interests include video compression, multimedia communication systems and formats, user experience and human perception of multimedia, and sensor signal processing.

Dr. Hannuksela received the Award of the Best Doctoral Thesis of the Tampere University of Technology in 2009 and the Scientific Achievement Award nominated by the Centre of Excellence of Signal Processing, Tampere University of Technology, in 2010. He has been an Associate Editor in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS OF VIDEO TECHNOLOGY since 2010.

Ye-Kui Wang received the B.S. degree in industrial automation from the Beijing Institute of Technology, Beijing, China, in 1995, and the Ph.D. degree in electrical engineering from the Graduate School in Beijing, University of Science and Technology of China, Beijing, in 2001.

He is currently a Senior Staff Engineer with Qualcomm, San Diego, CA. He was a Multimedia Standards Manager with Huawei Technologies, Bridgewater, NJ, from December 2008 to August 2011, a Principal Research Staff Member with Nokia Corporation, Tampere, Finland, from February 2003 to December 2008, and a Senior Researcher with the Tampere International Center for Signal Processing, Tampere University of Technology, Tampere, from June 2001 to January 2003. His current research interests include video coding, multimedia transport, and systems. He has co-authored more than 300 technical standardization contributions, about 50 academic papers, and more than 100 granted or pending patents or patent applications in his areas of interest.

Dr. Wang has been an active contributor to various multimedia standards, including video codecs, file formats, RTP payload formats, and streaming systems, developed in ITU-T VCEG, ISO/IEC MPEG, JVT, JCT-VC, 3GPP SA4, IETF, and AVS. He has been an editor for several standard specifications, including ITU-T H.271, SVC file format, multiview video coding, IETF RFC 6184, and IETF RFC 6190.

Stephan Wenger (M'98) received the Diploma and Ph.D. degrees in computer science from Technische Universitat (TU) Berlin, Berlin, Germany, in 1989 and 1995, respectively.

After working as an Assistant Professor with TU Berlin, he was engaged in various consulting roles, and was with Nokia's Research Center for five years and IPR/legal groups. Since 2007, he has been a Chief Technology Officer of VidyoCast, a division of Vidyo, Inc., Hackensack, NJ. He has published dozens of journal and conference papers, Internet RFCs, hundreds of standardization contributions to many standards setting organizations, and more than 30 granted or pending patents and patent applications. He is interested in the interface between media coding and transport over a variety of networks.