Video Transport Evaluation With H.264 Video Traces

Patrick Seeling, *Senior Member, IEEE,* and Martin Reisslein, *Senior Member, IEEE*

***Abstract*—The performance evaluation of video transport mechanisms becomes increasingly important as encoded video accounts for growing portions of the network traffic. Compared to the widely studied MPEG-4 encoded video, the recently adopted H.264 video coding standards include novel mechanisms, such as hierarchical B frame prediction structures and highly efficient quality scalable coding, that have important implications for network transport. This tutorial introduces a trace-based evaluation methodology for the network transport of H.264 encoded video. We first give an overview of H.264 video coding, and then present the trace structures for capturing the characteristics of H.264 encoded video. We give an overview of the typical video traffic and quality characteristics of H.264 encoded video. Finally, we explain how to account for the H.264 specific coding mechanisms, such as hierarchical B frames, in networking studies.**

***Index Terms*—H.264 encoded video, hierarchical B frames, medium grain scalability (MGS), network transport, simulation, traffic variability, video trace.**

I. Introduction

A. Motivation

THIS TUTORIAL on evaluating the performance of video network transport is motivated by two main developments, namely the increasing video network traffic and the emergence of the highly efficient H.264 video coding standard. Recent network traffic analyses and predictions indicate that video accounts for a growing proportion of the network traffic. For instance, Cisco Inc. predicts that video transmitted to and from mobile devices will account for 66 % of the global mobile data traffic by 2014 [1].

At the same time, the H.264/MPEG-4 Part 10 Advanced Video Coding standard [2] (which we refer to as H.264/AVC for brevity) and its Scalable Video Coding (SVC) extension [3] achieve significantly improved compression efficiency compared to the preceding MPEG-4 Part 2 video coding standard (which we refer to as MPEG-4) and the MPEG-2 video coding standard. For brevity we use the term “H.264” to refer to the standards family consisting of the H.264/AVC standard and its SVC extension. The compression efficiency of a video codec is generally characterized with the rate-distortion (RD) curve that shows the video quality (distortion), which is typically measured in terms of the Peak Signal to Noise Ratio (PSNR), as a function of the bit rate of the compressed video stream. For a given video quality, the lower the compressed bit rate, the more RD efficient is the compression. In typical scenarios, H.264/AVC and SVC are roughly twice as RD efficient as MPEG-4, i.e., for a given PSNR video quality, the average bit rates of H.264/AVC and SVC single-layer (non-scalable) streams are roughly half of the corresponding MPEG-4 streams. This improvement in RD compression efficiency is achieved through several novel encoding principles that we review in Section II. One main new concept is hierarchical bi-directional (B) frame prediction, whereby B frames form a hierarchical structure of B frames predicting other B frames. This novel concept has important implications for the timing of the network transport, which we explain in Section V.

Similarly, the RD efficiency of scalable video coding, which provides different streaming rates and qualities with a single video encoding, is significantly improved by H.264 SVC compared to MPEG-4. Whereas MPEG-4 scalable video coding had bit rate overheads on the order of tens of percent compared to single-layer encoding, novel medium grain scalable (MGS) coding mechanisms in H.264 SVC reduce the bit rate overhead to close to zero. This highly efficient MGS video coding mechanism is therefore highly promising for adaptive video streaming in heterogeneous or time-varying networking scenarios. Overall, due to the vast improvements in compression efficiency and the widespread adoption of H.264/AVC and SVC in multimedia application standards and industry consortia specifications, such as Digital Video Broadcasting (DVB), it is highly important to consider H.264 encoded video in networking studies.

B. Objective

The overall objective of this tutorial is to enable communications and networking generalists without prior knowledge in H.264 video coding (i) to design transport mechanisms for H.264 encoded video as well as (ii) to evaluate the performance of H.264 video transport mechanisms with H.264 video traces. More specifically, we give an overview of the novel H.264 video coding mechanisms that achieve the substantial improvements in RD efficiency compared to prior video coding standards, such as MPEG-4 video coding. We explain the implications of these novel encoding mechanisms for network transport. Throughout, we only briefly summarize the characteristics of prior video coding standards where such a summary is necessary to keep this tutorial self-contained. For more detailed background on prior video coding standards and

Manuscript received 2 May 2011; revised 15 August 2011. Supported in part by the National Science Foundation under Grant CRI-0750927.

P. Seeling is with the Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: pseeling@ieee.org).

M. Reisslein is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287-5706, USA (e-mail: reisslein@asu.edu).

Digital Object Identifier 10.1109/SURV.2011.082911.00067

their implications for network transport design and evaluation, we refer to related work, as detailed in Section I-D.

This tutorial article, jointly with the publicly available video trace library (`http://trace.eas.asu.edu`), provides a comprehensive trace-based evaluation methodology for network video transport. Video traces do not contain the actual encoded video (bit) stream; instead, they provide a meta-characterization of the encoded video stream. A video trace provides this meta-characterization by providing the quantities that are required for simulating the transport of the actual video with a communications or networking mechanism. Basic video traces provide the time stamp, encoded size (in byte), and PSNR quality of each encoded video frame. The characterization of the different types of H.264 video encoding requires more elaborate trace designs, as detailed in Section III.

C. Organization

This tutorial is structured as follows. In Section II, we give a brief overview of H.264 video coding. We first introduce single-layer (non-scalable) coding, covering both H.264/AVC video coding as well as H.264 SVC video coding into a single layer. Then, we introduce the H.264 SVC layer-scalable coding, including video coding with temporal (frame frequency) scalability, spatial (frame pixel resolution) scalability, and quality [signal to noise ratio (SNR)] scalability into multiple coding layers. Next, we introduce sublayer quality scalability, which is provided by H.264 medium grain scalability (MGS). H.264 MGS partitions a given layer of a quality scalable coding into several MGS layers (sublayers). Each of the subsequent main sections of this tutorial is organized into subsections according to the types of H.264 video coding into (A) single-layer coding, (B) layer-scalable coding, and (C) sublayer quality scalable coding.

In Section III, we introduce video trace structures for characterizing the different H.264 video encodings. In Section IV, we present video traffic and quality statistics for H.264 encoded video. Throughout Section IV, we use the two animation video sequences *Big Buck Bunny* and *Elephants Dream* to showcase how the different H.264 video encoding types affect video traffic and quality statistics. We also summarize the results from extensive video traffic and quality studies for a large set of long video sequences from a wide range of video genres. In Section V, we explain how to account for the characteristics of H.264 encoded video in the development and evaluation of video network transport mechanisms. We also explain how to use the H.264 video traces for realistic trace-driven simulations of H.264 video transport. In Section VI we summarize this tutorial and briefly outline emerging directions for research on video communications and networking.

D. Related Work

This tutorial article is related to preceding articles on video traffic characterization and evaluation of video transport. Video traces for the earlier video coding standards MPEG-1 through MPEG-4 and H.261 through H.263 are examined in [4]–[9].

The traffic and quality characteristics of the different types of H.264 encoded video have been examined in several studies. For instance, H.264 single-layer video has been studied in [10]–[14], while H.264 layer-scalable video has been examined in [15]–[19]. Sublayer-scalable video has been studied in [3], [19]–[21].

Each of these preceding studies focuses on a particular type of H.264 video encoding and its resulting traffic and quality characteristics. In contrast, Sections II and IV of this tutorial article provide a comprehensive treatment of the encoding mechanisms as well as video traffic and quality characteristics for all main existing types of H.264 video encoding. Thus, Sections II and IV provide the communications and networking generalist without background in H.264 video coding with a complete introduction to H.264 video coding and the statistical characteristics of the network traffic from H.264 encoded video. Sections III and V provide for the first time a description of the video trace structures for H.264 encoded video and guidelines for the appropriate use of these video traces in performance evaluation studies.

Throughout this tutorial, we focus on video traces created through the encoding of uncompressed video source materials with prescribed encoding parameters. An alternative approach is to characterize the traffic of encoded video in the public domain, such as encoded videos available from YouTube, in video traces [22], [23]. The main drawbacks of video traces based on encoded public domain videos are that the exact encoding parameters are unknown and that video quality characterization with the widely used PSNR is impossible as the PSNR evaluation requires the uncompressed video source.

The main alternatives to evaluating video network transport with video traces are evaluations with actual video or evaluations with video traffic models. Evaluations with actual video start with the uncompressed source video, carry out the encoding of the source video, simulate the transmission of the actual encoded video bit stream through the transport network, and evaluate the quality of the received video through comparison with the source video [24], [25]. Such evaluations have the advantage that they allow for the detailed analysis of the received video bit stream. However, these evaluations are very compute-intensive as they involve the actual video encoding and decoding; thus, limiting evaluations with long video sequences required for statistically rigorous results. Furthermore, evaluations with actual video require video signal processing expertise to operate the video codecs.

Video traffic models are parsimonious mathematical descriptions of video traffic. Video traffic models are generally derived from the statistical analysis of video traces and have been widely studied for MPEG-4 and earlier video coding standards, see for instance [26]–[30]. Traffic modeling for H.264 encoded video has only been considered in few studies so far, see e.g., [31]–[36], and is presently an active area of research. The trace structures presented in this tutorial and the accompanying video trace library (`http://trace.eas.asu.edu`) can serve as a basis for H.264 video traffic model development and verification.

A wide range of video transport mechanisms have been investigated through simulations driven by video traces of video encoded with MPEG-4 and earlier video coding standards, see for instance [37]–[46]. With the increasing importance of H.264 video coding, networking studies have begun to use the recently created traces of H.264 encoded video, see for instance [35], [47]–[51]. This tutorial seeks to support the use of H.264 video traces in networking studies by providing comprehensive readily accessible instructions for performance evaluation with H.264 video traces.

II. Overview of H.264 Video Coding

The H.264 family of video coding standards enhances the encoding loop consisting of a combination of intra-frame coding with block transform and inter-frame coding with motion compensated prediction from preceding MPEG video coding standards. Each video frame (picture) is either intra-coded (I), forward predictive coded (P) with motion compensated prediction from the preceding I or P frame, or bi-directionally predictive coded (B) according to its position in a group of pictures (GoP) structure. With *classical B frame prediction*, a B frame is encoded with motion compensated prediction from the preceding as well as the succeeding I or P frame, as illustrated in Fig. 1(a). With classical B frame prediction, which has been used in preceding MPEG standards and which is the default in H.264/AVC, a B frame is not used as prediction reference for another B frame.

We use GgBb to denote the GoP structure, whereby g denotes the total number of frames in a GoP and b denotes the number of B frames between successive I or P frames. For instance, G16B3 denotes the GoP structure IBBBPBBBPBBBPBBBIBBBP... and the GoP structure G16B15 is illustrated in Fig. 1. In the following we give an overview of the enhanced coding mechanisms of H.264 that build on the basic MPEG video coding loop and as a cumulative effect give the significantly improved RD coding efficiency of H.264 over earlier video coding standards.

A. Single-layer (Non-scalable) H.264 Video Coding

1) H.264/AVC: The H.264/AVC standard [2], [52], [53] introduced variable block sizes, such as 4×4, 8×4, 16×8, and 8×8 pixels for improved block-based predictive encoding. Intra-frame coding is improved through spatial intra-frame prediction, whereby blocks of a given frame are predicted from similar blocks of the same frame. Inter-frame prediction is improved through unequal weighing of multiple reference frames. Whereas with preceding MPEG video coding standards a block in a B frame was predicted through equal weighing of a block in one preceding I or P frame and a block in one succeeding I or P frame, H.264/AVC introduced multiple reference frames. That is, a block from several preceding I or P frames and another block from several succeeding I or P frames are selected and weighed unequally for predicting the B frame block.

The encoding loop is further enhanced through an in-loop deblocking filter that reduces the artifacts introduced through the block-based encoding. The MPEG video coding loop performs a block transform, followed by the quantization of the transform coefficients, and subsequent zig-zag scan and run-length coding. The resulting symbols are further compressed through entropy coding. In H.264/AVC, the entropy coding is improved through context-adaptive binary arithmetic coding (CABAC) [54], which is highly efficient but compute-intensive, or context-adaptive variable length coding (CAVLC), which is an improvement of preceding variable-length coding mechanisms. Individual coding decisions as well as the entire coding loop are improved through Lagrangian-based RD optimization that seeks to jointly minimize the bit rate while minimizing the introduced visual distortion. For instance, Lagrangian RD optimization is employed to find the best motion vectors for motion compensated prediction of a block from reference blocks in preceding and succeeding frames.

The H.264/AVC standard classifies all the video coding mechanisms discussed so far into the so-called video coding layer (VCL). In addition, the H.264/AVC standard defines a network abstraction layer (NAL) which contains functions for mapping the coded video data to a network transport layer. The coded video data is organized into NAL units (NALUs). Each NALU contains an integer number of bytes of video encoded data, as well as a one-byte header (four byte header for enhancement layers) [55]. The NALUs containing a frame are typically preceded by a prefix NALU describing the frame's NALUs.

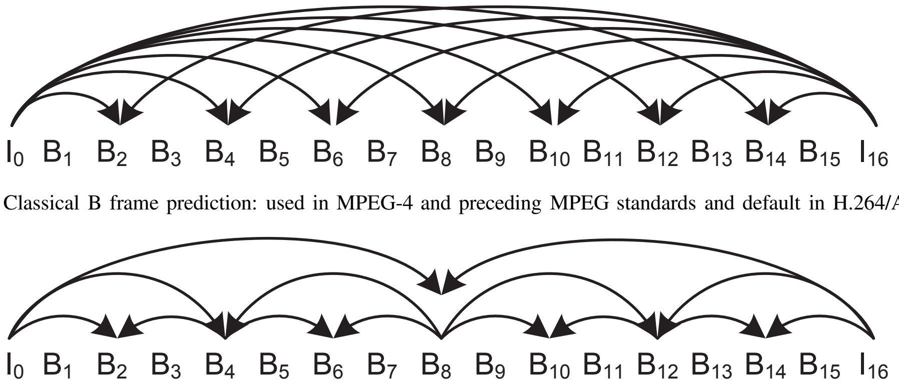
The H.264/AVC standard defines several encoding profiles, including the *main profile*, which includes all tools for achieving high RD efficiency, and several *high profiles* for efficiently encoding high definition (HD) video in the fidelity range extension (FRExt) amendment [13].

2) Single-layer H.264 SVC Coding: Although H.264 SVC has primarily been introduced to add scalable encoding features to H.264/AVC, H.264 SVC has important enhancements that improve single-layer encoding. We give an overview of these enhancements in this subsection and defer the scalability features to Sec. II-B.

The video coding standards preceding H.264/AVC followed strictly the classical B frame prediction illustrated in Fig. 1(a), i.e., B frames were not used to predict other B frames. H.264/AVC lifted this restriction through its generalized B frame concept that permitted B frame blocks to be used as reference for the motion compensated prediction of blocks of other B frames. In subsequent studies [3], this referencing of B frames emerged as the most promising avenue for developing the scalable video coding extension.

H.264 SVC employs the hierarchical B frame prediction illustrated in Fig. 1(b). In the illustrated typically employed dyadic hierarchy of B frames there are $\beta = 2^\tau - 1$ B frames between successive I or P frames, (whereby τ is a positive integer). We observe from Fig. 1(b) that frame B_4 is predicted from frame I_0 and frame B_8. Then, continuing in the hierarchical structure, frame B_6 is predicted from frames B_4 and B_8. In turn, frame B_5 is predicted from frames B_4 and B_6 (not illustrated by arrows to avoid clutter).

According to the prediction hierarchy, B frames are assigned to $\tau = \log_2(\beta + 1)$ temporal enhancement layers that are exploited for temporal scalability. In the example in Fig. 1(b) with $\beta = 15$ B frames between successive I frames, the I frames belong to the temporal base layer $T = 0$, frame B_8 belongs to the first temporal enhancement layer $T = 1$, frames B_4 and B_{12} belong to the second temporal enhancement layer $T = 2$, frames B_2, B_6, B_{10}, and B_{14} belong to the third temporal enhancement layer $T = 3$, and the remaining B frames belong to the fourth (highest) temporal enhancement


I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16}
I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16}

(a) Classical B frame prediction: used in MPEG-4 and preceding MPEG standards and default in H.264/AVC

(b) Hierarchical B frame prediction: default in H.264 SVC

Fig. 1. Illustration of classical and hierarchical B frame prediction for GoP structure G15B15 with a total of 16 frames and 15 B frames. (Prediction references for odd-indexed frames are omitted to avoid clutter.)

layer $T = \tau = 4$. Thus, there are a total of $\tau + 1$ temporal layers.

The RD efficiency of the hierarchical B frame prediction critically depends on the quantization parameter (scale) q used for quantizing the coefficients resulting from the block transform of the difference between the weighted reference blocks and the block to be encoded. The basic insight is that a B frame should be encoded with higher fidelity when more subsequent predictions depend on the B frame. Based on this insight, H.264 SVC introduced cascading quantizers for hierarchical B frame prediction. With cascading quantizers, the encoder increases the quantizer values from a basic B quantization parameter setting (resulting in a coarser, lower fidelity quantization) for the B frames in higher indexed temporal layers. (In addition, some encoding settings add a small integer to the quantization parameter of the I and P frames in the temporal base layer, which we denote by q, to obtain the basic B frame quantization parameter setting.)

Notice from Fig. 1 that classical and hierarchical B frame prediction fundamentally differ in the order in which the frames are encoded. Generally, before a given frame n can be encoded, all frames that are referenced by frame n need to be encoded so that they are available in encoded form as encoding references for frame n. This implies for classical B frame prediction that the frames can be encoded in the order $\mathrm{I}_0,\ \mathrm{I}_{16},\ \mathrm{B}_1,\ \mathrm{B}_2,\ldots$. This encoding order is typically preferred as it provides the smallest decoding delay at the receiver. (Specialized constrained and low-delay B frame prediction structures [3] are beyond the scope of this tutorial.) However, this encoding order limits the generalized B frame concept to reference only preceding B frames for the encoding of a given B frame. In contrast, with hierarchical B frame prediction, the frames are encoded in the order $\mathrm{I}_0,\ \mathrm{I}_{16},\ \mathrm{B}_8,\ \mathrm{B}_4,\ \mathrm{B}_2,\ \mathrm{B}_1,\ \mathrm{B}_3,\ldots$. We also note that these differences in frame encoding order imply different delays for network transport that are examined in Section V.

B. Layer-scalable H.264 Video Coding

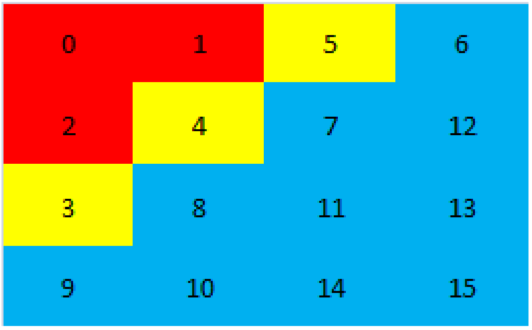
H.264 SVC supports layer-scalable coding providing temporal scalability, spatial scalability, and quality (SNR) scalability [3]. A layer-scalable encoding consists of a base layer and one or several enhancement layers identified by increasing layer identifiers. While H.264 SVC supports up to 128 layers, the actual number of layers in an encoding depends on the application scenario. With the currently specified profiles, there is a maximum of 47 enhancement layers [56].

1) Temporal Scalability: H.264 SVC provides temporal scalability, i.e., adaptation of the frame frequency, by exploiting the hierarchical B frame structure. Suppose that the full frame frequency, achieved by displaying all frames, is 30 frames/s. Consider the example in Fig. 1. Dropping temporal enhancement layer $T = 4$ consisting of frames $\mathrm{B}_1,\ \mathrm{B}_3,\ \mathrm{B}_5,\ldots$, halves the frame frequency to 15 frames/s. Dropping each successive lower-indexed layer halves the frame frequency. The base layer, consisting only of the I frames, provides a frame frequency of (30/16) frames/s.

2) Spatial Scalability: Spatial scalability provides different spatial frame resolutions, e.g., a common interframe format (CIF) 352×288 pixel base layer and a 4CIF 704×576 pixel full resolution obtained by decoding both base layer and enhancement layer. Each spatial layer employs motion compensated prediction and intra-prediction. In addition, H.264 SVC increases the RD coding efficiency through inter-layer prediction mechanisms, such as prediction of macroblock modes and associated motion parameters and prediction of the residue signal [3]. These inter-layer prediction mechanisms generally predict higher spatial enhancement layers from the base layer and lower spatial enhancement layers.

3) Quality Scalability: H.264 SVC Coarse Grain Scalability (CGS): H.264 SVC coarse grain scalability (CGS) provides up to eight quality layers that successively increase the fidelity (SNR) of the video frames [57]. H.264 SVC CGS employs the same inter-layer prediction mechanisms as H.264 spatial scalable coding, except that the operations relating to the scaling of the spatial frame resolution, such as up-sampling operations and the inter-layer de-blocking for intra-coded reference layer blocks, are not performed since all quality layers have the same spatial resolution. The successive increase in fidelity is achieved through re-quantization of the residual texture signal with successively smaller quantization step sizes [3].

As a result of the outlined inter-layer prediction mechanisms, a given CGS enhancement layer depends on the lower



0	1	5	6
2	4	7	12
3	8	11	13
9	10	14	15

Fig. 2. Illustration of division of transform coefficients of a 4×4 block into $M = 3$ MGS layers with weight vector **W** = [3,3,10]. Coefficients with indices 0–2 form MGS layer 1, while coefficients with indices 3–5 form MGS layer 2, and coefficients with indices 6–15 form MGS layer 3.

CGS enhancement layers and the base layer for decoding. In addition, each layer of a CGS encoding employs the hierarchical B frame prediction structure. An encoder with a basic configuration can switch to decoding fewer or more CGS enhancement layers at any intracoded (I) frame. With some decoder enhancements, switching to fewer CGS enhancement layers is possible at any frame [3, Section VI.C.].

4) Combined Spatiotemporal-SNR Scalability: The H.264 SVC standard supports combined scalability, i.e., the extraction of video streams with different combinations of frame frequencies, spatial resolutions, and SNR qualities (achieved through CGS or MGS) from one encoded bit stream.

C. Sublayer Quality Scalability: H.264 SVC Medium Grain Scalability (MGS)

A drawback of H.264 CGS is relatively poor RD performance when the differences in quality (and bit rate) between successive quality layers are small, as detailed in Section IV. Bit rate increments as small as one byte per video frame can be provided with fine grain scalability (FGS) [58], [59], which was part of MPEG-4 and initial investigations for H.264 SVC. The research leading to H.264 SVC found that bit rate adaptation at the level of individual bytes per frame resulted in high computational complexity. For H.264 SVC, a novel quality scalability approach lying between coarse grain scalability at the level of complete layers and fine grain scalability at the level of individual bytes was developed, namely medium grain scalability (MGS). MGS splits a given quality enhancement layer into up to 16 MGS layers and achieves RD efficiency very close to single-layer H.264 SVC RD efficiency (i.e., has a very small bit rate overhead) [19], [56].

MGS divides the transform coefficients of a block into MGS layers. A 4×4 pixel block results in 16 transform coefficients, as illustrated in Fig. 2. In conventional video coding, all coefficients are jointly zig-zag scanned, followed by run-length and entropy coding. In contrast, MGS exploits the decreasing importance of the higher-frequency (higher-indexed) transform coefficients, that is, the coefficient with index 0 is most important, followed by the coefficients with indices 1 and 2, followed by the coefficients with indices 3–5, and so on. Accordingly, the lower-indexed coefficients form the lower (more important) MGS layers [3]. Formally, for a 4×4 block, we let $\mathbf{W} = [w_1, w_2, \ldots, w_{16}]$ denote a vector of MGS weights that satisfy $\sum_{m=1}^{16} w_m = 16$. The weight w_m gives the number of transform coefficients contained in MGS layer m. We denote M for the number of MGS layers (which is equal to the number of non-zero weights). For example, in Fig 2 the enhancement layer is divided into $M = 3$ MGS layers represented by $\mathbf{W} = [3, 3, 10]$, i.e., $w_1 = 3$, $w_2 = 3$, and $w_3 = 10$, and all weights that are not specified are set to zero, i.e., $w_4 = w_5 = \cdots = w_{16} = 0$. As another example consider $\mathbf{W} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ which divides the enhancement layer into $M = 16$ MGS layers, each containing one transform coefficient.

For the widely used CABAC entropy coding [54] in H.264, the transform coefficients of an 8×8 block are divided into MGS layers by extending the zig-zag pattern of the 4×4 block. In particular, each weight w_m is multiplied by four and the coefficients are considered in the zig-zag pattern of the 8×8 block. For the above example of $\mathbf{W} = [3, 3, 10]$, the first MGS layer is formed by the first 12 transform coefficients in the 8×8 block zig-zag scan, the second MGS layer is formed by the next 12 transform coefficients, and the third MGS layer is formed by the remaining 40 transform coefficients.

H.264 MGS allows for flexible bit rate (and video quality) adaptation by varying the number of MGS layers for each video frame (also referred to as an access unit in H.264 terminology). This high level of flexibility is enabled through a novel high-level signaling mechanism [60]. H.264 achieves this flexibility at very low cost in terms of reduced RD efficiency through a number of innovative scalable coding techniques. Mainly, these coding techniques introduce a novel trade-off between the RD coding efficiency and the so-called drift error that occurs when the enhancement layer of a frame is discarded and reduces the quality of dependent frames. MPEG-4 fine grain scalability (FGS) employed motion compensated prediction only for the base layer, while the enhancement layer of a frame was encoded with reference only to the base layer of the frame. While this approach avoided drift errors, it resulted in RD inefficient encoding. At the other extreme, quality scalable MPEG-2 used the aggregate quality of base layer and available enhancement layers of a reference frame for the motion compensated prediction of the base layers of dependent frames. This approach of using the highest available quality as reference, makes the encoding highly RD efficient, but very susceptible to drift errors.

H.264 SVC MGS introduced the novel *key picture* concept to combine the benefits of the outlined MPEG-4 FGS and MPEG-2 approaches. Key pictures are the frames in the temporal base layer (see Section II-B1). Similar to MPEG-4 FGS, key frames use only the base layer of other key pictures as reference for motion-compensation prediction; thus, limiting the propagation of drift errors to the frames between two successive temporal base layer frames. In the illustration in Fig. 3, the frames I_0, P_4, and I_8 are key pictures. The frame P_4 is forward predicted from the base layer of frame I_0.

Similar to MPEG-2, H.264 SVC MGS uses the highest available quality representation given by the aggregate of base layer plus available MGS layers for RD efficient motion compensated prediction of the base layer and the enhancement layer of the frames between key pictures. As illustrated in

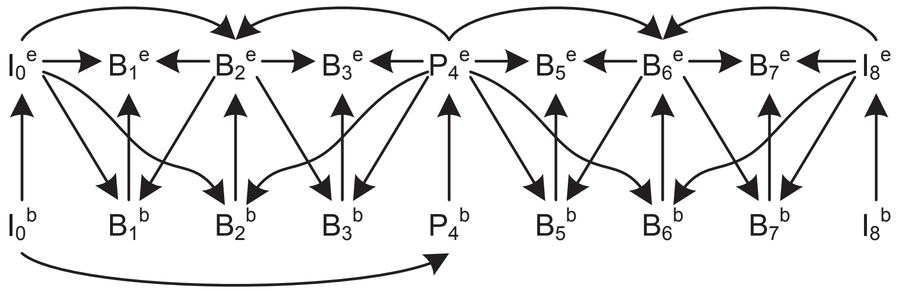


Fig. 3. Illustration of H.264 SVC MGS prediction dependencies for G8B3 GoP with a base layer (denoted by superscript b) and one enhancement layer (denoted by superscript e). The enhancement layer is partitioned into several MGS layers. The I and P frames are key frames, limiting the propagation of drift errors. The highest available qualities of the reference frames are used for the motion compensated prediction of the base layers of the B frames according to the hierarchical B frame structure.

Fig. 3, the prediction from the highest available quality follows the hierarchical B frame structure. For instance, the base layer of frame B_2 is predicted from the highest available qualities of frames I_0 and P_4. Following the hierarchical B frame structure, the highest available qualities of frames B_2 and I_0 predict the base layer of frame B_1. Thus, transmitting more MGS layers for frame B_2 can lead to an improvement in the PSNR quality of frame B_1, even though no additional MGS layers are transmitted for frame B_1.

In order to enable the adaptation of the number of MGS layers during network transport, each layer is typically separately packetized. For instance, with $M = 3$ MGS layers, each video frame results in a prefix NALU describing the frame, a NALU containing the encoded base layer, and three NALUs containing the MGS layers. Due to the prediction dependencies illustrated in Fig. 3, discarding an MGS layer from a reference frame (such as frame B_2) affects the quality of dependent frames (such as frames B_1 and B_3). Thus, the adaptation of MGS layers involves typically an RD optimization that seeks to first drop the MGS layers that make the smallest contribution to the average PSNR quality of the decoded video stream relative to their size (in byte), as discussed in more detail in Section III-C2.

III. Structure of H.264 SVC Video Traces

In this section we explain the structure of traces characterizing the different types of H.264 video encoding. We first present the basic trace structures for single-layer (non-scalable) video coding, followed by the additional features of the traces for layer-scalable and sublayer-scalable video coding.

There are two general types of video traces, namely (i) video frame size and quality traces, and (ii) offset distortion traces. For each video frame n, the offset distortion trace gives the PSNR video quality if subsequent frames $n+d, d = 1, 2, \ldots, g$, are replaced by frame n. Thus, the offset distortion traces permit the evaluation of the PSNR video quality if a correctly received frame (i.e., frame n) is re-displayed to conceal the loss of one or more subsequent frames [61], [62]. Unless noted otherwise we use the term *video trace* to refer to the set of both types of video traces for a given video encoding.

A. Single-layer H.264 Video Coding

Generally, a video trace library, including the library at `http://trace.eas.asu.edu` accompanying this tutorial, first requires the selection of a video title (sequence). Then, an encoded spatial resolution, such as the CIF 352×288 pixel or full HD 1920×1080 pixel format, a GoP pattern GgBb, and a quantization parameter (scale) q are usually selected to arrive at the video trace for a prescribed video encoding. Throughout, for single-layer video coding, we focus on encodings with a *fixed* quantization parameter q, which is also often referred to as variable bit rate (VBR) encoding [63], [64]. VBR encoding results typically in relatively small quality variations of the encoded video frames at the expense of large variations of the encoded frame sizes. In contrast, so-called constant bit rate (CBR) video encoding adjusts the quantization parameter so as to keep the frame sizes nearly constant at the expense of relatively large frame quality variations.

1) Frame Timing Characterization: We let $n,\ n = 0, 1, 2, \ldots, N-1$, denote the number of a given frame in display order, i.e., the order in which the frames are captured when shooting the video and displayed when playing back the video. Note that the frames are typically encoded in a different order—the so-called encoding order—to account for the frame dependencies of the motion compensated prediction, see Section II and Fig. 1. Let Δ denote the frame period (display time) of a frame in seconds, which is the inverse of the frame (display) frequency (in frames/second). In many current videos, the frame rate is 30 frames/s (or 25 frames/s), corresponding to a frame period of $\Delta = 33.33$ ms (or $\Delta = 40$ ms). Assuming that frame $n = 0$ is displayed at time zero, let $\delta_n,\ n = 0, 1, \ldots, N-1$, denote the time instant that frame n should be displayed. Note that δ_n is equivalent to the cumulative display time (duration) up to and including the display time of frame $n-1$, i.e., $\delta_n = n\Delta$.

2) Frame Size Characterization: We let $X_n^q,\ n = 0, 1, \ldots, N-1$, denote the frame size (number of byte) of video frame n encoded (compressed) with the (fixed) quantization parameter q. In order to avoid excessive clutter in the notation, we omit the superscript q when the dependency on the quantization parameter setting q is not directly relevant. The size X_n generally includes only the encoded video data from the video coding layer (VCL) and not the network encapsulation overhead that is added by the network abstraction layer (NAL). In particular, the size X_n includes the network abstraction layer unit (NALU) encapsulation overhead for frame n, i.e., the one-byte header, but no additional NALUs that are commonly inserted into the encoded bitstream, such as frame prefixes or stream-level information.

3) Video Frame Quality Characterization: Objective video quality assessment methods that accurately predict the subjective quality assessment of human viewers through automated computational signal processing techniques have been widely researched [65]–[69]. The peak signal to noise ratio (PSNR) has been found to give only moderately accurate approximations of the subjective video quality. Nevertheless, the PSNR is still widely used in video networking studies and video trace libraries due to its conceptual and computational simplicity. We let $Q_n^{q,Y},\ n = 0, 1, \ldots, N-1$, denote the quality in terms of the PSNR (in dB) of the luminance component of video frame n encoded with quantization parameter q (and subsequently decoded). Similarly, let $Q_n^{q,U}$ and $Q_n^{q,V},\ n = 0, 1, \ldots, N-1$, denote the corresponding qualities of the two chrominance components (U and V). Again, we omit the superscript q when not needed.

The offset distortion $Q_n^Y(d),\ n = 0, 1, \ldots, N-1,\ d = 1, 2, \ldots, g$, denotes the luminance PSNR frame quality if instead of frame $n+d$ the frame n is displayed ($Q_n^U(d)$ and $Q_n^V(d)$ are defined analogously). Thus, the offset distortion can be used to obtain the PSNR frame quality if the loss of a frame $n+d$ during network transport is concealed by the re-display of a received earlier frame n.

Some video traces provide additional video quality characterizations. For instance, the traces for the HD single-layer H.264 SVC encoded videos in the trace library at `http://trace.eas.asu.edu` include video quality metric (VQM) scores [70], [71]. The VQM score is a number between zero and one, whereby a score in the range 0–0.2 indicates excellent video quality, a score in 0.2–0.4 indicates good video quality, and higher scores indicate lower qualities. The VQM scores were obtained with the command-line VQM (CVQM) tool for successive five second segments of the video; thus, all frames within a given five second segment have the same VQM score.

4) Trace Format: A trace file (for brevity often referred to as *trace*) is typically a text file of ASCII characters. A trace file usually consists of a header portion and the actual video trace in table format. The header gives general information about the video, such as video title, frame (spatial) resolution, and frame frequency, as well as its encoding, such as employed encoder, GoP pattern, and quantization parameters. Additionally, the header may contain basic statistics for the video trace, such as minimum, maximum, and average frame size.

The tabular portion of the video trace file gives the characterization of the encoded video frames, whereby one line (row) of the table is dedicated to each encoded video frame. The line for a given frame gives usually frame number n, display instant δ_n, frame type (I, P, or B), frame size X_n (in byte), PSNR frame qualities Q_n^Y, Q_n^U, and Q_n^V, and, if available, additional quality characterizations. A trace containing all these detailed characterizations of an encoded frame is referred to as *verbose*, whereas a *terse* trace gives commonly only the frame size (and sometimes the frame number). We also note that some traces provide the frames in display order, whereas others provide them in encoder order. However, in this tutorial we let n consistently denote the frame number in display order.

B. Layer-scalable H.264 SVC Video Coding

The trace for each layer of a layer-scalable H.264 video encoding is similar in structure to the trace of a single-layer encoding. Generally, the frame sizes in a layer trace correspond to the sizes (in byte) of the considered coding layer. However, the quality characterization in a given layer trace corresponds to the video quality of the aggregate of all layers up to and including the considered layer.

Temporal scalability, i.e., the adaptation of the frame frequency, can be flexibly combined with any of the other H.264 SVC scalability modes. The frame size and quality traces for a prescribed video coding consist typically of a set of $\tau + 1$ traces, whereby each trace characterizes a given frame frequency. In particular, the trace for the lowest frame frequency corresponding to the temporal base layer $T = 0$ gives the frame sizes of the I and P frames, whereas the sizes of all other frames (which are in the higher temporal layers $T > 0$) are zero. Furthermore, the trace gives the video frame quality characterizations of the I and P frames. For the other frames, the trace gives the PSNR video frame quality that is achieved when re-displaying the preceding I or P frame in place of the considered frame. That is, the PSNR values from the offset distortion trace are included for the frames missing in a lower temporal layer to simulate the effect of re-display of the last successfully decoded frame.

The trace corresponding to temporal layer $T = 1$ includes the frames from the temporal base layer and temporal enhancement layer $T = 1$, effectively doubling the frame frequency compared to the temporal base layer. That is, the sizes and PSNR qualities of the frames in temporal layers $T = 0$ and 1 are given in the trace; whereas, for the frames in the higher temporal layers $T = 2, \ldots, \tau$, the size is given as zero and the quality is given as the offset distortion values of the frames in temporal layers $T = 0,\ 1$.

This structure of the temporal layer traces applies analogously to the higher temporal layers. The trace for the highest temporal layer $T = \tau$ contains all frame sizes and PSNR frame qualities and corresponds to the full frame frequency of the encoded video sequence.

C. Sublayer-scalable H.264 SVC MGS Coding

In addition to the video and encoding parameters characterizing a single-layer encoding, the selection of a vector of MGS weights is required to arrive at a prescribed H.264 MGS encoding and its corresponding set of traces in a video trace library. In the following we present the different MGS trace structures that are presently available. We focus on H.264 SVC MGS encodings with a base layer and one enhancement layer, which is divided into M MGS layers.

1) MGS Layer Traces: For an H.264 SVC MGS encoding with a base layer and M MGS layers, there are $M + 1$ MGS (quality) layer traces, which we index with $m = 0, 1, \ldots, M$. The layer trace $m = 0$ gives the sizes X_n^0 (in byte) and frame qualities Q_n^0 of the base layer of each frame n. Layer trace

$m,\ m = 1, \ldots, M$, gives the frame sizes X_n^m in MGS layer m and the video frame qualities Q_n^m for the aggregate of the base layer plus the MGS layers up to and including the considered MGS layer m. The total aggregate size for encoded frame n is obtained as $X_n = \sum_{m=0}^{M} X_n^m$.

2) Priority Level and Target Bit Rate Traces: A video bitstream with a target bit rate ranging from the bitrate of the base layer to the bitrate of the complete encoding can be extracted from an H.264 SVC MGS encoding by dropping selected MGS layers from selected frames. The extraction of a video stream with a prescribed target bit rate from a given H.264 MGS encoding so as to maximize the video quality poses a complex optimization problem. The complexity is introduced by the H.264 SVC MGS inter-frame dependencies, illustrated in Fig. 3, which make the lower layers of a frame dependent on the higher layers of a different frame. Dropping an MGS layer for a specific frame reduces not only the quality for this frame, but, typically, also the qualities of all dependent frames, each of which in turn may have several dependent frames. Good solutions with reasonable computational effort to this optimization problem are currently researched, see e.g., [21], [72]–[75].

We briefly describe two widely considered extraction mechanisms, namely the extraction mechanism [20] in the H.264 SVC Joint Scalable Video Model (JSVM) reference software (which leads to priority level traces) and the H.264 SVC temporal layer based extraction mechanism [73], [76] (which leads to the MGS-temporal layers in Section III-C3).

The H.264 SVC JSVM reference extraction approach [20], which is implemented in the reference *QualityLevelAssigner* and *BitstreamExtractor* tools, assigns a fixed priority (quality) level $P_n^0 = 63$ to the base layer of each frame n and a priority level P_n^m ranging from 0 (lowest priority) to 63 (highest priority) to each MGS layer m of each frame n. The priority levels are evaluated with RD optimization strategies using both the encoded bitstream and the original (unencoded) video source. The RD optimization seeks to maximize the PSNR video quality for each frame while meeting a given target bitrate over the entire duration of the video sequence.

The priority level traces characterize the frame sizes and PSNR frame qualities corresponding to the video bit stream extracted with a prescribed priority level P. In particular, for a given H.264 SVC MGS encoding there are 64 priority level traces indexed by the priority levels $P,\ P = 63, 62, \ldots, 1, 0$. The priority level trace P gives for each frame n the aggregate size ($X_n = \sum_{m: P_n^m \geq P} X_n^m$ in byte) and the PSNR frame quality of the MGS layers (plus the base layer) with a priority level of P or higher (i.e., priority levels $P,\ P+1,\ \ldots, 63$), as well as the number of included MGS layers $m, m = 0, 1, \ldots, M$. Note that $m = 0$ indicates that only the base layer is included for the frames, whereas $m = M$ means that all M MGS layers are included.

After the priority levels have been assigned, the bitstream is extracted by dropping the MGS layers with the lowest priority level $P = 0$ from all frames, then the MGS layers with the next lowest priority level $P = 1$, and so on, until the target bit rate is met. With this extraction approach, the number of MGS layers included (streamed) can vary significantly from frame to frame from zero to the maximally available number of MGS layers M. The target bit rate traces characterize the frame sizes and PSNR frame qualities as well as the number of included MGS layers.

Slight differences between the target bitrate and the average bitrate of the extracted stream are due to the encapsulation overhead (such as stream-level and frame prefix NALUs) which is considered by the extraction method, but not included in video traces giving only the size of video coding data. Also, the last incomplete GoP of a video sequence is taken into account by the extraction method, but is dropped in video traces including only full GoPs.

3) MGS-Temporal Layer Traces: The extraction mechanism based on the H.264 SVC temporal layers [73], [76] does not require the compute intensive RD optimized priority level assignment. Instead, this approach builds on the frame dependencies of the H.264 SVC temporal layers. In particular, the approach proceeds by first dropping the highest MGS layer M from the frames in the highest temporal layer τ, followed by the second highest MGS layer $M - 1$ from the frames in the highest temporal layer τ, and so on until the target bit rate is met. Once all MGS layers have been dropped (i.e., only the base layer remains) from the frames in the highest temporal layer τ, then the highest MGS layer M is dropped from the frames in the second highest temporal layer $\tau - 1$, then the second highest MGS layer $M - 1$ is dropped from these frames, and so on, until the target bit rate is met.

Formally, we define $\mu,\ \mu = 0, 1, \ldots, (\tau + 1)M$, as the MGS-temporal extraction threshold. The threshold $\mu = 0$ corresponds to transmitting only the base layer for all frames; we do not consider combinations of temporal scalability with MGS scalability, which would correspond to dropping the base layer for some frames. At the other extreme, the threshold $\mu = (\tau + 1)M$ corresponds to transmitting the base layer and all M MGS layers for the frames in all $(\tau + 1)$ temporal layers. For $\mu = 0, 1, \ldots, (\tau + 1)M - 1$, we define $T_\mu = \lfloor \frac{\mu}{M} \rfloor$ as the corresponding temporal layer. Frames in temporal layers $T = 0, \ldots, T_\mu - 1$ are extracted with the base layer and with all M MGS layers (whereby $T_\mu = 0$ means that no temporal layer has all M MGS layers), frames in temporal layer $T_\mu = \lfloor \frac{\mu}{M} \rfloor$ are extracted with the base layer and $m_\mu = \mu - T_\mu M$ MGS layers, and frames in temporal layers $T = T_\mu + 1, \ldots, \tau$ are extracted with only the base layer. For a given H.264 SVC MGS encoding there are thus a total of $(\tau + 1)M + 1$ MGS-temporal layer traces. Note that an extracted stream consisting of the base layer and all M MGS layers for temporal layers $0, 1, \ldots, T_\mu - 1$, base layer and m_μ MGS layers for temporal layer T_μ, and only the base layer for temporal layers $T_\mu + 1, \ldots, \tau$, has the corresponding MGS-temporal extraction threshold $\mu = T_\mu M + m_\mu$.

An alternate notation for MGS-temporal traces is the vector $(m_0, m_1, \ldots, m_\tau)$, whereby $m_T,\ m_T = 0, 1, \ldots, M$, denotes the number of MGS layers that are included for frames in temporal layer $T,\ T = 0, 1, \ldots, \tau$. The extraction threshold $\mu = T_\mu M + m_\mu$ corresponds to the vector notation $(m_0 = M, m_1 = M, \ldots, m_{T_\mu - 1} = M, m_{T_\mu} = m_\mu, m_{T_\mu + 1} = 0, \ldots, m_\tau = 0)$. We note that for some videos, the full set of MGS-temporal layer traces with $m_0 \geq m_1 \geq \cdots \geq m_\tau$ is available. For other videos, only the limited set of MGS-temporal layer traces indexed by the extraction threshold

μ, $\mu = 0, 1, \ldots, (\tau + 1)M$, is available, and we consider this more limited set throughout the remainder of this tutorial.

The MGS-temporal layer based extraction can be performed to meet a prescribed target bit rate over a range of time horizons; for instance, over one GoP, over multiple GoPs, or over the entire length of a video sequence. Performing the extraction over time horizons longer than one GoP may result in variations of the average bit rates of the individual GoPs contained in the considered time horizon; only the average bit rate over all GoPs in the considered time horizon meets the target bit rate.

IV. Traffic and Quality Characteristics of H.264 Encoded Video

In this section we (i) illustrate the H.264 video traffic and quality characteristics through the specific example sequences *Big Buck Bunny* and *Elephants Dream* and (ii) summarize the H.264 video traffic and quality characteristics obtained from extensive studies conducted for large sets of long video sequences covering a wide variety of video content genres. The purpose of this section is to inform communications and networking generalists about the typical H.264 video traffic and quality characteristics so as to guide the application and usage of H.264 encoded video in video communication and networking research.

The video traces corresponding to all H.264 video traffic and quality characteristics presented in this section are available from the video trace library `http://trace.eas.asu.edu` to facilitate their usage in communications and networking studies. In addition, this video trace library provides the video bit streams from the different single-layer encodings of *Big Buck Bunny* and *Elephants Dream*, which are freely-licensed films.

Big Buck Bunny (14,315 frames) and *Elephants Dream* (15,691 frames) are two popular animated video sequences in the HD 1920 × 1080 pixels format with a frame rate of 24 frames/s. The *Big Buck Bunny* video has relatively high levels of texture (detail) in the visual appearance of the main characters while the motion levels vary from low motion scenes to high motion scenes. The *Elephants Dream* video has relative less texture detail while having generally consistently high levels of motion and many instances of abrupt motion.

We converted the individual video frames (images) to the YUV format, whereby Y denotes the luminance (brightness) component while U and V denote the two chrominance (color) components, using the open-source FFMPEG tool (`http://www.ffmpeg.org`) and down-sampled the thus generated YUV video to CIF format using the tools available with the JSVM reference software. The videos were encoded in MPEG-4, H.264/AVC, and H.264 SVC with the respective MPEG/ITU reference software encoders. The H.264 encodings employed CABAC and Lagrangian RD optimization.

In the following evaluation of the video traffic and quality characteristics, we focus primarily on the rate-distortion (RD) curve and the rate variability-distortion (VD) curve which we define for the specific notation of single-layer encoded video, see Section III-A. These definitions are analogously applied to the aggregate frame size and quality of a stream extracted from a layer-scalable or sublayer-scalable encoding. The rate-distortion (RD) curve is a plot of the average luminance PSNR video quality

$$\bar{Q}^{q,Y} = \frac{1}{N} \sum_{n=0}^{N-1} Q_n^{q,Y} \qquad (1)$$

as a function of the average bit rate, which is obtained by dividing the average frame size

$$\bar{X}^q = \frac{1}{N} \sum_{n=0}^{N-1} X_n^q \qquad (2)$$

by the frame period, i.e., as $\bar{X}^q/\Delta$. In particular, the individual frame sizes X_n^q and qualities $Q_n^{q,Y}$ are from the video trace for quantization parameter q and the RD curve is obtained by plotting the points $\left(\bar{X}^q, \bar{Q}^{q,Y}\right)$ for a range of quantization parameters q. We remark that an alternative average luminance PSNR video quality $\bar{Q}'^{Y}$ can be obtained by first averaging the mean square error (MSE) [65], [69] values for the individual frames $M_n,\ n = 0, \ldots, N-1$ to obtain the mean MSE $\bar{M} = \frac{1}{N}\sum_{m=0}^{N-1} M_n$, followed by the conversion to PSNR (in dB) with $\bar{Q}'^{Y} = 10\log_{10}(255^2/\bar{M})$. We consider the video quality $\bar{Q}^{q,Y}$ obtained with the averaging in (1) throughout this article.

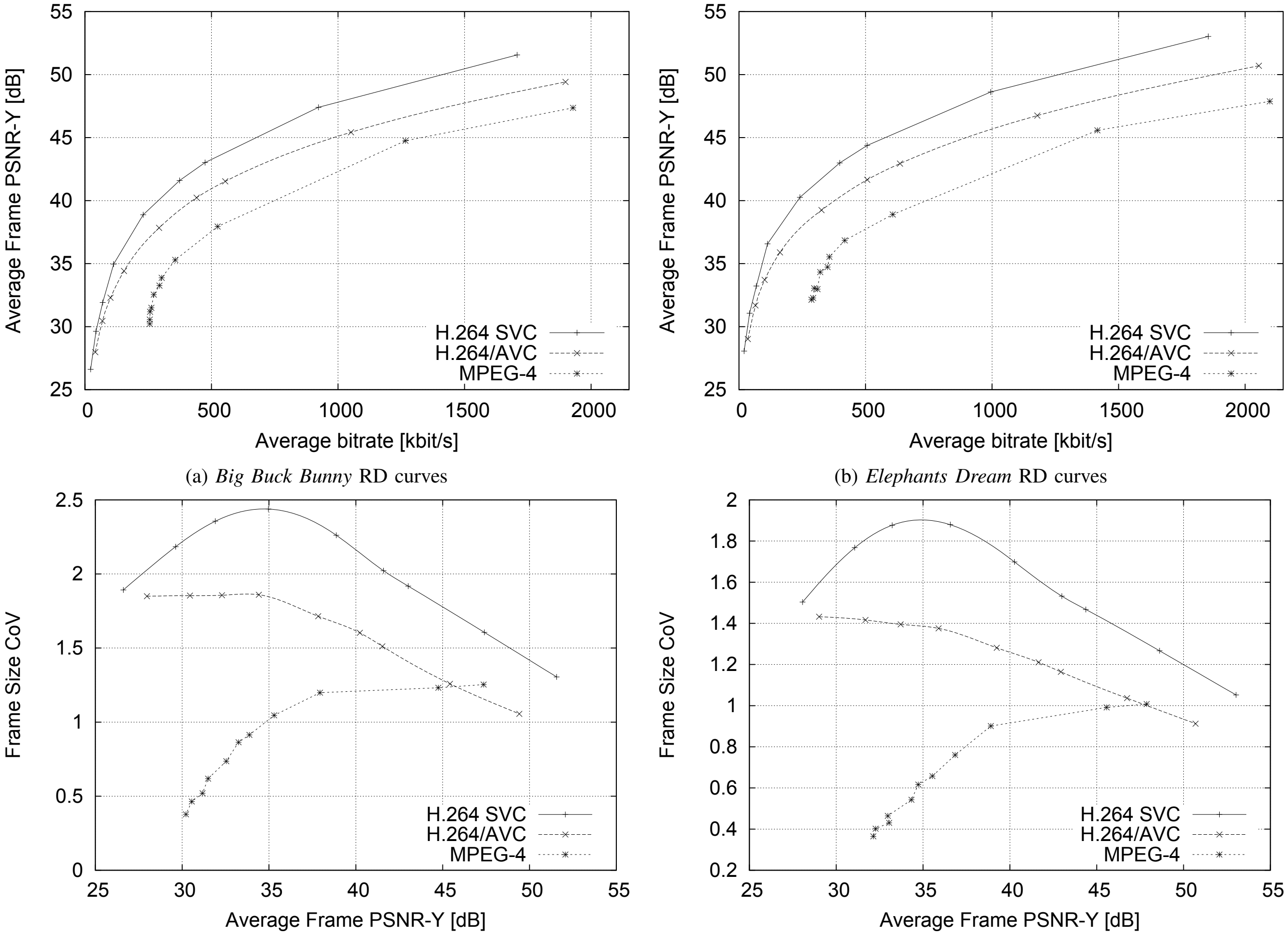
We characterize the video traffic variability with the rate variability-distortion (VD) curve [77], which is a plot of the standard deviation of the frame sizes normalized by the average frame size, i.e., of the coefficient of variation of the frame sizes

$$\mathrm{CoV}_X^q = \frac{1}{\bar{X}^q}\sqrt{\frac{1}{(N-1)}\sum_{n=0}^{N-1}(X_n^q - \bar{X}^q)^2}, \qquad (3)$$

as a function of the average luminance PSNR video quality $\bar{Q}^{q,Y}$. That is, the VD curve is obtained by plotting the points $\left(\bar{Q}^{q,Y}, \mathrm{CoV}_X^q\right)$ for a range of quantization parameters q.

A. Single-layer (Non-scalable) H.264 Video Coding

In Figure 4, we compare the RD and VD curves for *Big Buck Bunny* and *Elephants Dream* encoded with MPEG-4, H.264/AVC with classical B frames, and H.264 SVC with hierarchical B frames. The RD and VD results in Fig. 4 illustrate the general encoder characteristics verified for large sets of video test sequences [11], [12]: Compared to MPEG-4, H.264/AVC with classical B frames achieves significantly higher RD efficiency. H.264 SVC with hierarchical B frames, in turn, achieves significantly higher RD efficiency than H.264/AVC with classical B frames. In many typical encoding scenarios, H.264/AVC and H.264 SVC achieve more than double the compression ratio (of uncompressed frame size to mean compressed frame size) for the same PSNR video quality. For instance, we observe from Figs. 4(a) and (b) that for *Big Bug Bunny* and *Elephants Dream*, MPEG-4 achieves an average frame PSNR of 40 dB with a bit rate of approximately 730 kbps. H.264/AVC achieves 40 dB for *Elephants Dream* with approximately 375 kbps, while the finer textured *Big Buck Bunny* requires approximately 430 kbps. In turn, H.264 SVC is yet more RD efficient, achieving the 40 dB average frame PSNR with approximately 230 and 280 kbps for *Elephants Dream* and *Big Buck Bunny*, respectively.



(a) *Big Buck Bunny* RD curves

(b) *Elephants Dream* RD curves

(c) *Big Buck Bunny* VD curves

(d) *Elephants Dream* VD curves

Fig. 4. Rate-distortion (RD) and rate variability-distortion (VD) plots for the *Big Buck Bunny* and *Elephants Dream* video sequences in CIF format encoded in MPEG-4, H.264/AVC, and H.264 SVC with a single layer.

However, the substantial increases in compression efficiency with the H.264 encoders come at the expense of vastly increased video traffic variability, as indicated by the significantly higher CoV_X values. We observe from Figs. 4 (c) and (d) that for *Big Buck Bunny* and *Elephants Dream* the maximum CoV_X values increase from 1.25 and 1.0 with MPEG-4, to 1.85 and 1.4 with H.264/AVC, and then to 2.4 and 1.9 with H.264 SVC. We also note that *Big Buck Bunny* with its wide range of motion levels in the different scenes has higher CoV_X values than *Elephants Dream* with its consistently high motion. Generally, the coefficient of variation of the frame sizes CoV_X is typically below 1.5 for MPEG-4 [77], while the CoV_X can reach values around 2.4 for H.264/AVC and even values above 3.0 for H.264 SVC.

As a result of these higher traffic variabilities, H.264/AVC can support more video streams than H.264 SVC and even MPEG-4 can support more video streams than H.264/AVC and SVC when multiplexing few unsmoothed video streams over a fixed-capacity bufferless link [12]. Appropriate video traffic management, e.g., through buffering and smoothing, is therefore critical to retain the encoding performance gains of H.264/AVC and SVC during network transport. Importantly, H.264 SVC requires typically smoothing windows (i.e., blocks of video frames that are averaged) or multiplexer buffers twice as large as H.264/AVC to translate the encoding performance gain of H.264 SVC into a commensurate gain in the number of supported video streams over a given link [78].

Other key conclusions from extensive encoding experiments and traffic and quality analyses are:

- The RD efficiency of H.264 SVC with hierarchical B frames improves with increasing number of B frames in a GoP, i.e., G16B15 is the most RD efficient GoP structure with 16 frames in a GoP (but also has the highest traffic variability). For H.264/AVC with classical B frames, the GoP structure G16B3 tends to give the highest RD efficiency for 16-frame GoPs.
- The long-range dependence characteristics [79] of H.264/AVC and SVC video traffic a generally similar to those of MPEG-4 video traffic.
- The traffic characteristics of HD video encoded with the H.264 fidelity range extension (FRExt) cannot be obtained by simply scaling the traffic of H.264 video encodings with lower spatial resolutions. Instead, traces of actual encoded HD video are required for a realistic representation of HD video traffic.

B. Layer-scalable H.264 Video Coding

The main insights from extensive temporal and spatial layer scalable encodings are that the H.264 SVC layers have significantly higher RD efficiency and traffic variability than the corresponding layers of layer-scalable MPEG-4 encodings. After applying the same basic window smoothing, H.264 SVC layers still have higher traffic variability than MPEG-4 layers [15]. For combined spatio-temporal scalable H.264 SVC encodings, the characteristics of the temporal layers of each spatial resolution of the combined encoding are well approximated by the temporal layers in encodings for a fixed spatial resolution (i.e., encodings without spatial scalability).

In Figure 5, we include the RD and VD points for *Big Buck Bunny* and *Elephants Dream* for the H.264 SVC coarse grain scalability (CGS) encodings that fall into the plotted range. We encoded the sequences in CIF format with one base layer and three enhancement layers with respective quantization parameters 48, 38, 28, and 18. The RD points corresponding to the aggregate stream of the base layer and the first and second CGS enhancement layer(s) are included in Figs. 5(a) and (b). Additionally, the base layer and the aggregate stream up to and including the third enhancement layer for *Big Buck Bunny* have average bit rates of 36.4 kb/s and 1.09 Mb/s and average frame qualities of $\bar{Q}^Y = 27.5$ dB and 46.7 dB, and for *Elephants Dream* have average bit rates of 36.4 kb/s and 1.23 Mb/s and average frame qualities of $\bar{Q}^Y = 29.2$ dB and 48.5 dB, respectively (whereby these additional values lie outside the range of the plots in Fig. 5). We observe from Fig. 5 that H.264 CGS has lower RD efficiency compared to H.264 single-layer encodings. For a given PSNR video quality, the H.264 CGS bitrate is 18–40 % higher than the corresponding H.264 single-layer bit rate. More extensive studies [80] confirmed these typical results and found that larger differences in the quantization parameters of the CGS layers (and correspondingly fewer CGS layers) lead to slightly smaller bit rate overheads of 10–30 % for encodings with two CGS enhancement layers. However, for smaller quantization parameter differences, there are typically substantially higher bit rate overheads on the order of 30–80 %. Thus, with H.264 CGS the flexibility of adapting video bit rate and quality by adding or dropping CGS layers comes at the expense of a relatively high bit rate overhead.

The CoV_X values for the H.264 CGS encodings are in the range from 1.2–1.4 for *Big Buck Bunny* and in the range from 1.0–1.2 for *Elephants Dream*, whereby the point for the second enhancement layer is included in the plots in Fig. 5(c) and (d), respectively. These results indicate that the H.264 CGS streams have smaller traffic variabilities than the corresponding H.264 single-layer streams. In additional evaluations, we found that smoothing over one GoP reduces the CoV_X values, significantly, down to 0.65–0.9 for both sequences. Thus, at the GoP timescale, H.264 CGS and single-layer encodings have similar traffic variability; whereas, at the frame timescale, H.264 CGS has significantly lower traffic variability than the corresponding single-layer encodings.

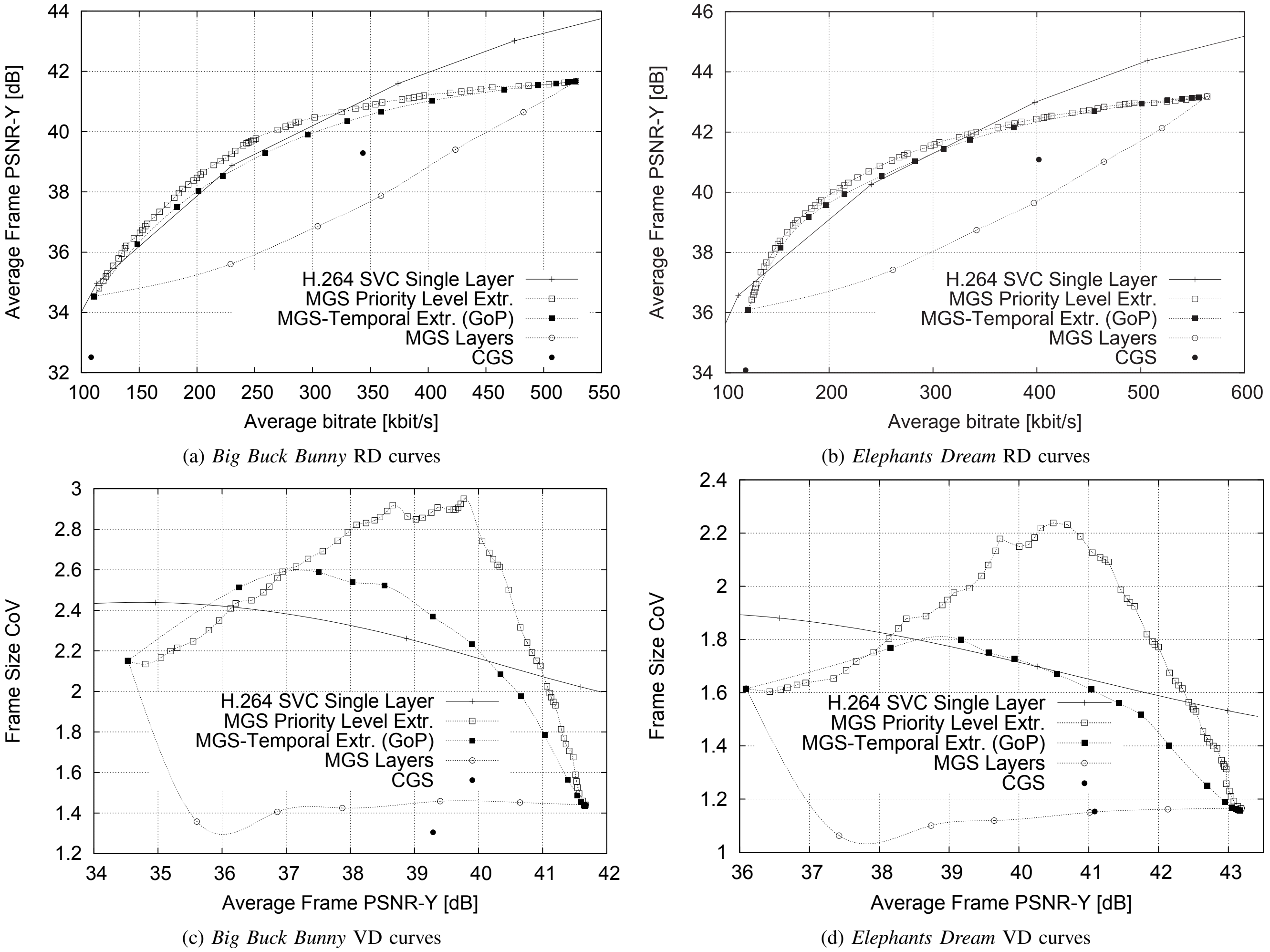
C. Sublayer-scalable H.264 Video Coding

In Figure 5, we plot the RD and VD curves of H.264 SVC MGS with MGS layer extraction, priority level extraction, and MGS-temporal layer extraction on the GoP time scale. We consider H.264 MGS encodings with one enhancement layer (with quantization parameter 35 for the base layer and 25 for the enhancement layer). We employ the MGS weights $\mathbf{W} = [1, 2, 2, 3, 4, 4]$. For these MGS encodings we employ slight encoder parameter optimizations over the basic non-scalable encoding setting so as to achieve good RD performance for practically realistic encoding times. Specifically, we reduce the search range for the motion-compensated prediction from 32 pixels for the single-layer to 16 for the MGS encodings. We employ the Hadamard transform domain distortion measure for sub-sample (sub-pixel) motion search with 16 search iterations and weighted bidirectional prediction for MGS encodings; while the single-layer encodings employ the default sum of absolute differences distortion measure with four search iterations and without weighted bi-directional prediction.

We observe from Figs. 5(a) and (b) that the MGS extraction method has a profound impact on the RD performance. Whereas extracting MGS layer by MGS layer gives rather poor RD performance, the extraction by priority levels gives excellent RD performance that closely approaches or even slightly exceeds the corresponding single layer RD performance in the lower to mid bit rate range. The priority level assignment P_n^m to the individual MGS layers m, $m = 0, 1, \ldots, M$, of the individual video frames $n = 0, 1, \ldots, N-1$, is based on compute-intensive RD optimization techniques that consider the RD efficiency, i.e., the contribution of each MGS layer m of each frame n toward the average PSNR video quality $\bar{Q}^Y$ of the video sequence relative to its size X_n^m (in byte). In the lower to mid bit rate range up to approximately 300 kbps, only those MGS layers from those frames that have the highest RD efficiency are extracted from the encoded bit stream for transmission. As we approach the upper end of the bit rate range, all MGS layers from all frames, i.e., even the least RD efficient MGS layers, are included. Thus, the RD optimal extraction can no longer mask the bit rate overhead for the scalable encoding, resulting in the drop-off of the H.264 MGS RD curves relative to the H.264 single-layer RD curves at the upper end of the bit rate range.

In contrast to the priority level based extraction, MGS layer based extraction gives poor RD performance as it does not consider the RD efficiency properties of the individual MGS layers of the individual video frames. The MGS layer extraction includes the same number of MGS layers m for each frame n irrespective of the position of frame n in the hierarchical B frame prediction structure. The MGS layer extraction thus counteracts the H.264 SVC MGS encoding strategy of utilizing high quality reference frames for the predictive encoding of lower layers of dependent frames (see Fig. 3).

We further observe from Fig. 5 that MGS-temporal layer based extraction provides a good compromise between high RD efficiency and low computational complexity of the extraction method. The MGS-temporal layer based extraction



(a) *Big Buck Bunny* RD curves

(b) *Elephants Dream* RD curves

(c) *Big Buck Bunny* VD curves

(d) *Elephants Dream* VD curves

Fig. 5. RD and VD plots for the *Big Buck Bunny* and *Elephants Dream* video sequences in CIF format encoded in H.264 SVC MGS with base layer and one enhancement layer with MGS vector $\mathbf{W} = \{1, 2, 2, 3, 4, 4\}$ for different extraction approaches. Also included are RD and VD points for H.264 SVC CGS.

considers the inter-frame dependencies in the hierarchical B frame encoding. In particular, to reduce the video bit rate, MGS layers are first dropped from the frames with the least number of dependent frames so as to minimize the quality degradations of the dependent frames. As we observe in Fig. 5, and is confirmed in [80], this MGS-temporal layer strategy conducted for each individual GoP closely approximates the RD performance of the priority level extraction while avoiding its high computational cost. Conducting the MGS-temporal layer based extraction over the entire video sequence gives essentially identical results to the priority level extraction (which also operates over the entire video sequence).

Another main characteristic of the H.264 SVC MGS RD performance [80] is that the setting of the MGS weights $\mathbf{W}$ has a relatively minor impact on the RD performance. The weight vector $\mathbf{W} = [1, 2, 2, 3, 4, 4]$ considered in Fig. 5 provides sufficiently many MGS layers M for highly flexible bit rate adaptation. Increasing the number of MGS layers to $M = 12$, e.g., with the weight vector $\mathbf{W} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3]$, gives very slightly lower RD performance due to the increased scalable coding overhead.

Furthermore, a main characteristic of H.264 SVC MGS is that the quantization parameters of the base layer and enhancement layer can be used to adjust the range over which the bit rate can be adapted. A wider bit rate range comes at the expense of slightly reduced RD efficiency due to the relatively lower-quality encoding reference provided by a lower-quality base layer. Importantly, as we observed in Fig. 5(a) and (b), the H.264 MGS RD performance drops off toward the upper end of the bit rate range. Therefore, the enhancement layer quantization parameter should be chosen such that the upper end of the quality range, i.e., the video quality with the full enhancement layer, is 1–2 dB higher than the desired upper end of the quality adaptation range.

Turning to the VD results in Fig. 5, we observe for the priority level extraction in the range of high RD efficiency (from about 36 dB to 41 dB for *Big Buck Bunny*), a significantly higher frame size CoV compared to the single-layer encoding. This indicates that the extraction of those MGS layers from those frames with the highest RD efficiency increases the variability (burstiness) of the video traffic. Effectively, the priority level extraction selects MGS layers such that relatively large frames become even larger, while small frames remain small. The VD behavior of the GoP time scale MGS-temporal layer extraction in the range of high RD efficiency is similar in that it exceeds the variability of the single-layer encoding;

although, the variability increase is less pronounced than for priority level extraction. (Additional evaluations, not included to avoid clutter, indicate that MGS-temporal layer extraction over the full video sequence gives similarly high frame size CoV values as the priority level extraction.) Toward the upper end of the quality range, the CoV values of priority and MGS-temporal layer extraction sharply drop, mainly because the normalizing average frame size $\bar{X}$ in Eqn. (3) becomes relatively larger as the RD efficiency of the encoding drops off.

We briefly remark that the slight differences in RD values and somewhat more pronounced differences in VD values between the single-layer encodings and the base layer (left-most point) of the MGS encodings in Fig. 5 are due to the slight encoding parameter optimizations for these MGS encodings.

Additional evaluations for the GoP time scale [80] have revealed that smoothing the frames over each GoP removes the variability that has been added by the MGS layer extraction. In particular, the GoP size CoV values of the MGS streams with both priority and MGS-temporal layer extraction conducted over the full video sequence are very close to the GoP size CoV for the single-layer encoding.

V. Evaluating Network Transport with H.264 Video Traces

Video traces can be employed to simulate video traffic in a wide range of network simulations. We refer to [9], [62] for general instructions, presented in the context of MPEG-4 video, for generating video traffic workloads from video traces as well as the estimation of video related performance metrics, such as the frame starvation probability. In this section, we focus on the unique aspects that arise in simulations with H.264 video traces. Throughout, for improved clarity, we focus on the transport of a single video stream.

A. Single-layer (non-scalable) H.264 Video

1) Playout Timing and Delays: The hierarchical B frame prediction in H.264 SVC introduces larger delays than the classical B frame prediction used in MPEG-4 (and by default in H.264/AVC). We compare both B frame prediction structures for a live streaming scenario without frame smoothing in Figure 6 and refer to [78] for a study considering the streaming of pre-recorded video and frame smoothing in detail. We consider the GoP structure G16B15, i.e., $\beta = 15$ B frames between successive I frames. We suppose that the frame capture time is negligible and that it takes one frame period each to encode, transmit, and decode a frame.

In Fig. 6, the video frames are denoted by frame type and capture (recording) time instant. For instance, I_0 denotes the intracoded (I) frame captured at time 0, while B_8 denotes the bidirectionally coded (B) frame captured at time instant 8Δ. The encode, transport, and decode time axes indicate when a frame is encoded, transmitted, and decoded. For instance, in Fig. 6(a), frame B_8 is encoded during the frame period from 24Δ to 25Δ, transmitted during the frame period from 25Δ to 26Δ, and decoded during the frame period from 26Δ to 27Δ. The display time axis gives the instants that each video frame is played out (displayed) on the screen. The offset (timeshift) between the capture axis (top axis) and the display axis (bottom axis) is the delay introduced by the video coding and transmission between the video capture (recording) and the video display. For the scenario illustrated in Fig. 6, we observe a delay of 19Δ with classical B frames and a delay of 22Δ with hierarchical B frames.

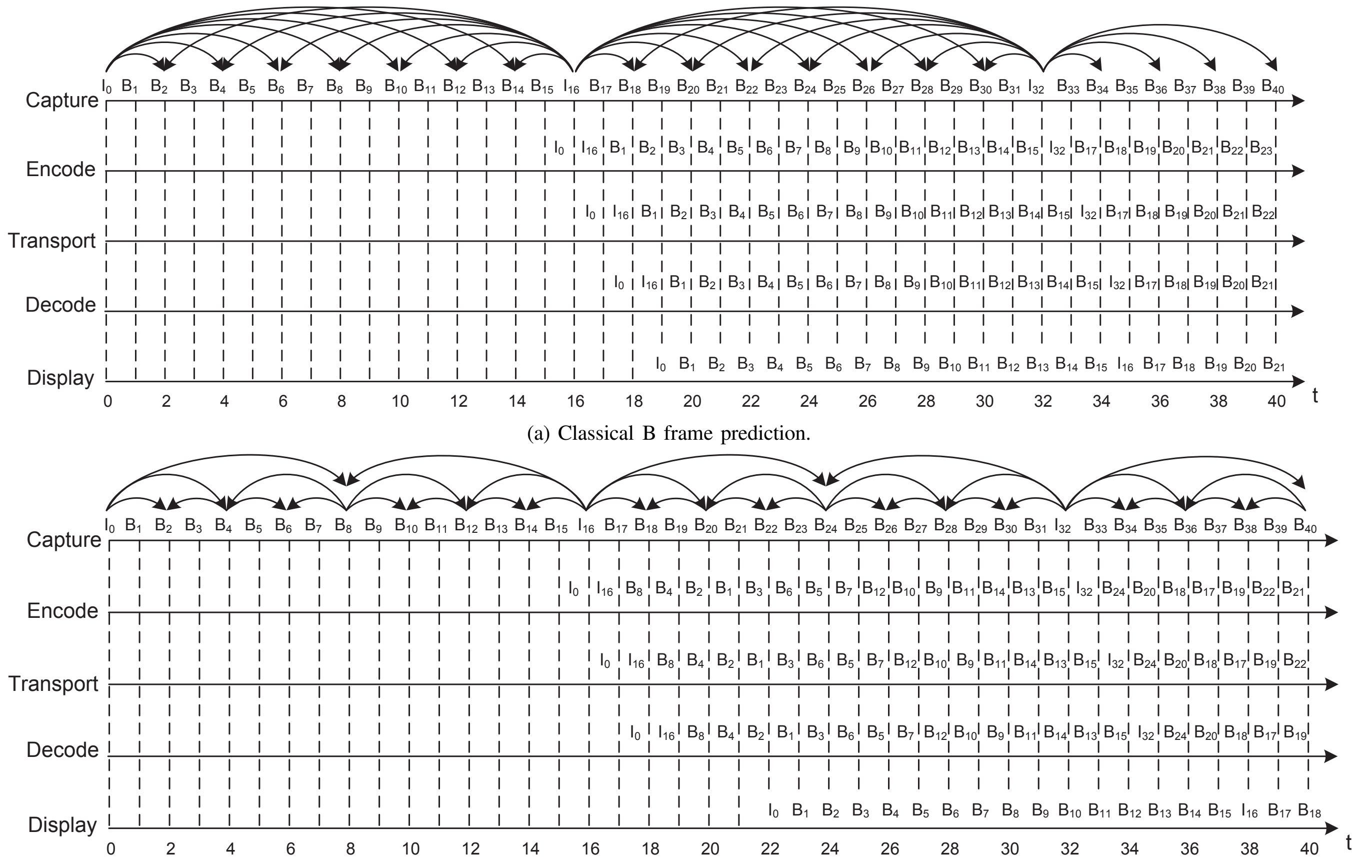
Examining closer the cause of these different delays for the case of live video streaming, we observe that the video frames are in display order on the capture and display axes, whereas, they are in encoding order on the encode, transmit, and decode axes. Specifically, for encoding a dependent frame, such as frame B_1, the encoder needs the encoded versions of the reference frames. With classical B frame prediction, frame B_1 only requires reference frames I_0 and I_{16}, whereas, hierarchical B frame prediction requires reference frames I_0, I_{16}, B_8, B_4, and B_2. Thus, an additional delay of three frame periods, i.e., 3Δ, is introduced by hierarchical B frame prediction. This additional delay becomes visible in Fig. 6 when comparing the frame sequences on the decode and display axes. In particular, in Fig. 6(b), the decoding of frame B_1 completes at time 23Δ; thus, the playback can start at the earliest at time 22Δ with frame I_0.

In the case of streaming prerecorded video, all encoded frames are available for immediate streaming (transport) upon a streaming request. Only the delays due to transport, decoding, and reordering (from encoding order to display order) are incurred. Specifically, in the classical B frame scenario illustrated in Fig. 6(a), the stream can effectively be requested at time instant 16Δ, i.e., the transport of the encoded frame I_0 can commence at 16Δ, and the playback can commence at time instant 19Δ, for a delay of 3Δ. We observe from Fig. 6(b) that the corresponding delay for the streaming of prerecorded video encoded with hierarchical B frames is 6Δ (effectively from time instant 16Δ to time instant 22Δ).

Generally, with β B frames between successive I and P frames, the live streaming delay is $(4 + \beta)\Delta$ for classical B frames and $(3 + \beta + \log_2(1 + \beta))\Delta$ for hierarchical B frames [78]. For the streaming of prerecorded video, the delays are 3Δ for classical B frames and $(2 + \log_2(\beta + 1))\Delta$ for hierarchical B frames. More detailed delay results, including delays for streaming systems with negligible computation time for encoding and decoding, are provided in [78]. It is important to keep in mind that this delay increase due to the hierarchical B frame dependencies is not the only additional delay encountered when streaming video encoded with H.264 SVC compared to video encoded with H.264/AVC or MPEG-4. In addition, H.264 SVC encoded video typically requires larger delays for smoothing out the higher frame size variability. Reaping the benefit of the smaller average bit rate of H.264 requires typically twice the smoothing delay compared to H.264/AVC [78].

2) Packetization: The H.264/AVC standard introduced the concept of network abstraction layer units (NALUs), which encapsulate the encoded video information. As noted in Section III, video traces typically characterize only the size of the encoded video data (including the NALU header).

In a practical streaming scenario, the video decoder requires typically first *sequence-level* header information that is con-


Capture
I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} I_{32} B_{33} B_{34} B_{35} B_{36} B_{37} B_{38} B_{39} B_{40}
Encode
I_0 I_{16} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{32} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23}
Transport
I_0 I_{16} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{32} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22}
Decode
I_0 I_{16} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{32} B_{17} B_{18} B_{19} B_{20} B_{21}
Display
I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16} B_{17} B_{18} B_{19} B_{20} B_{21}
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 t
Capture
I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} I_{32} B_{33} B_{34} B_{35} B_{36} B_{37} B_{38} B_{39} B_{40}
Encode
I_0 I_{16} B_8 B_4 B_2 B_1 B_3 B_6 B_5 B_7 B_{12} B_{10} B_9 B_{11} B_{14} B_{13} B_{15} I_{32} B_{24} B_{20} B_{18} B_{17} B_{19} B_{22} B_{21}
Transport
I_0 I_{16} B_8 B_4 B_2 B_1 B_3 B_6 B_5 B_7 B_{12} B_{10} B_9 B_{11} B_{14} B_{13} B_{15} I_{32} B_{24} B_{20} B_{18} B_{17} B_{19} B_{22}
Decode
I_0 I_{16} B_8 B_4 B_2 B_1 B_3 B_6 B_5 B_7 B_{12} B_{10} B_9 B_{11} B_{14} B_{13} B_{15} I_{32} B_{24} B_{20} B_{18} B_{17} B_{19}
Display
I_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} I_{16} B_{17} B_{18}
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 t

(a) Classical B frame prediction.

(b) Hierarchical B frame prediction, $\beta = 15$.

Fig. 6. Delay comparison of classical and hierarchical B frame prediction for live streaming with GoP structure G16B15. The time axis is in multiples of the frame period Δ.

veyed in parameter set NALUs [55]. Reasonable approximations for the sizes of these parameter set NALUs are 200 bytes for an H.264 SVC single layer encoding, 600 bytes for an H.264 SVC CGS encoding, and 900 bytes for an H.264 SVC MGS encoded video sequence. In addition, each frame is typically preceded by a prefix NALU ranging usually from 8 to 10 bytes. The prefix NALU informs the decoder about the frame dependencies and the scalability hierarchy, and is not reported in the video traces.

In many networking scenarios, the encoded video bitstream is transported using the Real Time Protocol (RTP) [81]. An RTP packet consists of the 12 byte RTP header, an 8 byte UDP header (20 bytes for TCP), and 20 byte IPv4 header (40 bytes for IPv6). The packaging of the NALUs into RTP packets is the subject of an Internet Draft [82] and follows largely the underlying principles from [83], [84]. An RTP packet may contain a single NALU, multiple NALUs (which may belong to the same frame or different frames), or a fragment of a NALU.

3) Example Evaluation: In this section, we provide an illustrative example for using a video trace for simulating a single video stream. In particular, we illustrate the packetization of the video traffic and the evaluation of the received video quality after lossy network transport.

a) Packetization: We consider the first 32 frames of the *Big Buck Bunny* video sequence, so as to illustrate the transmission of the sequence-level parameter set NALUs before the transmission of the first encoded video frame. We consider the *Big Buck Bunny* sequence in the CIF format encoded with H.264 SVC with hierarchical B frames into a single layer using the GoP pattern G16B15 and quantization parameter $q = 28$ for I frames (and 30 for B frames).

We focus in this illustrative example on using the video traces to simulate an individual stream. We consider transmission of the frames in encoding order. We suppose that the video stream is transmitted using RTP and IPv4; thus, the encapsulation overhead is 40 bytes per sent packet. We consider a network with a path maximum transfer unit (MTU) of 1500 bytes. The sender fragments NALUs larger than 1460 bytes to avoid fragmentation along the simulated transport path.

Generally, the sequence-level parameter set NALUs can be transmitted differently from the encoded video data NALUs, e.g., a higher transport quality of service can be used for these important signaling NALUs. We account for the parameter set NALUs with the 200 bytes in the first line of Table I. In our example, we aggregate for each frame the prefix NALU and the corresponding VCL NALU into one RTP packet (or several RTP packets when the MTU is exceeded). Table I illustrates the sizes of the encoded video frames (as obtained from the video trace) and the resulting sizes of the RTP packets (in bytes). We note that these first frames of the video include the opening credits which can be compressed very efficiently, resulting in small frame sizes (most B frames among the first

TABLE I
EXAMPLE SINGLE-LAYER H.264 SVC FRAME SIZE AND QUALITY TRACES AND RESULTING RTP PACKET SIZES

Enc.	Dis.	Size		PSNR-Y Quality			
ord.	ord.	Frame	Pkt.	Q_n^Y	$Q_n^Y(1)$	$Q_n^Y(2)$	$Q_n^y(3)$
	n	X_n					
		[byte]	[byte]	[dB]	[dB]	[dB]	[dB]
			200				
0	0	58	106	188.1	48.2	48.2	42.3
1	16	221	269	48.5	37.2	31.9	28.8
2	8	258	306	49.1	44.7	37.6	35.3
3	4	15	63	48.2	46.7	42.2	38.7
4	2	13	61	48.2	42.3	42.3	41.6
5	1	13	61	48.2	48.2	42.3	42.3
6	3	13	61	42.4	42.4	41.7	38.8
7	6	13	61	52.4	45.1	45.0	37.3
8	5	22	70	51.8	46.5	41.5	41.4
9	7	15	63	49.2	49.0	38.9	34.4
10	12	72	120	46.5	39.2	34.0	31.5
11	10	50	98	48.2	41.9	35.1	32.3
12	9	31	79	52.4	41.3	38.0	32.9
13	11	33	81	48.0	39.1	35.1	31.4
14	14	43	91	46.1	39.2	35.3	30.1
15	13	38	86	47.0	38.2	34.6	32.0
16	15	75	123	47.0	40.9	32.9	29.3
17	32	1335	1383	41.0	32.2	26.9	23.3
18	24	601	649	43.9	34.3	29.1	25.6
19	20	73	121	43.2	34.9	30.4	27.6
20	18	19	67	44.8	35.9	31.0	27.9
21	17	24	72	45.9	36.2	31.5	28.2
22	19	176	224	43.5	35.2	30.6	27.6
23	22	64	112	43.0	35.4	31.0	26.6
24	21	230	278	42.7	34.8	30.6	27.8
25	23	81	129	43.7	36.3	29.6	26.1
26	28	88	136	41.2	32.3	27.9	24.9
27	26	101	149	41.8	32.8	28.4	25.0
28	25	49	97	42.5	33.5	28.4	25.4
29	27	277	325	40.8	32.9	27.8	24.9
30	30	77	125	40.7	32.8	28.5	24.3
31	29	255	303	40.3	32.6	28.0	25.2
32	31	113	161	41.0	34.0	27.4	23.9

16 frames are smaller than 50 bytes) and relatively high PSNR qualities (above 46 dB).

b) Received Video Quality: Suppose that the RTP packet carrying frame number $n = 18$ in display order is not received. In order to determine the impact of the loss of frame $n = 18$ on the decodeability of other frames, we find the position of frame $n = 18$ in the B frame hierarchy in Fig. 1(b). We find that frame $n = 18$ falls into the second GoP of the video sequence, and, specifically, is in position 2 in the GoP display order. This frame in position 2 is needed for decoding frames 1 and 3 in the GoP (i.e., frames $n = 17$ and $n = 19$ in display order); no other frames in the GoP depend on the frame in position 2. For an elementary error concealment mechanism that redisplays the last successfully decoded frame for frames that are lost (or cannot be decoded), frame $n = 16$ is redisplayed for frames $n = 17,\ 18$, and 19. We obtain the PSNR values for these frames from the offset distortion trace for frame $n = 16$, reading off $Q_{16}^Y(1) = 37.2$ dB for frame $n = 17$, $Q_{16}^Y(2) = 31.9$ dB for frame $n = 18$, and $Q_{16}^Y(3) = 28.8$ dB for frame $n = 19$. We can read the PSNR values for the other frames directly from the video quality column in the video trace (which corresponds effectively to the offset $d = 0$), e.g., $Q_{16}^Y = 48.5$ dB for frame $n = 16$ and $Q_{20}^Y = 43.2$ dB for frame $n = 20$.

B. Layer-scalable H.264 Video

1) Temporal Layer Bit Rate and Quality Scaling: H.264 SVC encodings allow for temporal scalability, i.e., dynamically adapting the frame rate, by discarding or adding temporal layers $T = 1,\ 2, \ldots, \tau$, of hierarchical B frames. This dropping and adding of temporal layers allows for an initial scaling of the video bit rate and quality; additionally, the bit rate and quality can be adapted through dropping and adding of CGS layers and/or MGS layers. In order to match a prescribed target bitrate with temporal scalability, we can discard temporal layers, starting from the highest temporal layer $T = \tau$. In particular, a common approach is to convert a prescribed target bitrate into a byte budget for each GoP. Then, temporal layers are dropped within each individual GoP to meet the GoP byte budget.

The PSNR video quality after adaptation through dropping or adding temporal layers in conjunction with loss concealment through frame redisplay can be obtained from the offset distortion traces. For instance, dropping the highest temporal enhancement layer, layer $T = \tau = 4$, in the G16B15 GoP structure in Fig. 1(b) cuts the frame rate in half. The PSNR video qualities of the remaining displayed frames in layers $T = 0, 1, \ldots, \tau - 1$ can be read from the frame size and quality traces. For error concealment through frame redisplay, the PSNR qualities of the dropped frames can be read from the offset distortion traces of the frames in layers $T = 0, 1, \ldots, \tau - 1$; in particular, from the column for offset $d = 1$.

2) CGS Quality Scalability: In addition to temporal scalability, or instead of temporal scalability, the other layer scalability modes, namely CGS quality scalability or spatial scalability can be employed. We consider in the following CGS quality scalability. An H.264 SVC CGS encoding has the interframe dependencies illustrated in Fig. 1(b) in each layer, i.e., in the base layer and in each CGS enhancement layer. In addition, higher layers are encoded with reference to lower layers, e.g., the second CGS enhancement layer of frame B_8 depends on the first CGS enhancement layer and the base layer of frame B_8. Rate adaptation through dropping and adding CGS enhancement layers should consider these encoding dependencies to avoid transmitting CGS layers that cannot be decoded at the receiver due to missing encoding references, as elaborated in the following example adaptation strategies.

a) Rate/Quality Adaptation: A basic CGS quality scalability strategy is to add CGS enhancement layers (starting with the first CGS enhancement layer) to the base layer until a prescribed video bit rate is met. This adaptation of the number of CGS enhancement layers can be conducted on a range of time scales. For meeting a prescribed video bit rate over each GoP, for instance, the CGS enhancement layers are added to the base layers of all frames in a GoP until the GoP byte budget is exhausted. With this GoP time scale approach, the bit rate can be adjusted for each GoP according to the available network bandwidth.

A more complex adaptation strategy can combine varying the number of CGS enhancement layers with omitting the base layer (and all CGS enhancement layers) of some frames resulting effectively in combined temporal-CGS scalability. For instance for the G16B15 GoP pattern, illustrated in Fig. 1(b), with two CGS enhancement layers, the odd-indexed frames B_1, B_3, and so on, could be omitted while all three layers are transmitted for all other frames $\mathrm{I}_0,\ \mathrm{B}_2,\ldots,\mathrm{I}_{16}$. In this scenario, the PSNR quality of the odd-indexed B frames can be obtained from the offset distortion traces for the second enhancement layer of the even-indexed frames. For instance, the PSNR value for B_3 can be read from the offset distortion trace of the second enhancement layer of frame B_2 for offset $d = 1$. Continuing the example, suppose that for the even-indexed frames $\mathrm{B}_{18},\ \mathrm{B}_{20},\ldots,\mathrm{I}_{32}$, the base layer and only one enhancement layer are transmitted, while the odd-indexed frames $\mathrm{B}_{17},\ \mathrm{B}_{19},\ldots,\mathrm{B}_{31}$, are still omitted. Then, the PSNR quality of these odd-indexed frames can be obtained from the offset distortion traces of the first enhancement layer of the even-indexed frames.

Generally, rate adaptation mechanisms should transmit dependent frames with the same (or a smaller) number of layers as (than) the frames they depend on. For instance, if frame B_8 in Fig. 1(b) is sent with base layer and one enhancement layer (and frames I_0 and I_{16} are sent with two enhancement layers), then frame B_{12} should be send with only the base layer and one enhancement layer. Transmitting the second enhancement layer for frame B_{12} would be an inefficient use of bandwidth as frame B_{12} is missing the second enhancement layer forward reference from frame B_8.

b) Network Losses: As each H.264 SVC CGS layer has the dependency structure illustrated in Fig. 1(b), losses have similar effects as for single-layer streaming. If only higher CGS layers are lost for a frame, the PSNR can be read from the layer trace corresponding to the layers received for the frame. As an example, suppose that all frames were sent with the base layer and two enhancement layers. All three layers have been received for all frames, except that the second enhancement layer of frame B_{14} was lost. The received frame PSNR qualities can then be obtained from the trace for the second enhancement layer for frames $\mathrm{I}_0,\ \mathrm{B}_1,\ \ldots,\ \mathrm{B}_{12}$ and frame I_{16}. Frame B_{13} misses the backward reference from frame B_{14} in the second enhancement layer. Depending on the outcome of the motion-compensated prediction, the encoder may or may not have used this backward reference to encode the second enhancement layer of frame B_{13}. However, generally both forward and backward prediction are needed for correct decoding and we can therefore conservatively approximate the quality of frame B_{13} (and similarly frame B_{15}) from the first enhancement layer trace. On the other hand, if a frame is completely lost, the frame PSNR for re-display of a preceding frame can be read from the preceding frame's offset distortion trace for the appropriate layer, as illustrated in the example in Section V-B2a.

As noted in Section IV-B, H.264 SVC CGS quality scalability incurs high bit rate overheads when the number of CGS enhancement layers grows larger than about two or three. Low bit rate overheads in the 10–30 % range have only been observed with two CGS enhancement layers that permit the switching between three levels (base layer and base layer with one or two enhancement layers) of vastly different bit rates or video qualities, e.g., for three distinct classes of video service. Finer grained adaptations of the video bit rate or quality, e.g., in response to small to moderate temporary changes in network bandwidth, require typically H.264 SVC MGS encodings.

C. Sublayer Scalable H.264 SVC MGS Encoded Video

In this section, we explain how to simulate the transmission of H.264 SVC MGS encoded video. We explain how the different types of MGS traces introduced in Section III-C can be used to simulate rate adaptations and to determine the PSNR video quality at the receiver. We note that throughout, the base layer and each MGS layer is independently packetized, similarly to the case of layer-scalable encoded video.

1) MGS Layer Traces: Initially, we consider the simulation of the streaming with rate adaptation through dropping or adding complete MGS layers using the MGS layer traces. The MGS layer trace for a given MGS layer $m,\ m = 0, 1, \ldots, M$ (whereby $m = 0$ corresponds to the base layer) gives the PSNR frame quality for the scenario where for all frames the base layer and the MGS layers up to and including MGS layer m are received. Packet losses or rate adaptations through dropping and adding MGS layers lead to varying numbers for received MGS layers for the frames. H.264 SVC MGS uses the highest available quality of the reference frames for encoding dependent frames, i.e., the lower layers of a given frame are encoded with reference to the highest available layers of the reference frames. As a result, variations in the numbers of MGS layers in a given frame have implications for the PSNR qualities of dependent frames that can only be approximated with MGS layer traces.

For instance, consider the MGS layer trace excerpts in Table II for the encoding with the G16B15 GoP structure (illustrated in Fig. 1(b)) with $M = 6$ MGS layers. Suppose the base layer and all $M = 6$ MGS layers have been received for all frames, except frame $n = 1297$ for which only the base layer has been received. Frame $n = 1297$ corresponds to frame B_1 in the GoP structure in Fig. 1(b), i.e., it is not referenced in the encoding of any other frame in the GoP. Thus, the PSNR qualities of all other frames can be read from the MGS layer trace for $m = M = 6$. The base layer trace gives an approximation, more specifically a lower bound, of the displayed PSNR quality of $Q^{0,Y}_{1297} = 31.0$ dB for frame $n = 1297$ (B_1 in the GoP structure). To see this, note that the base layer trace considers the scenario where only the base layer is received for frame B_1 and all its reference frames, including frames I_0, B_2, B_4, and so on. Since these frames have been received with more layers, i.e., in higher quality, the actual decoded PSNR quality of frame B_1 is likely higher. Characterizing the exact PSNR frame qualities for all combinations of such cases is prohibitively complex. The priority level and MGS-temporal layer traces characterize the PSNR values for some of these combinations.

Next, suppose again that all frames are completely received but only the MGS layers up to $M = 5$ are received for frame $n = 1298$ (corresponding to frame B_2 in the GoP structure in Fig. 1(b)). Frames B_1 and B_3 are predicted from frame B_2. The PSNR values for frames B_1 and B_3 in MGS layer trace $m = 5$ consider the case where all frames, including all reference frames have the base layer and $m = 5$ MGS layers

TABLE II

EXAMPLE EXCERPTS FROM MGS LAYER TRACES FOR H.264 SVC MGS ENCODING OF *Big Buck Bunny* WITH GOP STRUCTURE G16B15 WITH ONE BASE LAYER AND ONE ENHANCEMENT LAYER WITH MGS WEIGHT VECTOR $\mathbf{W} = [1, 2, 2, 3, 4, 4]$. FOR EACH FRAME n IN DISPLAY ORDER AND MGS LAYER m, THE TRACES GIVE THE SIZE X_n^m, THE LUMINANCE PSNR FRAME QUALITY $Q_n^{m,Y}$, AND THE PRIORITY LEVEL P_n^m.

		Base Layer			MGS Layers																	
		$m=0$			$m=1$			$m=2$			$m=3$			$m=4$			$m=5$			$m=6$		
n		X_n^0	$Q_n^{0,Y}$	P_n^0	X_n^1	$Q_n^{1,Y}$	P_n^1	X_n^2	$Q_n^{2,Y}$	P_n^2	X_n^3	$Q_n^{3,Y}$	P_n^3	X_n^4	$Q_n^{4,Y}$	P_n^4	X_n^5	$Q_n^{5,Y}$	P_n^5	X_n^6	$Q_n^{6,Y}$	P_n^6
		[byte]	[dB]		[byte]	[dB]		[byte]	[dB]		[byte]	[dB]		[byte]	[dB]		[byte]	[dB]		[byte]	[dB]	
1296	I	5889	30.9	63	2131	31.4	42	3438	32.5	41	2570	33.4	40	3576	35.3	39	2653	36.8	38	1990	38.4	37
1297	B	249	31.0	63	538	31.9	10	155	32.9	9	118	33.8	8	157	35.7	7	98	37.3	6	101	39.0	5
1298	B	420	31.0	63	639	31.9	20	247	32.9	19	153	33.9	15	191	35.8	14	115	37.4	13	104	39.0	12
1299	B	258	31.1	63	429	32.0	5	168	33.0	4	104	34.0	3	140	35.9	2	86	37.5	1	86	39.1	0
1300	B	533	31.1	63	819	32.0	26	362	33.0	20	206	34.0	19	238	35.9	18	137	37.5	17	105	39.1	16
1301	B	253	31.1	63	502	32.1	10	192	33.2	9	116	34.1	8	141	36.1	7	93	37.6	6	80	39.2	5
1302	B	397	31.2	63	664	32.2	20	260	33.2	19	142	34.2	15	185	36.1	14	105	37.6	13	100	39.2	12
1303	B	239	31.2	63	402	32.2	5	140	33.2	4	105	34.2	3	138	36.1	2	90	37.6	1	84	39.2	0
1304	B	897	31.3	63	1026	32.1	32	556	33.2	26	301	34.2	25	303	36.0	24	211	37.6	23	146	39.1	22
1305	B	233	31.2	63	499	32.2	5	164	33.3	4	120	34.3	3	164	36.2	2	89	37.7	1	104	39.3	0
1306	B	391	31.2	63	730	32.2	20	271	33.3	19	161	34.3	18	209	36.2	17	121	37.7	16	112	39.2	15
1307	B	293	31.2	63	543	32.3	10	198	33.3	9	121	34.3	8	154	36.1	7	93	37.7	6	91	39.3	5
1308	B	686	31.2	63	1004	32.2	26	395	33.2	21	217	34.2	20	255	36.0	19	151	37.6	18	126	39.1	17
1309	B	225	31.2	63	471	32.2	10	169	33.3	9	112	34.2	8	148	36.1	7	82	37.7	6	84	39.2	5
1310	B	461	31.2	63	776	32.2	20	250	33.2	19	161	34.2	18	207	36.1	17	133	37.6	16	116	39.2	15
1311	B	272	31.3	63	602	32.2	10	193	33.3	9	130	34.2	8	183	36.0	7	114	37.6	6	115	39.2	5
1312	I	5391	31.3	63	2109	31.8	42	3268	32.9	41	2406	33.9	40	3238	35.6	39	2475	37.2	38	1765	38.7	37
1313	B	239	31.4	63	558	32.2	10	171	33.2	9	118	34.2	8	166	36.0	7	97	37.6	6	96	39.1	5
1314	B	492	31.4	63	732	32.3	20	275	33.3	19	152	34.3	15	209	36.1	14	126	37.7	13	106	39.1	12
1315	B	295	31.4	63	475	32.3	5	216	33.4	4	125	34.4	3	189	36.2	2	105	37.8	1	102	39.2	0
1316	B	692	31.4	63	958	32.3	26	419	33.4	21	236	34.4	20	280	36.2	19	176	37.7	18	130	39.1	17
1317	B	284	31.4	63	499	32.5	10	191	33.6	9	135	34.6	8	177	36.4	7	107	37.9	6	97	39.3	5
1318	B	458	31.5	63	689	32.6	20	285	33.7	19	164	34.7	15	223	36.5	14	131	38.0	13	109	39.4	12
1319	B	338	31.6	63	545	32.7	10	216	33.8	9	152	34.8	8	179	36.6	7	130	38.1	6	103	39.5	5
1320	B	1028	31.7	63	1069	32.7	37	648	33.8	26	351	34.8	25	375	36.7	24	238	38.1	23	174	39.4	22
1321	B	342	31.6	63	614	32.7	10	230	33.8	9	142	34.8	8	176	36.7	7	126	38.1	6	100	39.4	5
1322	B	531	31.5	63	764	32.7	20	350	33.7	19	182	34.7	15	251	36.6	14	139	38.1	13	107	39.4	12
1323	B	358	31.5	63	601	32.6	10	219	33.7	9	146	34.7	8	174	36.6	7	122	38.0	6	98	39.4	5
1324	B	729	31.4	63	998	32.4	26	457	33.5	21	264	34.6	20	284	36.4	19	175	37.9	18	121	39.2	17
1325	B	346	31.4	63	586	32.4	10	249	33.5	9	154	34.6	8	165	36.4	7	114	37.9	6	93	39.3	5
1326	B	557	31.5	63	824	32.4	20	350	33.5	19	202	34.5	15	232	36.2	14	144	37.8	13	105	39.2	12
1327	B	387	31.5	63	635	32.4	10	251	33.4	9	166	34.4	8	198	36.2	7	110	37.8	6	89	39.2	5
1328	I	5110	31.5	63	2070	32.0	42	3185	33.1	41	2298	34.0	40	3149	35.8	39	2396	37.4	38	1686	38.8	37

available for decoding. However, for frames B_1 and B_3, the reference frames I_0 and B_4 (and the other reference frames B_8 and I_{16}) are available with all $m = M = 6$ MGS layers, whereas reference frame B_2 is available with only $m = 5$ MGS layers. One possible approximation is to use the PSNR quality from the $m = 5$ MGS layer trace for frames B_1 ($n = 1297$) and B_3 ($n = 1299$), i.e., $Q_{1297}^{5,Y} = 37.3$ dB and $Q_{1299}^{5,Y} = 37.5$ dB (as well as frame B_2, i.e., $Q_{1298}^{5,Y} = 37.4$ dB). Another possible approximation is to average the PSNR values from the $m = 5$ and $m = M = 6$ MGS layer traces for frames B_1 and B_3.

Generally, when evaluating video quality it is important to keep in mind that the PSNR provides a moderately accurate measure of the subjective (viewer perceived) video quality [65]–[68]. Nevertheless, the PSNR is widely used as it has low computational complexity and can thus be provided for long video traces, which are required for evaluating network transport mechanisms with high levels of statistical confidence. When applying approximations to determine the PSNR values from video traces in an evaluation of network transport mechanisms, the same approximation should be consistently applied to all mechanisms considered in a comparative study.

The effects of dropping layers to adapt to lower available network bandwidth can similarly be approximated. For instance, the base layer and all $m = 6$ MGS layers may be streamed up to and including the I frame $n = 1312$ in Table II, while only the base layer and the first $m = 5$ MGS layers are streamed for subsequent frames. Then, the PSNR qualities of frames $n = 1313$ and onward can be approximated by the PSNR values in the $m = 5$ MGS layer trace (possibly averaged with the PSNR values in the $m = 6$ MGS layer trace).

As observed in Section IV-C, adapting bit rates by dropping MGS layers for all frames irrespective of their position in the hierarchical B frame structure gives low RD efficiency. This is mainly because dropping the MGS layers uniformly from all frames counteracts the key frame encoding mechanism of H.264 SVC MGS that uses higher layers of some frames as encoding references for the lower layers of other frames. We proceed therefore to simulating the streaming of MGS encoded video using the priority levels traces and the MGS-temporal layer traces in the next subsections. Nevertheless, the complex frame and MGS layer dependencies illustrated in this section need to be carefully considered when determining the PSNR frame qualities at the receiver.

2) Target Bit Rate Traces: The target bit rate traces can be used to simulate the transmission of a video stream that meets a prescribed average bit rate over the duration of the full encoded video sequence. The frame sizes in the target bit rate traces do not include the packetization overheads, nor additional NALUs, such as prefix NALUs. If a simulation scenario requires packetized frames, the packetization overhead needs to be added to the frame sizes. Furthermore, if a simulation scenario requires that the individual MGS layers are packetized and streamed separately, the number of extracted MGS layers m for each video frames needs to be read from the target bit rate trace. Subsequently, the sizes of the individual m MGS layers (without packetization overhead) can be read from the individual MGS layer traces. When simulating with the target bit rate traces it is important to keep in mind that the underlying RD optimization of the bit stream extraction over the complete video stream results typically in significant short-term variations of the bit rate around the target bit rate.

3) Priority Level Traces: Alternatively, the MGS stream transmission can be directly simulated with the priority level traces which give the aggregate frame size X_n, PSNR video quality Q_n, and number of MGS layers m for a given priority level $P,\ P = 63, 62, \ldots, 1, 0$ (whereby $P = 63$ indicates the highest priority). The priority level P can be used in a wide variety of ways to prioritize the transmission of the video packets or to adapt the video bit rate. For instance, the high priority video data with priority levels in the range 63 to P could be transmitted with a higher priority class in a differentiated services network, while video data with priority levels $P - 1$ to 0 is transmitted with lower priority.

Excerpts of the priority level traces for $P = 30$ and $P = 20$ for *Big Buck Bunny* are given in Table III. Notice that the frame sizes in Table III are obtained by summing the sizes of the layers with a priority level of P or higher, that is, X_n in Table III is obtained as $X_n = \sum_{m: P_n^m \geq P} X_n^m$ from Table II. However, the frame PSNR qualities Q_n^Y in Table III cannot be obtained from Table II. Instead, the frame PSNR qualities in priority level traces were obtained by carrying out the actual bitstream extraction according to the priority levels followed by decoding and comparison with the original (unencoded) frames.

a) Rate/Quality Adaptation: For an example of the rate adaptation, consider again Table III. Suppose that up to and including frame $n = 1312$ there is sufficient bandwidth for transmitting all MGS layers with a priority level of $P \geq 20$, but then a reduction in bandwidth allows transmission of only MGS layers with priority levels $P \geq 30$. We observe for the example in Table III that increasing the priority level threshold from 20 to 30 results in the dropping of all MGS layers from the frames corresponding to B frames $\mathrm{B}_2,\ \mathrm{B}_4, \mathrm{B}_6,\ \mathrm{B}_{10}$, and so on in the hierarchical B frame prediction structure in Fig. 1(b), while the number of MGS layers is reduced from $m = 6$ to $m = 1$ for the "middle" B frame B_8. The I frames in the temporal base layer retain all $m = 6$ MGS layers. Notice that even though no MGS layers are dropped from the odd-indexed B frames, their PSNR qualities are still reduced since only lower quality reference frames are available after the switch to the higher priority level threshold $P = 30$. For instance, the PSNR quality of frame $n = 1313$ would be 38.2 dB if all reference frames were received according to the $P = 20$ threshold, but is reduced to 37.8 dB with the $P = 30$ threshold.

TABLE III
EXCERPTS FROM PRIORITY LEVEL TRACES FOR H.264 SVC MGS ENCODING OF *Big Buck Bunny* WITH $M = 6$ MGS LAYERS FOR PRIORITY LEVEL $P = 30$, WHICH INCLUDES MGS LAYERS WITH PRIORITY LEVELS OF 30 AND HIGHER, AND PRIORITY LEVEL $P = 20$. THE TRACES GIVES FOR EACH FRAME n, THE FRAME TYPE, THE AGGREGATE FRAME SIZE X_n, THE LUMINANCE PSNR QUALITY Q_n^Y, AND THE NUMBER OF INCLUDED MGS LAYERS m ($m = 0$ INDICATES THAT ONLY THE BASE LAYER IS INCLUDED).

		$P = 30$			$P = 20$		
n		X_n	Q_n^Y	m	X_n	Q_n^Y	m
		[byte]	[dB]		[byte]	[dB]	
1296	I	22247	38.4	6	22247	38.4	6
1297	B	249	37.6	0	249	38.1	0
1298	B	420	37.4	0	1059	38.4	1
1299	B	258	37.1	0	258	38.1	0
1300	B	533	37.1	0	1714	38.6	2
1301	B	253	36.9	0	253	38.1	0
1302	B	397	37.0	0	1061	38.5	1
1303	B	239	37.3	0	239	38.4	0
1304	B	1923	37.7	1	3440	39.1	6
1305	B	233	37.1	0	233	38.4	0
1306	B	391	36.7	0	1121	38.4	1
1307	B	293	36.6	0	293	38.0	0
1308	B	686	36.8	0	2302	38.7	3
1309	B	225	36.8	0	225	38.1	0
1310	B	461	37.0	0	1237	38.4	1
1311	B	272	37.6	0	272	38.1	0
1312	I	20652	38.7	6	20652	38.7	6
1313	B	239	37.8	0	239	38.2	0
1314	B	492	37.3	0	1224	38.4	1
1315	B	295	36.9	0	295	38.1	0
1316	B	692	36.9	0	2305	38.7	3
1317	B	284	36.7	0	284	38.1	0
1318	B	458	36.9	0	1147	38.4	1
1319	B	338	37.0	0	338	38.1	0
1320	B	2097	37.7	1	3883	39.4	6
1321	B	342	37.0	0	342	38.1	0
1322	B	531	36.7	0	1295	38.3	1
1323	B	358	36.7	0	358	37.9	0
1324	B	729	36.7	0	2448	38.7	3
1325	B	346	36.8	0	346	37.9	0
1326	B	557	37.1	0	1381	38.4	1
1327	B	387	37.5	0	387	38.0	0
1328	I	19894	38.8	6	19894	38.8	6

Generally, we give the following two recommendations for using the priority level traces in simulations with bit rate or quality adaptations:

(R1) All M MGS layers should be included for all I frames, which translates for a given MGS layer trace into a maximum priority level $P_{\max} = \min_{\{n:\ n=kg\}} P_n^M$ for $k = 0, 1, \ldots$, i.e., the minimum is taken over all I frames. For the example trace in Table II, setting the priority level to $P_{\max} = 37$ or lower ensures that all $M =$ MGS layers are included for all I frames; whereas, a priority level higher than 37 would exclude the highest-indexed MGS layers.

(R2) Rate or quality adaptations through adapting the priority level should only be made on a GoP by GoP basis, i.e., at I frames.

When both of these guidelines are followed, the streamed frame sizes and the received PSNR frame qualities (without losses in the network) can be read from the priority level traces.

We proceed to justify these two recommendations by considering the cases when one or two of the recommendations are violated. First, when a trace with a priority level P higher than $P_{\max}$ is used for network simulations, then some I frames will miss their highest MGS layers. As long as the same priority level trace P is used for the entire stream duration (and no network losses occur), then the received PSNR frame qualities can be read from the priority level trace P. However, switching during the streaming simulation from (or to) priority level trace $P_1,\ P_1 > P_{\max}$, to (or from) another priority level trace P_2 (whereby P_2 is arbitrary) may require approximations of the frame PSNR. For instance, suppose that priority level trace P_1 is used up to and including an I frame n, for which trace P_1 includes $m_1,\ m_1 < M$, MGS layers. Suppose that trace P_2, which includes $m_2,\ m_2 \neq m_1$, MGS layers for I frame n, is used for frame $n+1$ and onward. Then, the actual PSNR values for frames $n+1, \ldots, n+g-1$ are not contained in trace P_2 since trace P_2 is for m_2 MGS layers in I frame n. However, I frame n, which serves as a reference for frames $n+1, \ldots, n+g-1$, has $m_1,\ m_1 \neq m_2$, MGS layers in I frame n in the actual simulated stream. Thus, the PSNR values for frames $n+1, \ldots, n+g-1$ would need to be approximated, e.g., by averaging the PSNR values from traces P_1 and P_2.

Next consider switching priority level traces inside a GoP. For instance, suppose that priority level trace $P = 20$ in Table III is used up to and including frame $n = 1304$ and trace $P = 30$ is used for frame $n = 1305$ and onwards. Then, the actual PSNR values for frames $n = 1305$ through 1311 are not in these traces and would need to be approximated, or, in this particular case, could be obtained from MGS-temporal layer trace $\mu = 2M = 12$.

In summary, by following the two recommendations (R1) and (R2), the actual PSNR values can be read from the traces and approximations be avoided. The two recommendations (R1) and (R2) are in accordance with a wide range of practical streaming scenarios as omitting MGS layers from I frames results in PSNR values that are too low for many streaming applications and adapting bit rates at most once per GoP provides sufficient flexibility for many networking protocols.

b) Network Losses: The evaluation of the impact of the loss of some MGS layers or even the base layer of a frame n during network transport depends critically on the position of the frame in the prediction structure in Fig. 3. If frame n has no dependent frames, then there is no error propagation to dependent frames and only frame n is affected. If frame n has dependent frames, then the PSNR quality of frame n and all its dependent frames are affected. The PSNR quality of an affected frame can be obtained (or conservatively approximated) from a priority level trace or MGS-temporal layer trace with the same (or a smaller) number of MGS layers for frame n as frame n has after the loss in the simulation, and the same (or a smaller) number of MGS layers for the reference frames of frame n.

For instance, suppose in a simulated transmission of priority level trace $P = 20$ in Table III the MGS layer of frame $n = 1298$ is lost. Then, the PSNR qualities of the affected frames 1297–1299 can be conservatively approximated from the priority level trace $P = 30$ in Table III. This approximation would be quite conservative since the reference frames 1300 and 1304 have significantly fewer MGS layers in the $P = 30$ trace compared to the received video in the simulation. A closer approximation is provided by the $\mu = 14$ MGS-temporal layer trace in Table IV, which has the same number of MGS layers for reference frames 1296, 1312, 1304, and 1300, and the same number of MGS layers for frame 1298 as frame 1298 has after the simulated network loss. From Table IV, we obtain the PSNR qualities of frames 1297, 1298, and 1299 after the loss as 37.8 dB, 37.9 dB, and 37.8 dB.

TABLE IV
EXCERPTS FROM MGS-TEMPORAL LAYER TRACES OF *Big Buck Bunny* WITH $M = 6$ MGS LAYERS FOR DIFFERENT MGS-TEMPORAL EXTRACTION THRESHOLDS $\mu = T_\mu M + m_\mu$.

			$\mu = 14$		$\mu = 16$		$\mu = 20$	
n		T	X_n [byte]	Q_n^Y [dB]	X_n [byte]	Q_n^Y [dB]	X_n [byte]	Q_n^Y [dB]
1296	I	0	22247	38.4	22247	38.4	22247	38.4
1297	B	4	249	37.8	249	37.8	249	38.2
1298	B	3	420	37.9	420	37.9	1306	38.7
1299	B	4	258	37.8	258	37.9	258	38.3
1300	B	2	1714	38.6	2158	39.0	2400	39.1
1301	B	4	253	37.8	253	37.9	253	38.3
1302	B	3	397	37.9	397	37.9	1321	38.9
1303	B	4	239	38.1	239	38.1	239	38.5
1304	B	1	3440	39.1	3440	39.1	3440	39.1
1305	B	4	233	38.0	233	38.0	233	38.4
1306	B	3	391	37.5	391	37.6	1392	38.8
1307	B	4	293	37.6	293	37.8	293	38.3
1308	B	2	2085	38.5	2557	38.9	2834	39.1
1309	B	4	225	37.7	225	37.9	225	38.3
1310	B	3	461	37.6	461	37.7	1487	38.7
1311	B	4	272	37.8	272	37.8	272	38.2
1312	I	0	20652	38.7	20652	38.7	20652	38.7
1313	B	4	239	38.0	239	38.0	239	38.3
1314	B	3	492	37.8	492	37.8	1499	38.7
1315	B	4	295	37.7	295	37.9	295	38.3
1316	B	2	2069	38.5	2585	39.0	2891	39.1
1317	B	4	284	37.8	284	37.9	284	38.3
1318	B	3	458	37.8	458	37.8	1432	38.9
1319	B	4	338	37.8	338	37.8	338	38.2
1320	B	1	3883	39.4	3883	39.4	3883	39.4
1321	B	4	342	37.8	342	37.8	342	38.2
1322	B	3	531	37.5	531	37.6	1645	38.8
1323	B	4	358	37.6	358	37.7	358	38.1
1324	B	2	2184	38.4	2732	39.0	3028	39.2
1325	B	4	346	37.5	346	37.6	346	38.1
1326	B	3	557	37.6	557	37.6	1731	38.8
1327	B	4	387	37.7	387	37.7	387	38.1
1328	I	0	19894	38.8	19894	38.8	19894	38.8

4) MGS-Temporal Layer Traces: While the priority level based extraction provides the RD optimal selection of MGS layers to transmit for each frame, the RD optimization is computationally demanding. Application scenarios with limited computing resources or tight time constraints may not allow for compute intensive RD optimal extraction of MGS layers. For such networking scenarios where priority level information is not available, transmitting MGS layers according to MGS-temporal layer extraction is a low-complexity alternative that approximates the priority level based extraction.

Table IV illustrates the MGS-temporal layer traces for three adaptation scenarios. In the left example with $\mu = 14$, all $M = 6$ MGS layers are included for frames in temporal layers up to $T_\mu - 1 = \lfloor \frac{\mu}{M} \rfloor - 1 = 1$, i.e., for temporal layers 0, and 1, while $m_\mu = \mu - T_\mu M = 2$ MGS layers are included for frames in temporal layer $T_\mu = 2$. Only the base layer ($m = 0$) is included for frames in temporal layers $T_\mu + 1 = 3$ and 4. Notice that similar to the priority level traces, the frame sizes in the MGS-temporal layer traces can be obtained by adding the frame sizes of the corresponding MGS layer traces, e.g., for frame $n = 1296$, $\sum_{m=0}^{6} X_{1296}^{m}$ from Table II is equal to $X_{1296} = 22247$ bytes in Table IV. However, the PSNR quality in the MGS-temporal layer traces in Table IV cannot be obtained from the MGS layer traces in Table II; rather, actual encoding, bitstream extraction, and decoding operations of the H.264 SVC codec need to be conducted to obtain the PSNR values in the MGS-temporal layer traces

For simulating rate/quality adaptations and determining the impact of network losses, generally the same considerations as explained for priority level traces in Section V-C3 apply. In particular, recommendation (R1) still holds since switching to (or from) an MGS-temporal layer trace $\mu,\ \mu < M$, to another MGS-temporal layer trace would require approximations of the PSNR values of the frames between the last I frame from the old trace and the first I frame from the new trace (as at least one of these I frames would have less than M MGS layers).

The MGS-temporal layer traces indexed with the extraction threshold $\mu,\ \mu = 0, 1, \ldots, (\tau + 1)M$, add MGS layers to the frames strictly in the order of the frames in the B frame hierarchy, i.e., MGS layers are only added for frames in a temporal layer after all M MGS layers have been added for all frames in the lower temporal layers. This strict order is in contrast to the priority level approach where some MGS layers may be added for frames in a higher temporal layer even though frames in lower temporal layers have less than M MGS layers. Due to this strict ordering of adding MGS layers to temporal layers in the MGS-temporal layer traces, recommendation (R2) can be relaxed as follows for simulations with MGS-temporal layer traces. For downswitching from an MGS layer trace μ_h to a trace $\mu_l,\ \mu_l < \mu_h$, at any frame in the encoder transmission order, the PSNR values can be obtained from the MGS layer traces $\mu = \mu_l, \ldots, \mu_h$, by matching the number of MGS layers of the frames in the simulated stream to these traces. For instance, consider transmitting according to trace $\mu_h = 20$ for every frame that is transmitted in encoding order up to and including frame $n = 1300$ in Table IV, and then using trace $\mu_l = 14$. Then, the PSNR quality of frames up to and including frame 1300 can be read from MGS layer trace $\mu_h = 20$. Frames 1301–1303 have no MGS layers and depend on frames 1300 and 1304 (as well as frames 1296 and 1312), which have $M = 6$ MGS layers; thus, the PSNR qualities of frames 1301–1303 can be read from the $\mu = 18$ trace. The PSNR qualities for frames 1304 and onwards can be read from the $\mu_l = 14$ trace.

For upswitching from an MGS-temporal layer trace μ_l to a trace $\mu_h,\ \mu_h > \mu_l$, the PSNR values can be obtained from traces $\mu = \mu_l,\ \mu_h$ when the switching occurs at a frame that has M MGS layers in both traces. For instance, consider transmitting according to trace $\mu = 14$ up to and including frame $n = 1304$ and then using trace $\mu = 20$. Then, the PSNR qualities of frames up to 1304 can be read from the $\mu = 14$ trace, while the PSNR values of the following frames can be read from the $\mu = 20$ trace.

The strict order structure of the MGS-temporal layer traces allows for the examination of a wider range of adaptations than permitted by recommendations (R1) and (R2) without requiring approximations. However, the recommendations (R1) and (R2) are still good starting points for basic simulation evaluations.

VI. Conclusion

A. Tutorial Summary

This tutorial has provided a comprehensive methodology for evaluating the network transport of H.264 encoded video with H.264 video traces. This tutorial considered video encoded into a single-layer (non-scalable), encoded video that is scalable in the temporal, spatial, or quality (SNR) dimensions at the granularity of complete layers, and encoded video that is quality scalable at the sublayer granularity. We first reviewed the main H.264 coding techniques to provide background for communications and networking generalists to understand the implications of H.264 video encoding for video network transport. We introduced trace structures that characterize the frame size (in byte) and quality (in PSNR or other objective metrics) of the different types of H.264 encoded video. We then summarized the main results of extensive traffic studies of H.264 encoded video. One main result was that H.264 SVC single-layer encoding with hierarchical B frames achieves significantly higher rate-distortion (RD) performance (i.e., significantly smaller bit rates for same video quality) than H.264/AVC and MPEG-4, while resulting in significantly higher traffic variability. Another main result was that H.264 SVC Medium Grain Scalability (MGS), which provides quality scalability at the sublayer level, allows for highly flexible video bitrate and quality adaptation while achieving RD efficiency close to H.264 SVC single-layer encoding.

We provided extensive guidelines for simulation studies examining H.264 video network transport. We explained how to account for the hierarchical B frame dependencies in the timing of frame transmissions and playout. We also explained how to generate video packet flows from the video traces. For the different H.264 scalability modes, we explained how video bit rate and quality adaptations can be simulated with H.264 video traces.

B. Future Video Communications and Networking Research Directions

We conclude this tutorial with an outlook on future directions for the trace characterization of encoded video and, generally, on future directions for video communications and networking research. An important emerging direction for the evaluation of video transport over networks is three-dimensional (3D) video. 3D video typically involves the transmission of two views, i.e., a left view and a right view, for each video frame. Multi-view coding (MVC), i.e., the RD efficient encoding of the left and right views, has attracted significant interest in the video compression research community [85], [86]. As MVC techniques mature, the design and evaluation of network transport mechanisms for encoded 3D video will likely become an important research area [87]. Trace characterizations of encoded 3D video will be important for facilitating research and evaluations of network transport of 3D encoded video and are currently in preparation.

We foresee a number of important emerging areas for research on video communications and networking. We believe that H.264 video traces and the tutorial instructions on evaluations with H.264 video trace will facilitate the exploration and performance evaluation of video transport mechanisms in these emerging research areas. First, we note that a large portion of the early video transport research has focused on the efficient transport of a single video stream, e.g., through the smoothing of a single VBR video stream [88]–[91], by a single dedicated set of network nodes. More recently, collaborative approaches have attracted significant attention. For instance, collaborative smoothing approaches jointly smooth several ongoing video streams; thus, achieving improved statistical multiplexing gains [41], [92], [93]. Similarly, with cooperative video streaming, several nodes cooperate to improve the transmission of a video stream [94]–[98]. Combining these two strategies, i.e., collaborative transmission of several video streams while exploiting the cooperative effects of several nodes may give substantial efficiency gains.

We also foresee significant future research efforts on optimizing video transport in specific network types, such as wireless networks [99]–[101]. The inherent constraints and properties of specific network types, such as wireless channel characteristics or cognitive radio resource management, require transport mechanisms that efficiently accommodate both the characteristics of the encoded video as well as the transport services provided by the network type.

Furthermore, the integrated internetworking of different types of networks, e.g., in Fiber-Wireless (FiWi) networks that combine an optical (fiber) access network with a wireless access or local area network poses unique challenges for the efficient transport of encoded video [102]–[107]. Future integrated internetworking efforts will likely expand the reach of FiWi networks to encompass optical networks that span metropolitan areas [108]–[110]. Developing and evaluating integrated networking mechanisms, such as medium access control protocols and scheduling mechanisms for this increasing number of integrated types of networks will pose challenging research issues with the prospect of high efficiency increases.

Finally, we believe that throughout the research efforts on video streaming, energy-efficiency will become an increasingly important goal and will require extensive research efforts [111], [112].

ACKNOWLEDGMENTS

We are indebted to Dr. Geert Van der Auwera, formerly with Arizona State University, Tempe, and now with Qualcomm Inc. and Prof. Lina Karam of Arizona State University for sharing deep insights and many helpful pointers in the area of video signal processing. We are grateful to Dr. Heiko Schwartz of the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany, for sharing detailed insights into H.264 MGS video coding. The assistance of Anthony Rose of Arizona State University with the tutorial and trace web site preparation is greatly appreciated.

REFERENCES

[1] Cisco, Inc., "Visual networking index: Global mobile data traffic forecast update," Feb. 2010.

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

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[4] W.-C. Feng, *Buffering Techniques for Delivery of Compressed Video in Video-on-Demand Systems*. Kluwer, 1997.

[5] F. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, Nov./Dec. 2001.

[6] M. W. Garret, "Contributions toward real-time services on packet networks," Ph.D. dissertation, Columbia University, 1993.

[7] M. Krunz, R. Saas, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," in *Proc. IEEE Infocom*, Apr. 1995, pp. 455–462.

[8] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," *Performance Evaluation*, vol. 30, no. 1-2, pp. 69–85, 1997.

[9] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Commun. Surveys Tutorials*, vol. 6, no. 3, pp. 58–78, Third Quarter 2004.

[10] M. Alvarez, E. Salami, A. Ramirez, and M. Valero, "A performance characterization of high definition digital video decoding using H.264/AVC," in *Proc. IEEE Int. Symposium on Workload Characterization*, 2005, pp. 24–33.

[11] G. Van der Auwera and M. Reisslein, "Traffic characteristics of H.264/AVC variable bit rate video," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 164–174, Nov. 2008.

[12] G. Van der Auwera, P. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 698–718, Sep. 2008.

[13] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG-4 AVC fidelity range extensions: Tools, profiles, performance, and application areas," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Sep. 2005, pp. 593–596.

[14] D. Marpe, T. Wiegand, and G. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *IEEE Commun. Mag.*, vol. 44, no. 8, pp. 134–143, Aug. 2006.

[15] G. Van der Auwera, P. David, M. Reisslein, and L. J. Karam, "Traffic and quality characterization of the H.264/AVC scalable video coding extension," *Advances in Multimedia, Article ID 164027*, pp. 1–27, 2008.

[16] X. Li, P. Amon, A. Hutter, and A. Kaup, "Performance analysis of inter-layer prediction in scalable video coding extension of H.264/AVC," *IEEE Trans. Broadcast.*, vol. 57, no. 1, pp. 66–74, Mar. 2011.

[17] C. Mazataud and B. Bing, "A practical survey of H.264 capabilities," in *Proc. Communication Networks and Services Research Conference (CNSR)*, 2009, pp. 25–32.

[18] T. Wiegand, L. Noblet, and F. Rovati, "Scalable video coding for IPTV services," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 527–538, Jun. 2009.

[19] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194–1203, Sep. 2007.

[20] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1186–1193, Sep. 2007.

[21] E. Maani and A. K. Katsaggelos, "Optimized bit extraction using distortion modeling in the scalable extension of H.264/AVC," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 2022–2029, Sep. 2009.

[22] F. Fitzek, P. Seeling, and M. Reisslein, "Video and audio trace files of pre-encoded video content for network performance measurements," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, Jan. 2004.

[23] A. Al-Tamimi, R. Jain, and C. So-In, "Modeling and generation of AVC and SVC-TS mobile video traces for broadband access networks," in *Proc. ACM SIGMM Conference on Multimedia Systems (MMSys)*, Phoenix, AZ, Feb. 2010, traces at WUSTL high-definition video trace library, `http://www1.cse.wustl.edu/~jain/sam`.

[24] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video transmission," *Journal of Information Science and Engineering*, vol. 24, pp. 425–440, 2008.

[25] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid—a framework for video transmission and quality evaluation," in *Proc. Computer Performance Evaluation, Modelling Techniques and Tools, Lecture Notes in Computer Science, Vol. 2794*, 2003, pp. 255–272.

[26] A. Alheraish, S. Alshebeili, and T. Alamri, "A GACS modeling approach for MPEG broadcast video," *IEEE Trans. Broadcast.*, vol. 50, no. 2, pp. 132–141, Jun. 2004.

[27] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "On modeling MPEG video traffics," *IEEE Trans. Broadcast.*, vol. 48, no. 4, pp. 337–347, Dec. 2002.

[28] X. D. Huang, Y. H. Zhou, and R. F. Zhang, "A multiscale model for MPEG-4 varied bit rate video traffic," *IEEE Trans. Broadcast.*, vol. 50, no. 3, pp. 323–334, Sep. 2004.

[29] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Study of long-duration MPEG-trace segmentation methods for developing frame-size-based traffic models," *Computer Networks*, no. 44, pp. 177–188, 2004.

[30] W. Zhou, D. Sarkar, and S. Ramakrishnan, "Traffic models for MPEG-4 spatial scalable video," in *Proc. IEEE Globecom*, Dec. 2005, pp. 256–260.

[31] Z. Avramova, D. DeVleeschauwer, K. Laevens, S. Wittevrongel, and H. Bruneel, "Modelling H.264/AVC VBR video traffic: comparison of a markov and a self-similar source model," *Telecommunication Systems*, vol. 39, no. 2, pp. 91–102, 2008.

[32] M. Dai, Y. Zhang, and D. Loguinov, "A unified traffic model for MPEG-4 and H.264 video traces," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 1010–1023, Aug. 2009.

[33] D. Fiems, B. Steyaert, and H. Bruneel, "A genetic approach to Markovian characterisation of H.264/SVC scalable video," *Multimedia Tools and Applications, in print*, 2011.

[34] S. Kempken and W. Luther, "Modeling of H.264 high definition video traffic using discrete-time semi-Markov processes," in *Proc. Int. Teletraffic Congress (ITC), Lecture Notes in Computer Science 4516*, Jun. 2007, pp. 42–53.

[35] A. Lazaris and P. Koutsakis, "Modeling multiplexed traffic from H.264/AVC videoconference streams," *Computer Communications*, vol. 33, no. 10, pp. 1235–1242, Jun. 2010.

[36] N. M. Markovich, A. Undheim, and P. J. Emstad, "Classification of slice-based VBR video traffic and estimation of link loss by exceedance," *Computer Networks*, vol. 53, no. 7, pp. 1137–1153, May 2009.

[37] C. Cicconetti, L. Lenzini, E. Mingozzi, and G. Stea, "Design and performance analysis of the real-time HCCA scheduler for IEEE 802.11e WLANs," *Computer Networks*, vol. 51, no. 9, pp. 2311–2325, Jun. 2007.

[38] J.-W. Ding, C.-T. Lin, and S.-Y. Lan, "A unified approach to heterogeneous video-on-demand broadcasting," *IEEE Trans. Broadcast.*, vol. 54, no. 1, pp. 14–23, Mar. 2008.

[39] T. Gan, K.-K. Ma, and L. Zhang, "Dual-plan bandwidth smoothing for layer-encoded video," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 379–392, 2005.

[40] P. Koutsakis and M. Paterakis, "Policing mechanisms for the transmission of videoconference traffic from MPEG-4 and H.263 video coders in wireless ATM networks," *IEEE Trans. Vehicular Technol.*, vol. 53, no. 5, pp. 1525–1530, 2004.

[41] S. Oh, B. Kulapala, A. Richa, and M. Reisslein, "Continuous-time collaborative prefetching of continuous media," *IEEE Trans. Broadcast.*, vol. 54, no. 1, pp. 36–52, Mar. 2008.

[42] M. Reisslein, F. Hatanto, and K. Ross, "Interactive video streaming with proxy servers," *Information Sciences*, vol. 140, no. 1-2, pp. 3–31, Jan. 2002.

[43] J. Roberts, "Internet traffic, QoS, and pricing," *Proc. IEEE*, vol. 92, no. 9, pp. 1389–1399, 2004.

[44] T. H. Szymanski and D. Gilbert, "Internet multicasting of IPTV with essentially-zero delay jitter," *IEEE Trans. Broadcast.*, vol. 55, no. 1, pp. 20–30, Mar. 2009.

[45] S. Wright, "Admission control in multi-service IP networks: a tutorial," *IEEE Commun. Surveys Tutorials*, vol. 9, no. 2, pp. 72–87, Second Quarter 2007.

[46] X. Zhong and C.-Z. Xu, "Energy-efficient wireless packet scheduling with quality of service control," *IEEE Trans. Mobile Computing*, vol. 6, no. 10, pp. 1158–1170, Oct. 2007.

[47] D. Banodkar, K. Ramakrishnan, S. Kalyanaraman, A. Gerber, and O. Spatscheck, "Multicast instant channel change in IPTV systems," in *Proc. of Int. Conf. on Communication Systems Software and Middleware*, Jan. 2008, pp. 370–379.

[48] H. Hu, J. Yang, Z. Wang, H. Xi, and X. Wu, "Scene aware smooth playout control for portable media players over random VBR channels," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2330–2338, Nov. 2010.

[49] P. Koutsakis, "Using traffic prediction and estimation of provider revenue for a joint GEO satellite MAC/CAC scheme," *Wireless Networks*, vol. 17, no. 3, pp. 797–815, Apr. 2011.

[50] Z. Wang, H. Xi, G. Wei, and Q. Chen, "Generalized PCRTT offline bandwidth smoothing based on SVM and systematic video segmentation," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 998–1009, Aug. 2009.

[51] R. Zhang, R. Ruby, J. Pan, L. Cai, and X. Shen, "A hybrid reservation/contention-based MAC for video streaming over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 389–398, Apr. 2010.

[52] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.

[53] A. Puri, X. Chen, and A. Luthra, "Video coding using the H.264/MPEG-4 AVC compression standard," *Journal of Visual Communication and Image Representation*, vol. 19, no. 9, pp. 793–849, Oct. 2004.

[54] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[55] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1149–1163, Sep. 2007.

[56] H. Schwarz and M. Wien, "The scalable video coding extension of the H.264/AVC standard," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 135–141, Mar. 2008.

[57] J. D. Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Architectures for fast transcoding of H.264/AVC to quality-scalable SVC streams," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1209–1224, Nov. 2009.

[58] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[59] P. Seeling, P. de Cuetos, and M. Reisslein, "Fine granularity scalable (FGS) video: Implications for streaming and a trace-based evaluation methodology," *IEEE Commun. Mag.*, vol. 43, no. 4, pp. 138–142, Apr. 2005.

[60] P. Amon, T. Rathgen, and D. Singer, "File format for scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1174–1185, Sep. 2007.

[61] P. Seeling, M. Reisslein, and F. H. Fitzek, "Offset trace-based video quality evaluation after network transport," *Journal of Multimedia*, vol. 1, no. 2, pp. 1–13, May 2006.

[62] P. Seeling, F. Fitzek, and M. Reisslein, *Video Traces for Network Performance Evaluation*. Springer, 2007.

[63] I. Dalgic and F. Tobagi, "Performance evaluation of ATM networks carrying constant and variable bit-rate video traffic," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1115–1131, Aug. 1997.

[64] T. Lakshman, A. Ortega, and A. Reibman, "VBR video: Tradeoffs and potentials," *Proc. IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[65] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 165–182, Jun. 2011.

[66] S. Hemami and A. Reibman, "No-reference image and video quality estimation: Applications and human-motivated design," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 469–481, Aug. 2010.

[67] A. Moorthy, K. Seshadrinathan, R. Soundararajan, and A. Bovik, "Wireless video quality assessment: A study of subjective scores and objective algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 4, pp. 587–599, Apr. 2010.

[68] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[69] H. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2005.

[70] M. Pinson, S. Wolf, and G. Cermak, "HDTV subjective quality of H.264 vs. MPEG-2, with and without packet loss," *IEEE Trans. Broadcas.*, vol. 56, no. 1, pp. 86–91, Mar. 2010.

[71] M. Pinson and S. Wolf, "Application of the NTIA General Video Quality Metric VQM to HDTV quality monitoring," *Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-07)*, January 2007.

[72] X. Huang, J. Liang, H. Du, and J. Liu, "Lloyd-max quantization-based priority index assignment for the scalable extension of H.264/AVC," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS)*, 2010, pp. 117–120.

[73] B. Gorkemli, Y. Sadi, and A. Tekalp, “Effects of MGS fragmentation, slice mode and extraction strategies on the performance of SVC with medium-grained scalability,” in *Proc. IEEE Int. Conf. on Image Processing*, Sep. 2010, pp. 4201–4204.

[74] H. Lee, Y. Lee, D. Lee, J. Lee, and H. Shin, “Implementing rate allocation and control for real-time H.264/SVC encoding,” in *Digest of Technical Papers International Conference on Consumer Electronics (ICCE)*, Jan. 2010, pp. 269–270.

[75] R. Li, J. Sun, and W. Gao, “Fast weighted algorithms for bitstream extraction of SVC Medium-Grain scalable video coding,” in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2010, pp. 249–254.

[76] H. Kirchhoffer, D. Marpe, H. Schwarz, and T. Wiegand, “A low-complexity approach for increasing the granularity of packet-based fidelity scalability in scalable video coding,” in *Proc. Picture Coding Symposium (PCS)*, 2007.

[77] P. Seeling and M. Reisslein, “The rate variability-distortion (VD) curve of encoded video and its impact on statistical multiplexing,” *IEEE Trans. Broadcast.*, vol. 51, no. 4, pp. 473–492, Dec. 2005.

[78] G. Van der Auwera and M. Reisslein, “Implications of smoothing on statistical multiplexing of H.264/AVC and SVC video streams,” *IEEE Trans. Broadcast.*, vol. 55, no. 3, pp. 541–558, Sep. 2009.

[79] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, “Long-range dependence in variable-bit-rate video traffic,” *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1566–1579, Feb./March/Apr. 1995.

[80] R. Gupta, A. Pulipaka, P. Seeling, L. Karam, and M. Reisslein, “H.264 coarse grain scalable (CGS) and medium grain scalable (MGS) encoded video: A trace based traffic and quality evaluation,” *submitted, preprint available at http://mre.faculty.asu.edu/CGS_MGS_Traffic.pdf*, 2011.

[81] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[82] S. Wenger, Y.-K. Wang, T. Schierl, and A. Eleftheriadis, “RTP payload format for scalable video coding,” draft-ietf-avt-rtp-svc-27.txt, Feb. 2011.

[83] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, “RTP Payload Format for H.264 Video,” RFC 3984 (Proposed Standard), Internet Engineering Task Force, Feb. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3984.txt

[84] S. Wenger, Y.-K. Wang, and T. Schierl, “Transport and signaling of SVC in IP networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1164–1185, Sep. 2007.

[85] A. Vetro, T. Wiegand, and G. Sullivan, “Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard,” *Proc. IEEE*, vol. 99, no. 4, pp. 694–707, Apr. 2011.

[86] A. Vetro, A. Tourapis, K. Muller, and T. Chen, “3D-TV content storage and transmission,” *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 384–394, Jun. 2011.

[87] C. Gurler, B. Gorkemli, G. Saygili, and A. Tekalp, “Flexible transport of 3-D video over networks,” *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[88] W.-C. Feng and J. Rexford, “Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video,” *IEEE Trans. Multimedia*, vol. 1, no. 3, pp. 302–313, Sep. 1999.

[89] M. Krunz, “Bandwidth allocation strategies for transporting variable-bit-rate video traffic,” *IEEE Commun. Mag.*, vol. 37, no. 1, pp. 40–46, Jan. 1999.

[90] M. Reisslein and K. Ross, “High-performance prefetching protocols for VBR prerecorded video,” *IEEE Network*, vol. 12, no. 6, pp. 46–55, Nov./Dec. 1998.

[91] B. Vandalore, W.-C. Feng, R. Jain, and S. Fahmy, “A survey of application layer techniques for adaptive streaming of multimedia,” *Real-Time Imaging Journal*, vol. 7, no. 3, pp. 221–235, 2001.

[92] Z. Antoniou and I. Stavrakakis, “An efficient deadline-credit-based transport scheme for prerecorded semisoft continuous media applications,” *IEEE Trans. Netw.*, vol. 10, no. 5, pp. 630–643, Oct. 2002.

[93] J. Yuen, E. Chan, and K. Lam, “Real time video frames allocation in mobile networks using cooperative prefetching,” *Multimedia Tools and Applications*, vol. 32, no. 3, pp. 329–352, Mar. 2007.

[94] Y. Chen, B. Wang, W. Lin, Y. Wu, and K.J.R. Liu, “Cooperative peer-to-peer streaming: An evolutionary game-theoretic approach,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1768–1784, Jul. 2010.

[95] X. Liu, G. Cheung, and C.-N. Chuah, “Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast,” *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 1346–1357, Oct. 2010.

[96] Z. Guan, T. Melodia, and D. Yuan, “Optimizing cooperative video streaming in wireless networks,” in *Proc. IEEE Int. Conf. on Sensor, Mesh and Ad Hoc Communications and Networking (SECON)*, Jun. 2011.

[97] W. Lin, H. Zhao, and K.J.R. Liu, “Cooperation stimulation strategies for peer-to-peer wireless live video-sharing social networks,” *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1768–1784, Jul. 2010.

[98] C. Liang, M. Zhao, and Y. Liu, “Optimal bandwidth sharing in multiswarm multiparty P2P video-conferencing systems,” *IEEE Trans. Netw.*, vol. 20, no. 10, pp. 1346–1357, Oct. 2010.

[99] A. Dua, C. Chan, N. Bambos, and J. Apostolopoulos, “Channel, deadline, and distortion (CD^2) aware scheduling for video streams over wireless,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 3, pp. 1001–1011, Mar. 2010.

[100] M. Jubran, M. Bansal, L. Kondi, and R. Grover, “Accurate distortion estimation and optimal bandwidth allocation for scalable H.264 video transmission over MIMO systems,” *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 106–116, Jan. 2009.

[101] J. Nightingale, Q. Wang, and C. Grecos, “Optimised transmission of H.264 scalable video streams over multiple paths in mobile networks,” *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2161–2169, Nov. 2010.

[102] A. Dhaini, P.-H. Ho, and X. Jiang, “WiMAX-VPON: A framework of layer-2 VPNs for next-generation access networks,” *IEEE/OSA J. Optical Commun. Netw.*, vol. 2, no. 7, pp. 400–414, Jul. 2010.

[103] N. Ghazisaidi, M. Maier, and M. Reisslein, “VMP: A MAC protocol for EPON-based video-dominated FiWi access networks,” *submitted for review, preprint: http://mre.faculty.asu.edu/VMP.pdf*, 2011.

[104] H.-M. Jeong, M.-J. Lee, D.-K. Lee, and S.-J. Kang, “Design of home network gateway for real-time A/V streaming between IEEE1394 and Ethernet,” *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 390–396, May 2007.

[105] X. Liu, N. Ghazisaidi, L. Ivanescu, R. Kang, and M. Maier, “On the tradeoff between energy saving and QoS support for video delivery in EEE-based FiWi networks using real-world traffic traces,” *IEEE/OSA J. Lightwave Technol.*, vol. 29, no. 18, pp. 2670–2676. Sep., 2011.

[106] M. Maier, N. Ghazisaidi, and M. Reisslein, “The audacity of fiber-wireless (FiWi) networks,” in *Proc. ICST Int. Conf. on Access Networks (ACCESSNETS)*, Oct. 2008, pp. 16–35.

[107] P. Pangalos, J. De La Torre Velver, M. Dashti, A. Dashti, and H. Aghvami, “Confirming connectivity in interworked broadcast and mobile networks,” *IEEE Network*, vol. 21, no. 2, pp. 13–20, March/April 2007.

[108] A. Carena, V. De Feo, J. Finochietto, R. Gaudino, F. Neri, C. Piglione, and P. Poggiolini, “RingO: an experimental WDM optical packet network for metro applications,” *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1561–1571, Oct. 2004.

[109] H.-S. Yang, M. Herzog, M. Maier, and M. Reisslein, “Metro WDM networks: Performance comparison of ring and star topologies,” *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1460–1473, Oct. 2004.

[110] I. White, M. Rogge, K. Shrikhande, and L. Kazovsky, “A summary of the HORNET project: a next-generation metropolitan area network,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 9, pp. 1478–1494, Nov. 2003.

[111] A. Abdel Khalek and Z. Dawy, “Energy-efficient cooperative video distribution with statistical QoS provisions over wireless networks,” *IEEE Trans. Mobile Computing, in print*, 2011.

[112] N. Xu, J. Yang, M. Needham, D. Boscovic, and F. Vakil, “Toward the green video CDN,” in *Proc. of IEEE/ACM Int. Conf. on Green Computing and Communications (GreenCom)*, 2010, pp. 430–435.

Patrick Seeling is an Assistant Professor in the Department of Computer Science at Central Michigan University (Mount Pleasant, MI). He received his Dipl.-Ing. Degree in Industrial Engineering and Management from the Berlin Institute of Technology (Berlin, Germany) in 2002 and his Ph.D. in Electrical Engineering from Arizona State University (Tempe) in 2005. He was a Faculty Research Associate and Associated Faculty with the Department of Electrical Engineering at Arizona State University from 2005 to 2007. From 2008 to 2011, he was an Assistant Professor in the Department of Computing and New Media Technologies at the University of Wisconsin-Stevens Point (Stevens Point, Wisconsin, USA). Patrick Seeling has published over 40 journal articles and conference papers, as well as books, book chapters, and tutorials in the areas of multimedia, optical, and wireless networking and engineering education. He serves in editorial and reviewer capacities for different journals and as technical program committee member for conferences sponsored by several societies. Patrick Seeling is a Senior Member of the ACM and IEEE.

Martin Reisslein is a Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU), Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994–1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin and lecturer at the Technical University Berlin. From 2000–2005 he was an Assistant Professor and from 2005–2011 an Associate Professor with ASU. He currently serves as Associate Editor for the *IEEE/ACM Transactions on Networking* and for *Computer Networks*. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG-4 and H.264 encoded video, at `http://trace.eas.asu.edu`. His research interests are in the areas of video traffic characterization, wireless networking, optical networking, and engineering education.