CrossMark

# SVCEval-RA: an evaluation framework for adaptive scalable video streaming

**Wilder E. Castellanos**[1] · **Juan C. Guerri**[1] · **Pau Arce**[1]

**Abstract** Multimedia content adaption strategies are becoming increasingly important for effective video streaming over the actual heterogeneous networks. Thus, evaluation frameworks for adaptive video play an important role in the designing and deploying process of adaptive multimedia streaming systems. This paper describes a novel simulation framework for rate-adaptive video transmission using the Scalable Video Coding standard (*H.264/SVC*). Our approach uses feedback information about the available bandwidth to allow the video source to select the most suitable combination of *SVC* layers for the transmission of a video sequence. The proposed solution has been integrated into the network simulator NS-2 in order to support realistic network simulations. To demonstrate the usefulness of the proposed solution we perform a simulation study where a video sequence was transmitted over a three network scenarios. The experimental results show that the *Adaptive SVC* scheme implemented in our framework provides an efficient alternative that helps to avoid an increase in the network congestion in resource-constrained networks. Improvements in video quality, in terms of *PSNR* (Peak Signal to Noise Ratio) and *SSIM* (Structural Similarity Index) are also obtained.

## 1 Introduction

The massive demand for video content from video streaming services such as Amazon Streaming Service, YouTube, iTunes, Netflix and BitTorrent streaming, has led to the increase

✉ Juan C. Guerri
  jcguerri@dcom.upv.es

  Wilder E. Castellanos
  wilcashe@upv.es

  Pau Arce
  paarvi@iteam.upv.es

[1] Institute of Telecommunications and Multimedia Applications (iTEAM), Universitat Politècnica de València, Valencia, Spain

⚛ Springer

in video traffic on the Internet. According to the most recent Cisco Visual Networking Index (VNI) [9], predicts that IP video traffic will represent 79 % of all traffic by 2018. Because video traffic accounts for a significant portion of the IP data traffic, and given the growing demand of high quality video, the development of new mechanisms for video rate adaptation over different network scenarios is a topic of intense study. In addition, on account of the heterogeneity of the current networks which include fixed and wireless links as well as non-homogeneous clients with different decoding and display capabilities, it is required that video applications can adapt to the network conditions. The adaptation of multimedia streaming involves many challenges. On the one hand, the algorithms for estimation of the network state, the rate control mechanisms and the network/transport layer protocols are required to assure the provision of fair and stable services. On the other hand, applications and suitable coding techniques also must be developed for performing content adaptation to the free resources on the network. As an answer of these challenges many solutions have been proposed in the last decade among them *DCCP* (Datagram Congestion Control Protocol) [14], *TFRC* (*TCP Friendly Rate Control*) [15], *RTMP* (Real Time Messaging Protocol) [28] and recently, *DASH* (Dynamic Adaptive Streaming over HTTP) [17]. All these solutions have different objectives, but agree on the target goal of assisting the network to provide adaptive media streaming.

Regarding the content adaptation, Scalable Video Coding (*H.264/SVC*) [18] is a coding technique that offers an efficient alternative for adaptive content distribution. It was standardized in 2007 as an extension of the *H.264/AVC* (Advanced Video Coding) standard, and several SVC-based solutions for video transmission have been proposed so far, for example the works presented in [2, 3, 8, 27, 30, 31]. A video stream encoded with *H.264/SVC* consist of several layers (one base layer and multiple enhancement layers) each with different characteristics of quality. Moreover, the adaptability feature of the bit rate, inherent of the scalable coding scheme, provides a natural and efficient way to adapt the video source rate to the available network throughput. This bit-rate adaptation has a low computational cost in *SVC* since it is performed without the need to transcode: layers can be added and removed with low complexity operations. In addition, improvements in the subjective quality of videos encoded with *SVC*, compared with *H.264/AVC*, has been demonstrated in some studies such as [29, 43].

An open issue in the development of adaptive video delivery systems is the lack of freely available video evaluation tool-set where researchers and network developers can test the performance of their network-adaptive techniques. One possibility for support of the latter is to use the *Evalvid-RA* (Evalvid Rate Adaptive) framework developed by Arne Lie et al. [25]. *Evalvid-RA* is a simulation tool that supports rate adaptive video transmission using *MPEG-4 VBR* (Variable Bit Rate) videos. The adaptive mechanism of this platform involves switching between different pre-encoded versions of the video (each with different levels of quality) according to bandwidth variations. However, a more efficient solution is provided by *H.264/SVC*, which supports encoding of a video in different qualities within the same bit-stream. This includes different resolutions, different frame rates (fps) and different quality levels. Although some evaluation platforms have been proposed to assess the transmission of *SVC* videos, such as *SVEF* (Scalable Video-streaming Evaluation Framework) [11], *EvalSVC* [24] and *myEvalSVC* [21]. However, none of these evaluation platforms enables network-adaptive streaming operation at the video source, which always injects video streams with the highest video quality (i.e. at the highest bit rate) to the network.

We propose in this paper a novel simulation framework called *SVCEval-RA* (*SVC Evaluation Platform for Rate-Adaptive Video*), which can be used in the assessment of rate-adaptive video streaming using Scalable Video Coding. Our approach is a modified and enhanced version of the *myEvalSVC* tool-set. More precisely, we have implemented in SVCEVal-RA a procedure to adapt the bit rate of the traffic source adding or removing SVC layers from the video stream based on the estimation of the available bandwidth. This procedure (referred to hereafter as the *Adaptive SVC scheme* of *SVCEval-RA*) provides better conditions to carry out performance evaluations of network protocols for adaptive video transmission, compared with the other evaluation platforms. For providing a more realistic simulation environment, we have integrated our algorithms with the well-known simulation tool *NS-2* [38]. Nevertheless, as discussed later, the proposed architecture can be integrated within any network simulator.

In order to demonstrate the usefulness of our proposed solution we perform a simulation study. We set different network scenarios in order to evaluate the rate adaptation under available bandwidth variations. Specifically, we aim to explain how the *Adaptive SVC scheme* implemented in *SVCEval-RA* allow the video source to adapt the transmission rate under bandwidth variations. We perform a simulation study where our framework is used in the evaluation of rate-adaptive scalable video streaming over three network scenarios. These scenarios emulate some common patterns of the available bandwidth in network scenarios such as *WLANs* and Internet. The test videos used in the simulations have been encoded according to the Scalable Video Coding standard (*H.264/SVC*) with temporal and quality scalability. The experimental results show that *SVCEval-RA* provides an efficient platform to perform simulation studies that involve adaptive video transmission, enabling researchers to investigate new strategies of streaming media, both at the network/transport layer and at the application layer.

The main goal of this paper is to present the *SVCEval-RA* architecture, to validate its performance and lastly to exemplify the utilization of our platform in the configuration of simulation studies, showing how the video source adjusts its data rate transmitting only the SVC layers that can be efficiently handled by the network.

The rest of the paper is organized as follows. Section 2 summarizes the most important related works on video transmission evaluation. Section 3 gives first a brief introduction to the methods for adaptive scalable video streaming and then provides an overview of the scalable video coding. In Section 4 the main components of the proposed *SVCEval-RA* framework are introduced and explained, and the main capabilities are demonstrated in Section 5. The results are discussed in Section 6 and finally, the conclusions and future work are presented in Section 7.

## 2 Related works

In video transmission evaluation it is necessary to have some video characterization available. Basically, three methods can be used to characterize encoded video in networking research: *i*) video bit stream, *ii*) video traffic model, and *iii*) video traffic trace. The first option involves the generation of a video bit stream using the output of a video encoder, so that it contains the complete video information. This method is focused mainly on real networks. Therefore, expertise and equipment are necessary for both video coding and networking in order to perform research studies on video transmission evaluation with actual bit streams. In addition, the difficulty to carry out a reliable assessment of new protocols is another limitation. Some

examples of video bit stream include *REALTRACER* [41] and *ORB* [20]. The second method is based on using video traffic models, which are mathematical models that describe the essential properties of the real video streams. These models are generally developed based on the statistical analysis of video trace samples of real video traffic. Several models for video traffic have been proposed in the literature. For instance, the studies of Dai et al. [10], Ivrlac et al. [19] and Espina et al. [12] describe traffic models for *H.264/AVC* encoded video and the recent surveys presented by Tanwir and Perros [36, 37] where several *VBR* video traffic models are analysed. Regarding traffic models for scalable video, can be consulted in the references [13, 35, 40]. Although using mathematical modelling has advantages such as the rapid generation of the results for video traffic and queue statistics analysis, the complexity of the actual heterogeneous networks makes realistic simulation a better tool, especially when an evaluation of the perceived visual quality of the video is required. Finally, the third method is the generation of video traffic traces which are an abstraction of real video stream. A video trace is a text file where the characteristics of real video traffic, such as the frame number, frame type and frame size, are described. These video traces can be employed to simulate video traffic in a wide range of network simulation tool, which has been considered a viable approach for evaluation of various video delivery systems. Several simulation tools of video traffic based on trace files have been developed, for example the *EvalVid* tools-set [23] and its extended versions for *NS-2* suggested by Ke et al. [22] and Lie et al. [25]. These platforms support trace-driven simulations of *H.263* and *H.264/AVC* video transmission. Concerning simulation platforms for scalable video, in recent years, have been implemented few solutions, such as *SVEF* (Scalable Video-streaming Evaluation Framework) [11]. *SVEF* is a tools-set for empirical evaluation of real- time *SVC* streaming. However, its practical application is limited because *SVEF* tool-set cannot support experiments that involve new technologies or network protocols. Moreover, it does not take the spatial scalability into consideration and only supports *SVC* videos with a single dependency layer. Other evaluation platforms for scalable video streaming are *EvalSVC* (Evaluation of *SVC*) and *myEvalSVC* simulation framework, which were developed by Le et al. *[24]* and Ke [21] respectively. These simulations platforms extend the *EvalVid* framework in order to provide a simulation framework where *SVC* video transmissions can be evaluated over a general network simulator as NS-2. Nevertheless, these frameworks do not provide a suitable solution for rate-adaptive video investigation. In order to solve the above-mentioned problem, we proposed the *SVCEval-RA* evaluation framework for more realistic simulations of adaptive scalable video transmission. The main contributions of SVCEval-RA compared to referenced works are:

- The *SVCEval-RA* framework is enhanced to support rate adaptive scalable video. Therefore, in contrast to the tools developed so far where the *SVC* stream is always transmitted with the highest quality (i.e. at the highest bit rate), with *SVCEval-RA* video source can transmit only the *SVC* layers that can be supported by the network.
- The modular structure of *SVCEval-RA* makes possible that third party modules can be easily integrated in order to extend its capabilities. For instance, transport protocols for video streaming with their own algorithms for network resource estimation, such as *TFRC* (*TCP* Friendly Rate Control) and *RTMP* (Real Time Messaging Protocol).
- Simulation studies using adaptive *SVC* techniques can be performed using SVCEval in different network scenarios (e.g. Internet, *WLANs* and *MANET*s -Mobile Ad-hoc NETworks) where video streaming services require adaptation to the network conditions, such as *DASH*.

- In contrast to the other simulation platforms, *SVCEval-RA* supports *SVC*-encoded video flows with both quality and temporal scalability. In addition, our simulation framework enables the MGS (Medium Grain Quality Scalability) coding scheme (instead of CGS that is used in most simulation frameworks). Using MGS allows us to obtain a better coding efficiency and granularity, since it allow us to switch between layers at any time.
- A simulator-independent tool, written in C++, is also provided. This software integrates, in only one application, the tasks that must be executed for the interfacing between the encoder and the simulator.

## 3 Adaptive scalable video streaming

The methods for video streaming covers a wide range of technologies: from simple progressive download, where a single video file is downloaded over *HTTP* and played back at the same time, to real time streaming techniques using packet-based delivery, for instance with the Real-time Transport Protocol (*RTP*) [16, 33]. However, these traditional video transmission schemes lack of the necessary flexibility and adaptability since the video file is delivered to the end-user with a constant bit rate through a persistent source-destination connection. Thus, if a particular transmission channel experiences congestion, the quality of the video can significantly degraded (see Fig. 1a).



Fig. 1 Comparison of the traditional video streaming, the scalable video streaming and the adaptive video streaming
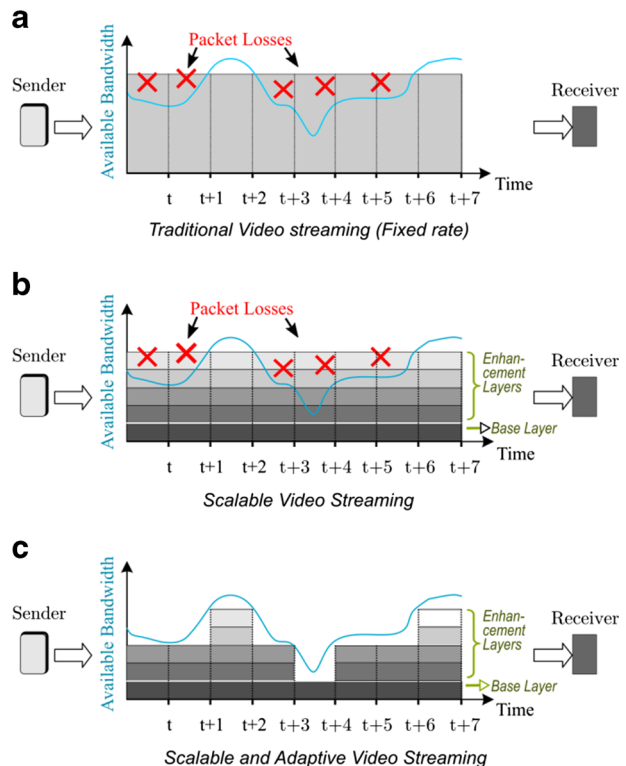
The scalable video coding (*H.264/SVC*) provides features to generate different representations of the same video integrated within a same bit stream. A video encoded using the *SVC* standard has a layered structure where the layers correspond to different quality, spatial or temporal representations. A *SVC* video is composed of a base layer, which corresponds to the lowest representation, and one or more enhancement layers that increase the video quality when these are added to the base layer. The layered scheme of *SVC* can provide higher robustness during video streaming over networks with high packet loss rates. This robustness allows the receiver to decode the video even if there is a high packet loss rate (see Fig. 1b) because if enhancement layers are lost, the receiver may still decode received video as long as the base layer is correctly received. However, it is not a bandwidth-efficient solution because of the large amount of data sent and lost due to packet losses. Therefore, in order to adaptively control the bit rate of the video source, the adoption of cross-layer mechanisms in video streaming is required. Cross-layer solutions involves information exchange and optimization between the application layer operations (e.g. source coding), the network layer and the transport protocols in order to obtain optimal combinations of video bit rates and routing policies. These solutions can be used for supporting adaptive video streaming schemes. In fact, the combined usage of *SVC* and adaptive streaming techniques can significantly contribute to increasing the quality of video streaming services while the bandwidth efficiency is achieved (see Fig. 1c).

### 3.1 H.264/SVC scalable video coding

In this section, we present a brief description of the main technical features of *SVC*, the scalable extension of the H.264/AVC standard [18]. A more detailed explanation of the fundamentals of *SVC* can be found in the study of Schwarz et al. [32].

With *H.264/SVC*, the encoder produces a scalable bit-stream, which consists of a multiple layers. A base layer provides a basic video quality (e.g. low spatial or temporal resolution) and adding enhancement layers improves the quality (e.g. increases spatial resolution or frame rate). There are three modes of video scalability supported by *SVC*: *temporal scalability*, *spatial scalability* and *quality scalability*. When using *temporal scalability*, layers improve the frame rate. With *spatial scalability*, the base layer is coded at a low spatial resolution and enhancement layers give progressively higher spatial resolution. Finally, *quality scalability* refers to scaling in terms of the level of compression applied to the source video. This is primarily controlled using the quantization parameter (*QP*). With quality scalability the base layer contains a strongly compressed version of each picture, and enhancement layers incorporate more information to increase the SNR (Signal-Noise-Ratio) value. The *H.264/SVC* standard supports combined scalability, i.e. a scalable video can use any combination of the three types of scalabilities. For instance, Fig. 2 shows a *SVC* stream encoded with four temporal levels $\{T_0, T_1, T_2 \text{ and } T_3\}$ and two quality levels $\{Q_0 \text{ and } Q_1\}$. Thus, eight scalable layers are generated by combining these levels as is listed in Fig. 2. The base layer consists of the lowest temporal resolution ($T_0$) and the lowest quality level $Q_0$ (i.e. frames 1 and 8). In addition, an example of enhancement layer can be the layer consisting of the temporal layer $T_2$ and the quality level $Q_1$. This encoded video stream exploits the hierarchical prediction structure using B-pictures for enabling temporal scalability. Moreover, the coding structure of the quality scalability uses the key picture concept (other coding structures can be consulted in [32, 39]).
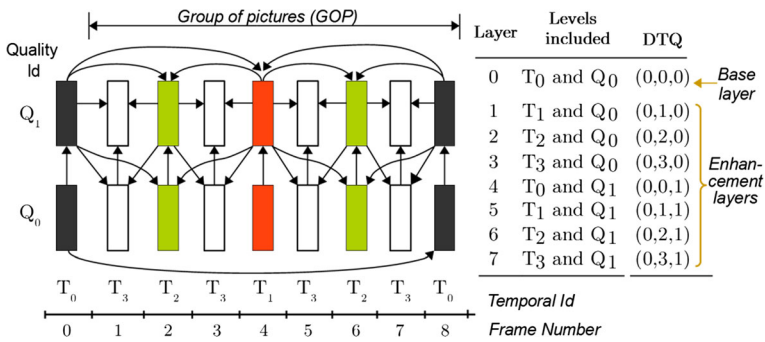
**Fig. 2** Example of coding structure of a SVC stream with temporal and quality scalability

In *H.264/SVC*, the codec is divided in two subsystems: the Video Coding Layer (*VCL*) and the Network Abstraction Layer (*NAL*). Basically, the *VCL* is in charge of the source video coding and the *NAL* is the interface between the encoder and the actual network protocol, which will be used to transmit the encoded bit-stream. In this work, we focus our attention on the *NAL* subsystem, since it provides the required information to identify the data relating to each layer.

In the VCL subsystem, each picture of the video is partitioned into macroblocks that each covers a rectangular area of $16 \times 16$ luminance samples. The macroblocks of a picture are organized in slices each of which can be parsed independently of other slices in a picture. Each slice encoded in *VCL* is encapsulated into Network Abstraction Layer Units (*NALU*), which are suitable for transmission over networks. The main characteristic of the *NALU*s used in *SVC* is a sequence of 3 bytes that contain three identifiers: dependency identifier (*DID*), temporal identifier (*TID*) and quality identifier (*QID*). These identifiers represent a point in the spatial, temporal and quality scalable dimensions, respectively. The values of *DID*, *TID* and *QID* are also known as *DTQ* parameters. For instance, a *NALU* of the base layer should be identified as (0,0,0) and for instance the enhancement layer consisting of the $T_2$ and $Q_1$ levels should be identified as (0,2,1). The inspection of the *DTQ* values permits to identify the *NALU*s belonging to a specific enhancement layer. This fact is particularly important since the *NALU*s may be identified and removed from the *SVC* encoded video, in order to reduce the bit rate. The remaining *NALU*s may integrate a valid video bit-stream with a lower visual quality. Nevertheless, the dependencies of the layers would be taken into account. Layers in *SVC* can be decoded independently but there is a logical dependency between them. This interdependency must be considered in order to obtain a correct decoding of the video. In the example shown in Fig2, the arrow lines represent dependencies between frames in a combined scalable stream. For instance, the frame 2 of the layer $T_2Q_1$ depends on the layer $T_0Q_1$ and the layer $T_1Q_1$ (which also depends on base layer). Because of these dependencies, discarding a quality layer from a reference frame (e.g. frame 2) affects the quality of dependent frames (e.g. frames 1 and 3).

## 4 The SVCEval-RA evaluation framework

In this section, we describe the main characteristics of our proposed *SVCEval-RA* framework. The main goal of the proposed framework is to assess the transmission of rate-adaptive video

supporting *SVC* encoding. Our framework performs video transmissions over a packet net-
work using a general network simulator. Thus, any specific trace format defined by the
integrated simulator can be easily adapted in our evaluation platform. However, in order to
illustrate its operation we integrated our algorithms to the well-known NS-2 simulator. In this
paper, we focus on showing how our approach can be used as a complete framework for
evaluation of video transmission in different network scenarios. For instance, simulation
studies of network scenarios that include rate control protocols based on *TFRC* [15] can be
performed using our platform. Simulations of *MANET* (Mobile Ad hoc NETworks) are also
supported. For this kind of networks, the mechanisms for estimating the available bandwidth
can be provided by a QoS routing protocol, e.g. *AQA-AODV* (Adaptive QoS-Aware for Ad hoc
On-demand Distance Vector), which can be integrated into SVCEval-RA. More detailed
information about *AQA-AODV* can be consulted in the references [5, 6]. Best-effort protocols
without rate controller, in wired or wireless networks also can be used in our framework. In
this last case, the adaptation of the transmission rate can be configured in the simulation script
file. It is important to note that the accuracy of the video stream adaptation mainly depends of
the performance of the network protocols used in the simulation. For instance, an inaccurate
estimation of the available bandwidth provided by a network protocol may lead to an
inaccurate adaptation of the SVC stream.

Figure 3 illustrates the workflow of the *SVCEval-RA* evaluation framework. The software
modules inherited from the *JSVM* (Joint Scalable Video Model) package [26] and *SVEF*
(Scalable Video coding Streaming Evaluation Framework) scripts [11] are represented in grey.
The whole process, from the encoding of the original video source to the evaluation after the
streaming over a simulated network can be summarized in four phases, better detailed in the
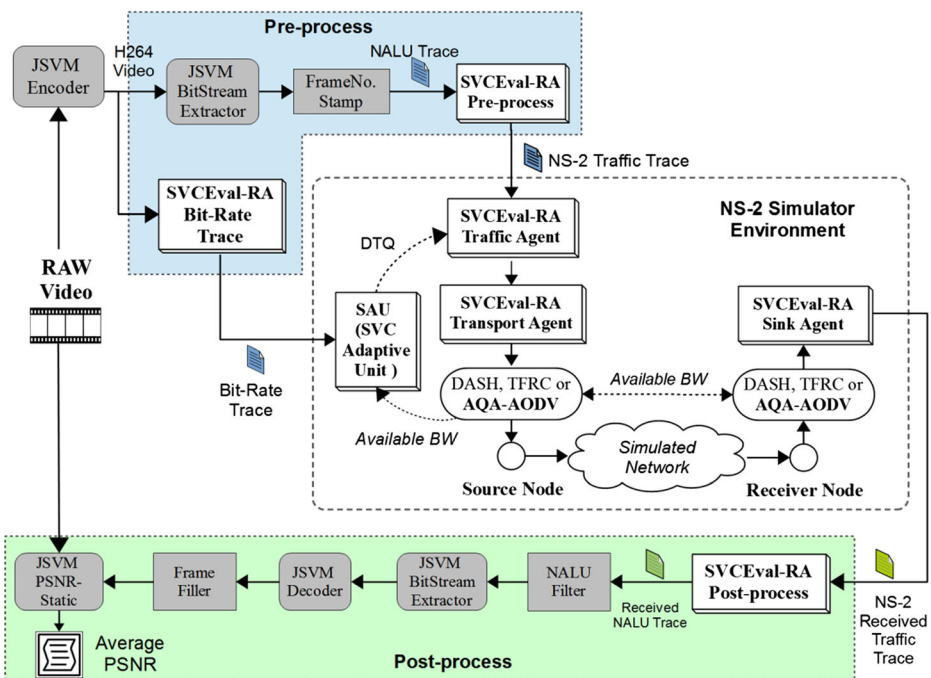


**Fig. 3** Workflow of the SVCEval-RA framework

following sub-sections: encoding, pre-processing, simulation and post-processing. In the encoding phase, the original uncompressed video is encoded in *H.264/SVC* format through the *JSVM* Encoder and a corresponding trace file (*NALU trace*) is produced. Pre-process consists of the generation of the *Bit Rate Trace* and the *NS-2 Traffic Trace* file. During simulation, the *NALU*s are forwarded to the receiver through a simulated network. Moreover, the source can adjust its data rate by selecting a set of *SVC* layers to be transmitted based on the estimation of the available bandwidth. After simulation, the received *NALU trace* is processed in order to produce a *YUV* file, which may be affected by missing frames due to transmission losses, unsatisfied decoding dependencies or excessive delay. Finally, the *PSNR* (Peak Signal-to-Noise Ratio) is evaluated by comparing the original uncompressed video with the distorted one. In Fig. 3, the different phases are further detailed.

## 4.1 Video encoding

The framework uses *RAW* video files stored in the standard *YUV* format. The *RAW* video is then encoded according to the *H.264/SVC* coding scheme. Encoding is performed using the *JSVM* software, which is the reference implementation of the *H.264/SVC* standard. In particular, the *JSVM* tool *H264EncoderStatic* encodes a *YUV* video by configuring all the *H.264/SVC* encoding settings, including the type of scalability to be used, and the number of layers that will compose the output stream. After that, the *NALU trace* file is produced from the *H.264/SVC* encoded video using the *BitStreamExtractor JSVM* tool. This *NALU trace* file includes a different register for each packet of the *NALU* sequence. Among other parameters, each register specifies the size of the packet, the *DTQ* values and the packet type, which are needed during the decoding phase. Combined scalability (temporal and *SNR* scalability) can be used in the codification of the video sequences. In addition, *SNR* scalability can use of the *MGS* scheme, which provides better coding efficiency and granularity since it allow us to switch between layers at any time.

## 4.2 The Pre-process

The first task of the pre-process is to add the frame number information to the *NALU trace* file using *the F-N Stamp* tool. On the other hand, the *Bit Rate Trace* is generated. This *Bit Rate Trace* file provides information about the bit rate required to transmit each layer of the *SVC* encoded video. Table 1 shows a fragment of a *Bit Rate Trace*. The meanings of all fields are as follows: layer id, frame rate (Hz), bit rate, *DID*, *TID*, *QID* and *PSNR* (dB). For instance, according to the Table 1, layer 2 with *DTQ* values (0,3,0) has a bit

**Table 1** Example of Bit Rate Trace

| Layer id | Frame rate | Bit-rate | DID | TID | QID | PSNR (dB) |
|----------|-----------|----------|-----|-----|-----|-----------|
| 0 | 3.7500 | 139.90 | 0 | 1 | 0 | 19.6600 |
| 1 | 7.5000 | 197.40 | 0 | 2 | 0 | 22.4023 |
| 2 | 15.000 | 273.30 | 0 | 3 | 0 | 27.1010 |
| 3 | 30.000 | 335.40 | 0 | 4 | 0 | 35.5127 |
| … | … | … | … | … | … | … |

rate of 273.3 kbps which is the required bit rate to transmit the layer 2 and its dependent lower layers. The value of 27.1 dB is an estimation of the *PSNR* value that would be obtained if all *NALU*s of the layer 2 and the dependent layers were correctly received after the video transmission.

The diagram shown in Fig. 4 describes in general form, the main tasks involved in the generation of the *Bit Rate Trace* file. Basically, the procedure involves generating sub-streams extracting each *SVC* layer from the entire scalable video sequence. These extracted sub-streams are decoded and then the *PSNR* is calculated. In the measurement of the *PSNR* both sequences (sub-stream and original *RAW* video) must have the same spatial and temporal resolutions, therefore frame copy operations (Frame Filler) must be applied to the extracted sub-streams. This method is suggested by Sohn et al. in [34] and it consists of duplicating the frames of the lower layer so that both sequences have the same number of frames. We provide in Section 5.2 a more detailed description of the method used in this study to calculate the *PSNR* of the test sequences.

As it is described in Fig. 3, after the *Bit Rate Trace* file is generated, the *SVCEval-RA Pre-process* tool adapts the *NALU trace* file to the appropriated format required by the *SVCEval-RA Traffic Agent*. The new trace file, named *NS-2 Traffic Trace*, contains the following fields for each record: sending time, frame size, the *DTQ* values (*DID*, *TID*, *QID*) and the number of fragmented packet. Finally, the *Bit Rate Trace* and the *NS-2 Traffic Trace* are sent to the *SVC* Adaptive Unit and the *SVCEval-RA Traffic Agent* integrated in NS-2, respectively.



Fig. 4 Flowchart of the generation of the Bit Rate Trace
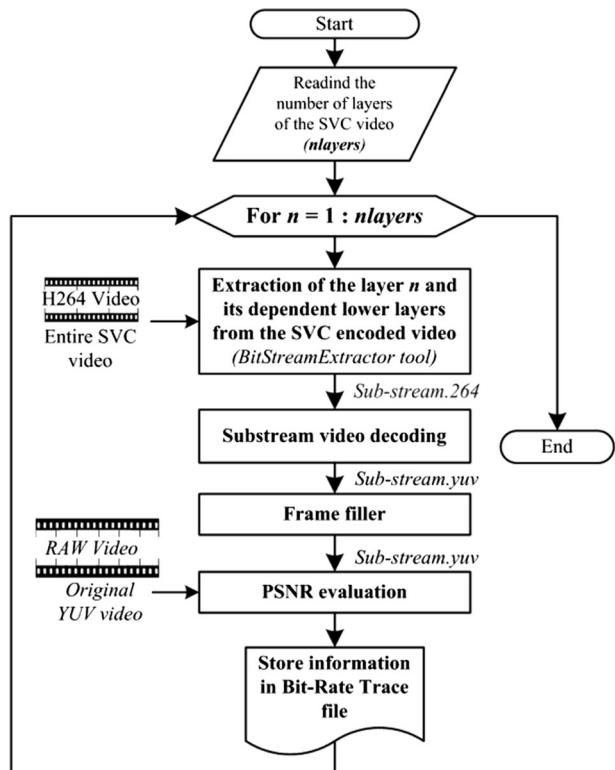
### 4.3 Simulation

As shown in Fig. 3, four *NS-2 agents* were implemented. In the sender side are the *SVCEval-RA Traffic Agent*, the *SVCEval-RA Transport Agent* and the *SVC Adaptive Unit* (*SAU*). In the receiver side is the *SVCEval-RA Sink Agent*. The *SVCEval-RA Traffic Agent* reads each record from the *NS-2 Traffic Trace* file, generates the corresponding number of packets and sends them to the lower layer at the appropriate time according to the user settings specified in the simulation script file. The tasks related with our *adaptive SVC scheme* are executed in the *SAU* and the *Traffic Agent*, which makes possible the successful adaptation of the sending rate transmitting only the SVC layers that can be efficiently handled by the network.

The *SVCEval-RA Transport Agent* is an extension of the UDP agent of NS-2. More specifically, it records the sending time, packet ID and the packet size of each transmitted packet. Then those packets go down to the lower layers and to the simulated networks.

As mentioned above, our framework can be integrated with different protocols (e.g. *TFRC* or *AQA-AODV*), which must provide an algorithm to estimate the available bandwidth along a route. The value of available bandwidth estimated by the streaming protocol is sent to the *SVC Adaptive Unit* (*SAU*). The *SAU* searches in the *Bit Rate Trace* the *DTQ* parameters of the highest *SVC* layer that can be transmitted without exceeding the bandwidth constraint. The *SVC* layer selected is the layer with a bit rate value less or equal than the data rate supported by the network.

The *SAU* sends the *DTQ* parameters of the selected layer to the *SVCEval-RA Traffic Agent*. It will transmit only the *NALU*s identified with the same *DTQ* parameters as well as the *NALU*s of the dependent lower layers.

At the receiving side, the *SVCEval-RA Sink Agent* is used to receive the fragmented video packets sent by the *SVCEval-RA Transport Agent*. This agent also records the receiving time, frame number, packet size, *DID*, *TID*, *QID*, packet ID and the sending time in the *Receiver Traffic Trace* file. The diagram shown in Fig. 5 describes in general form, the main tasks involved in the adaptation process of video transmission using our *Adaptive SVC* scheme.

**Fig. 5** General adaptation scheme of SVCEval-RA
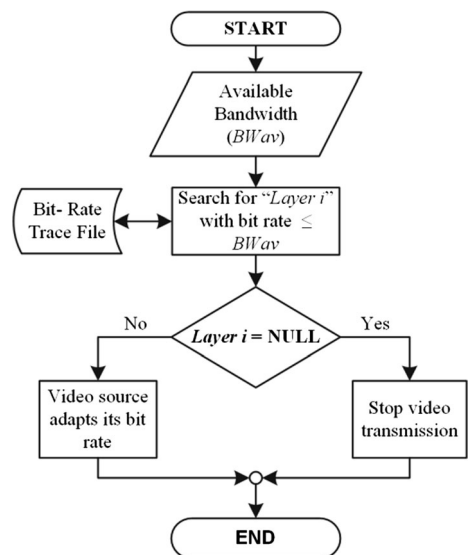
### 4.4 Post-processing and decoding

The main post-processing tasks are illustrated in the lower part of Fig. 3. After simulation, the *Received Traffic Trace* is first processed by the *SVCEval-RA Post-process* program in order to obtain the *Received NALU Trace* file. Afterward, the *Received NALU Trace* file is passed to the *NALU*-Filter tool which: *i*) removes *NALU*s received after the play-out buffer deadline, *ii*) reorders the *NALU*s according to the sending order and, *iii*) removes *NALU*s with unfulfilled decoding dependencies. The filtered *Received NALU Trace* file is used to reconstruct the *H.264/SVC* stream using the *BitStreamExtractor* and the *JSVM Encoder* tools. The obtained *SVC* distorted video can then be decoded to a *YUV RAW* file by means of the *JSVM H264DecoderStatic* tool. However, the resulting *YUV* video could have, in general, fewer frames than the original *YUV* video for two reasons: *i*) missing *NALU*s, and *ii*) static decoder skips lost frames without considering a replacement for them. Therefore, in order to properly evaluate the *PSNR*, the receiver *YUV* video is further modified by the *Frame-Filler*. This tool adds, for each missing frame, a copy of the immediately previous one obtaining the final *YUV* video. This final *YUV* video can be compared with the original *YUV* video in order to measure the *PSNR* for evaluating the end-to-end delivered video quality. Furthermore, others performance metrics such as the packet loss rate and end-to-end delay, can be obtained from the *NS-2 Traffic traces* generated during network simulation.

## 5 Performance evaluation

In this section, we first provide the simulation settings for the performance analysis and then we analyse the simulations results. The main objective of our simulation study is to demonstrate the effectiveness of *SVCEval-RA* in the transmission of *Adaptive SVC* video over different network scenarios. Two video streaming strategies were evaluated in order to demonstrate the usefulness of our proposed framework. On one hand, a video sequence was transmitted according with the *Adaptive SVC* scheme of the *SVCEval-RA* platform. On the other hand, using the *Non-Adaptive SVC* scheme of the *myEvalSVC* framework, the same video sequence was also transmitted over the simulated scenarios. The two *SVC* streams were evaluated in terms of throughput of the video flow, the lost packet rate and the *PSNR*.

### 5.1 The simulation environment

The network simulator NS-2 has been used to test the performance of our framework. The network scenarios considered in our evaluation are depicted in Fig. 6. These scenarios provide persistent fluctuations of the available bandwidth in order to assess the rate adaptive mechanisms of *SVCEval-RA* during a streaming of a video sequence.

The first network scenario (described in Fig. 6a) has two parts: from $t = 10$ s to 50 s, the available bandwidth is decreased at intervals of 10 s from 1.5 Mbps to 100 Kbps. In the second part, the available bandwidth is increased from 100 Kbps to 1.5 Mbps at intervals of ten seconds. This network scenario evaluates the behaviour of the video source to abrupt fluctuations in the available bandwidth. Such variations are common in Internet data communication for instance, when the cross traffic in the bottleneck of the path varies significantly on account of arriving or departing traffic from other users [1].
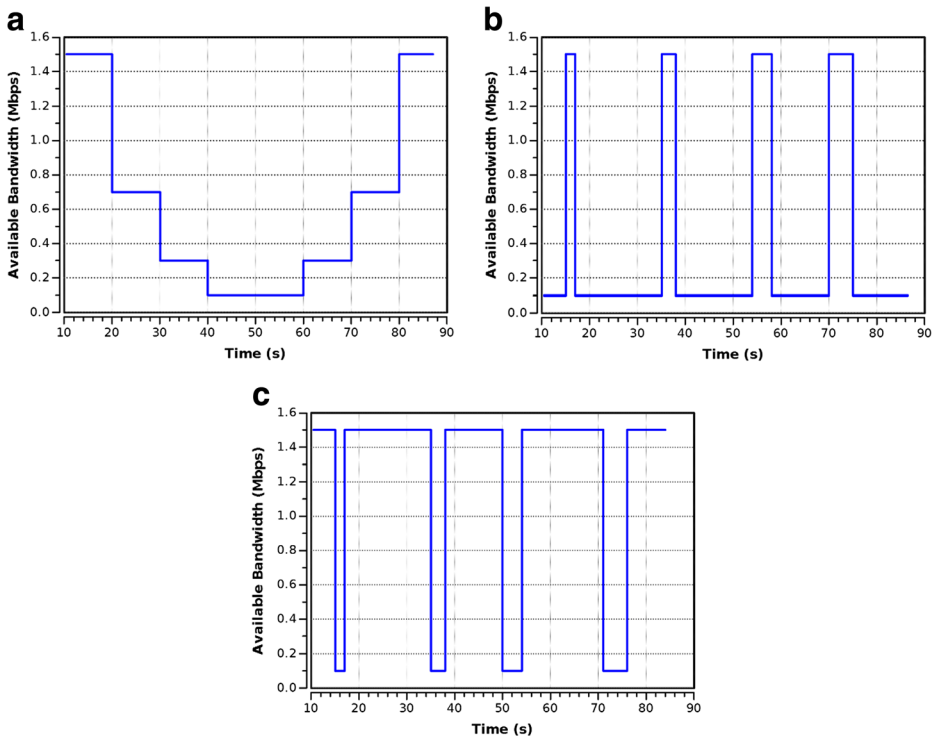
Fig. 6 Network scenarios. Scenario 1: Persistent variations of the available bandwidth (**a**), Scenario 2: "Positive" peaks in the available bandwidth (**b**) and Scenario 3: "Negative" peaks in the available bandwidth (**c**)

The second scenario (represented in Fig. 6b) supplies a constant low available bandwidth of 0.1 Mbps with high bandwidth spikes of 1.5 Mbps. The duration of these spikes is increased over time from 1.5 s to 5 s. This scenario aims to measure the reaction time of the video source upon extreme variations of the available bandwidth.

The third network scenario presents the opposite situation of the second scenario. Scenario 3 supplies a continuous high available bandwidth of 1.5 Mbps with low bandwidth troughs of 100 Kbps as shown in Fig. 6c. Duration of these troughs is also increased over time, the first trough lasts 1.5 s increasing up to 5 s in the last one.

The available bandwidth variations of the scenarios 2 and 3 are commons in practice, especially in 802.11 wireless networks [1]. These scenarios are intended to validate the performance of *SVCEval-RA* and the behaviour of the *adaptive SVC scheme* implemented in the SVCEval-RA Agents with more persistent and short-term bandwidth fluctuations compared with the scenario 1.

As far as video source is concerned, we concatenate the well-known video sequences "*HIGHWAY*" and "*HALL*" [44] with *CIF* resolution to form a basic testing video of 2300 frames. The video sequence has been encoded according to *H.264/SVC* standard with two types of scalability: temporal and quality. As *SVC* codec, the *JSVM* codec [26] was used. All the values for the video related parameters are reported in Table 2. The video sequence was encoded in five temporal layers (from $T_0$ to $T_4$). At the same time, we can add up to three extra levels of quality scalability (from $Q_0$ to $Q_3$) at each temporal level. For quality scalability we use *MGS* layers. The use of *MGS* layers for quality scalability allows source video to discard the data units from the enhancement layers without affecting the result bit-stream. In total, we obtained twenty video

**Table 2** Video parameters

| Parameters | Description/Value |
| --- | --- |
| Original video file | YUV format |
| Size | 2300 frames |
| Frame per second | 30 |
| Encoded video file | H.264/SVC |
| Type of scalability | SNR (MGS) |
| B-frames | Yes |
| GOP size | 16 frames |

layers (from $L_0$ to $L_{19}$) from the combination of layers ($T_i$ $Q_j$). Moreover, depending of the number of layers transmitted, the output bit rate varies from 60.1 kbps (sending layer 0 alone) to 1.23 Mbps (sending layers 0 to 19). The bit rates associated to each layer are shown in Table 3. These values are aggregated, which means that to transmit Layer 3 we also have to transmit the dependent lower layers, i.e. 0, 1, and 2. Therefore, the total bandwidth required would be of 136.3 Kbps.

Figure 7 gives a graphical description of the bit rates obtained according to the temporal levels and the *MGS* layers. Moreover, a rate-distortion analysis in terms of average *Y-PSNR* (*PSNR* for the luminance component in the *YUV* colour space) versus average bit rate was computed off-line (see Fig. 8). The computation of the *Y-PSNR* curves were performed by stripping out the layers, measuring the average bit rate, decoding the resulting video, and computing the average *Y-PSNR*. Each of these curves represents a temporal layer and each point corresponds to a *MGS* layer (from

**Table 3** Description of the video layers of the coded sequence

| Layer ID | Ti Qi | Frame rate | Aggregated Bit rate (Kbps) | PSNR (dB) |
| --- | --- | --- | --- | --- |
| 0 | $T_0$ $Q_0$ | 1.875 | 60.1 | 22.5 |
| 1 | $T_1$ $Q_0$ | 3.75 | 80.5 | 24.7 |
| 2 | $T_2$ $Q_0$ | 7.5 | 105.6 | 27.8 |
| 3 | $T_3$ $Q_0$ | 15 | 136.3 | 31.3 |
| 4 | $T_4$ $Q_0$ | 30 | 170.0 | 37.5 |
| 5 | $T_0$ $Q_1$ | 1.875 | 94.9 | 22.9 |
| 6 | $T_0$ $Q_2$ | 1.875 | 224.4 | 23.6 |
| 7 | $T_0$ $Q_3$ | 1.875 | 365.3 | 24.1 |
| 8 | $T_1$ $Q_1$ | 3.75 | 130.7 | 25.1 |
| 9 | $T_1$ $Q_2$ | 3.75 | 307.4 | 26.3 |
| 10 | $T_1$ $Q_3$ | 3.75 | 485.2 | 27.0 |
| 11 | $T_2$ $Q_1$ | 7.5 | 182.3 | 28.4 |
| 12 | $T_2$ $Q_2$ | 7.5 | 430.4 | 29.4 |
| 13 | $T_2$ $Q_3$ | 7.5 | 660.9 | 30.6 |
| 14 | $T_3$ $Q_1$ | 15 | 252.9 | 31.8 |
| 15 | $T_3$ $Q_2$ | 15 | 639.1 | 32.7 |
| 16 | $T_3$ $Q_3$ | 15 | 958.1 | 34.0 |
| 17 | $T_4$ $Q_1$ | 30 | 331.1 | 38.0 |
| 18 | $T_4$ $Q_2$ | 30 | 836.5 | 39.4 |
| 19 | $T_4$ $Q_3$ | 30 | 1231.4 | 41.0 |

**Fig. 7** Description of the video layers
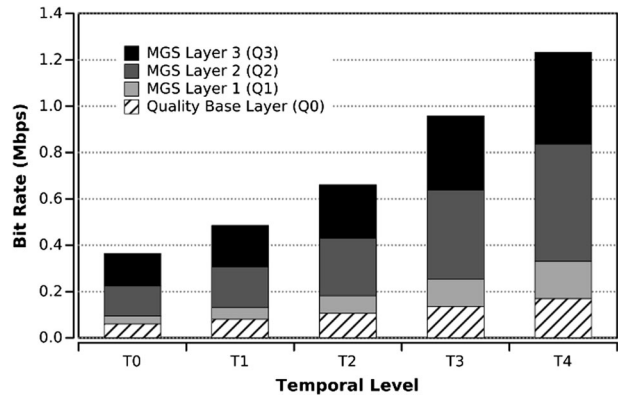


$Q_0$ to $Q_3$). This figure describes the increase in the video quality (in terms of Y-PSNR) depending on the number of quality and temporal layers that make up the video.

## 5.2 Performance metrics

Several objective metrics to estimate the video quality have been defined in the literature [7]. In this paper, we used the Peak Signal-to-Noise Ratio (*PSNR*) and *SSIM* (Structural Similarity Index) [42] for evaluating the video quality. While *PSNR* has a simple calculation and clear physical meanings, *SSIM* index has a good correlation with perceived video quality. To correctly estimate the *PSNR* and *SSIM*, it is necessary to establish a synchronization mechanism between sent and received frames. This is required since the frames may be lost during transmission due to congestion, or because the adaptive rate controller considered appropriate not to send some layers. For example, when video source only transmits the layer 0 ($T_0$ $Q_0$), the computation of the *PSNR* and *SSIM* must take into account the lack of the frames belonging to other layers. Otherwise, the calculation would not be correct since we are comparing frames between source and destination that do not match each other. Figure 9a shows a basic example where the sent and received frames are not synchronized and therefore the *PSNR* calculation is wrong. The correct method (shown in Fig. 9b) involves duplicating the frames of the lower layer so that both sequences have the same number of frames.

**Fig. 8** Video properties used in the simulation study
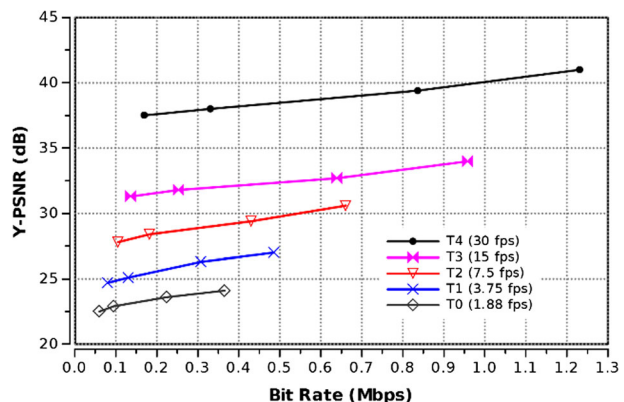
**Fig. 9** Synchronization of frames during the PSNR and SSIM Index Calculation
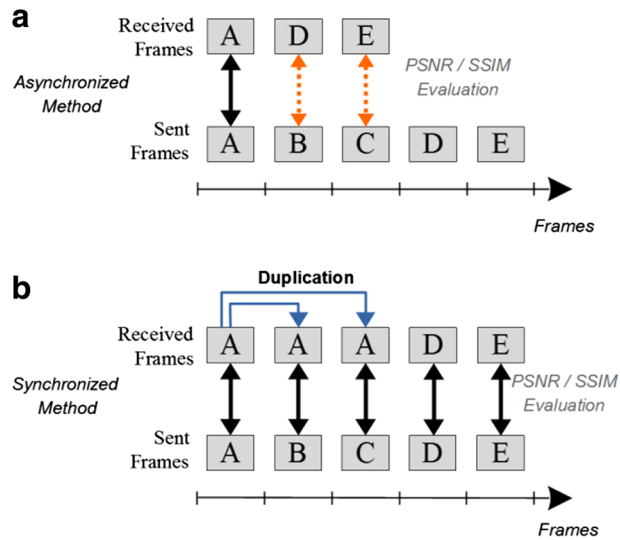


Others metrics used in this study are the maximum throughput achieved during transmission and lost packet rate. The measured throughput will provide a perspective on how the bit rate adaptation is performed and the lost packet rate will be an indicator of network congestion.

# 6 Results

In this section, we describe the experimental results obtained in the simulation study. Each point in the curves represents the average over 10 simulations and a confidence level of 95 %. As mentioned above, two video streaming strategies were used: *Adaptive SVC* and *Non-adaptive SVC*, in order to demonstrate the usefulness of our proposed framework.

## 6.1 Scenario 1: persistent variations of the available bandwidth

The results obtained for the scenario 1 are shown in Fig. 10. In addition, the available bandwidth fluctuation provided by the scenario 1 is also plotted. Figure 10a shows the throughput achieved during the video transmission. The lost packet rate, the luminance *PSNR* (*Y-PSNR*) and the *SSIM* index are shown in the Fig. 10b, c and d, respectively. On the other hand, Table 4 summarizes mean values of the evaluated metrics for each segment in which a bandwidth variation was performed. In addition, the layers that were transmitted for each segment are also described in the column two, where *BL* means Base Layer and the combination of the *T* and *Q* parameters identifies the temporal and *MGS* layer, respectively.

As seen in Fig. 10a, during the intervals from 10 s to 20 s and from 80 s to 86 s all *SVC* layers of the video sequence can be transmitted. This fact is explained because the rate required to send the highest layer (1.23 Mbps for layer 19, see Table 3) and its dependent lower layers can be supported by the link capacity. Consequently, all packets can be successfully transmitted (lost packet rate equal to 0, see Fig. 10b) and therefore the quality of the received video is the maximum that can be achieved (Fig. 10c and d). However, as can be seen in Fig. 10a, the available bandwidth decreases and this fact creates the need to adjust the transmission rate of
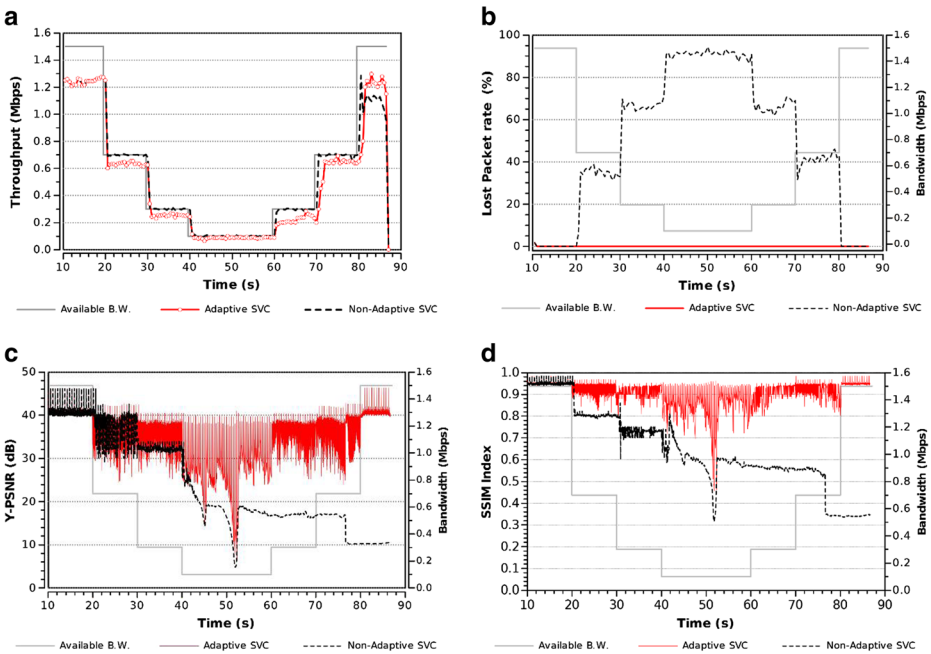
**Fig. 10** Throughput of the video flows (**a**), Lost Packet Rate (**b**), Y-PSNR (**c**) and SSIM Index (**d**) for Scenario 1

the video application in order to avoid the network congestion. For instance, for the intervals 30–40 s and 60–70 s the available bandwidth is 300 Kbps and the video source maximum can transmit the layer $T_3Q_1$ and its dependent lower layers (see Table 4). Because of this, the 100 % of the packets are successfully transmitted but the video is decoded with a lower quality (an average *Y-PSNR* of 35 dB and *SSIM* of 0.91 for these intervals).

When a video source does not adapt its data rate (*Non-Adaptive SVC* curves in Fig. 10) some packets are lost. In the case of the segments from 30 to 40 s and from 60 to 70 s, the percentages of lost packets in these intervals are of 65 % and 67 % respectively. This high rate of dropped packets significantly affects the quality of the received video.

**Table 4** Average values of the results for the scenario 1

| Time interval (s) | SVC layer using Adaptive SVC | Lost packet rate (%) | | Y-PSNR (dB) | | SSIM index | |
|---|---|---|---|---|---|---|---|
| | | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC |
| 10–20 | $BL + T_4Q_3$ | 0.1 | 0.1 | 40.8 | 40.8 | 0.95 | 0.95 |
| 20–30 | $BL + T_3Q_2$ | 0.0 | 31.9 | 35.8 | 36.4 | 0.92 | 0.81 |
| 30–40 | $BL + T_3Q_1$ | 0.0 | 64.7 | 34.9 | 32.2 | 0.91 | 0.73 |
| 40–60 | $BL + T_1Q_0$ | 0.0 | 90.5 | 27.5 | 18.3 | 0.84 | 0.60 |
| 60–70 | $BL + T_3Q_1$ | 0.0 | 67.2 | 35.1 | 16.8 | 0.91 | 0.56 |
| 70–80 | $BL + T_3Q_2$ | 0.0 | 42.4 | 35.8 | 14.7 | 0.92 | 0.48 |
| 80–86 | $BL + T_4Q_3$ | 0.0 | 2.9 | 40.9 | 10.3 | 0.95 | 0.34 |

When the available bandwidth drastically decrease to 100 Kbps (interval from 40 s to 60 s) the layer $T_1Q_0$ is the highest layer that can be transmitted (see Table 4), according to the video properties described in the section 5.1 (see Fig. 8). As we can see in the Fig. 10a and b, the mechanisms of our framework allow the video source to adjust its bit rate avoiding network congestion. Thus, there were no loss packets during this segment and therefore the average *Y-PSNR* obtained is about 27.5 dB (see Fig. 10c) and the *SSIM* index is 0.84. In contrast, using *Non-Adaptive SVC*, the percentage of packet loss reaches the 90 %. Therefore, the quality of the received video drastically decrease and an average *Y-PSNR* of 18 dB is obtained (about 11 dB of difference with *Adaptive SVC*). Similarly, the *SSIM* index has a decrease and it is 28 % lower than the *SSIM* obtained by adapting the *SVC* stream.

Note that, the high loss packet rate from time instant 40 s significantly affect the decoding of the frames transmitted subsequently. In addition, the frames transmitted near the time instant 52 s have a significant decrease in their *PSNR* values due to the high level of motion that these frames have, such as is explained in Fig. 11. This figure outlines the motion variance over time for the test video sequence. The motion level is calculated using Sum of Absolute Difference (*SAD*) between the two frames and it is given by Eq. (1).

$$\text{SAD}_{n,m} = \sum_{i=1}^{N} \sum_{j=1}^{M} |B_n(i, j) - B_m(i, j)| \tag{1}$$

Where $B_n$ and $B_m$ are the two frames of size $N \times M$, and $i$ and $j$ denote pixel coordinates.

As seen in Fig. 11, the video sequence from frame 1 to frame 1240 is very similar. However, the spatio-temporal complexity of the frames 1240–1270 is incremented because of the changes in the foreground of the scene. Similarly, there is a high value of the *SAD* in the frame 2000 because of the abrupt transition that occurs when the "HIGHWAY" video ends and starts the concatenated video sequence ("HALL" video).

## 6.2 Scenario 2: "Positive" peaks in the available bandwidth

The second scenario was intended to test the behaviour of the adaptation algorithms of *SVCEval-RA* under high bandwidth peaks of different durations. Figure 12 represents the performance of the *SVCEval-RA* platform in terms of throughput, lost packet rate, *Y-PSNR* and *SSIM* when the network scenario produce short-term variations of the available bandwidth. In the Table 5 is described the transmitted layers and the average values of the results for each intervals time.
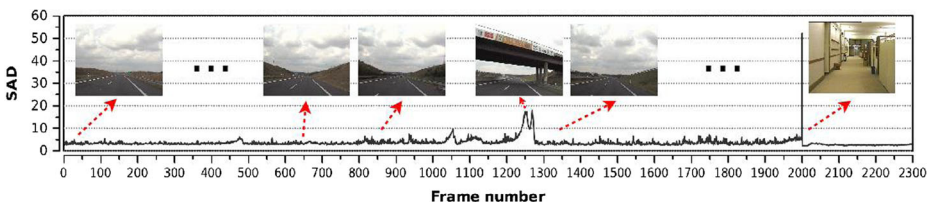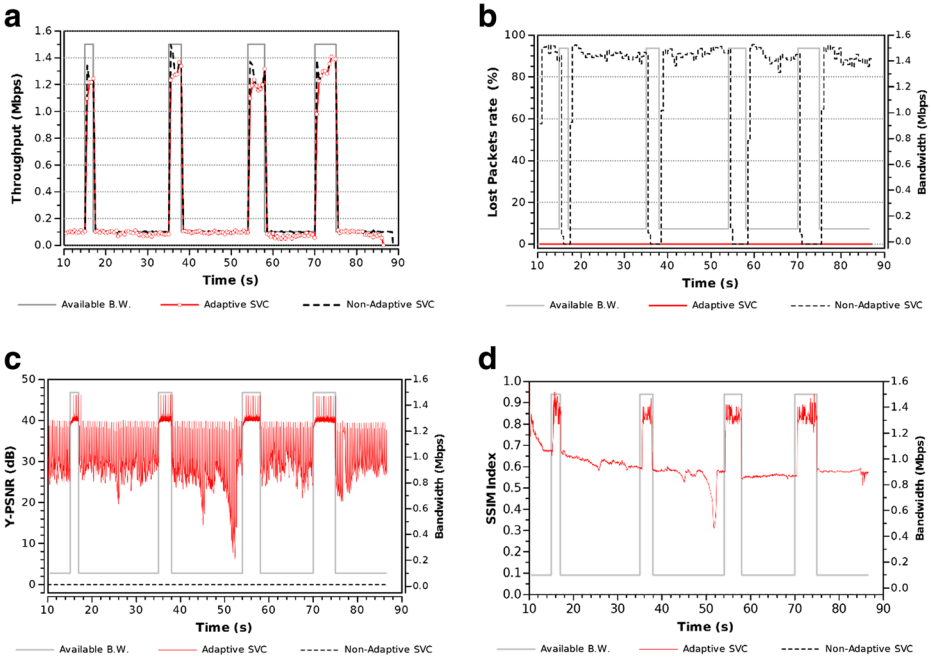


**Fig. 11** SAD estimation of the test sequence

**Fig. 12** Throughput of the video flows (**a**), Lost Packet Rate (**b**), Y-PSNR (**c**) and SSIM Index (**d**) for Scenario 2 ("Positive" peaks in the available bandwidth)

The results demonstrate that the rate-adaptive scheme implemented in the *SVCEval-RA Traffic Agent* has a significant rapid response to bandwidth fluctuations. Consequently, the throughput achieved using the *Adaptive SVC scheme* is close to the throughput achieved by the non-adaptive scheme. This rapid reaction is illustrated at $t = 15$ s when there was a sudden increase in the bandwidth and the adaptation of the bit rate was immediately performed by the *Traffic Agent* (Fig. 12a). Although the duration of the first bandwidth peak is only two seconds, the *Adaptive SVC* flow can rapidly adapt its bit rate, achieving a similar throughput

**Table 5** Average values of the results for the scenario 2

| Time interval (s) | SVC layer using Adaptive SVC | Lost packet rate (%) | | Y-PSNR (dB) | | SSIM index | |
|---|---|---|---|---|---|---|---|
| | | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC |
| 10–15 | $BL + T_1Q_0$ | 0.00 | 89.07 | 29.61 | 0.00 | 0.73 | 0.00 |
| 15–17 | $BL + T_4Q_3$ | 0.00 | 23.47 | 39.84 | 0.00 | 0.84 | 0.00 |
| 17–35 | $BL + T_1Q_0$ | 0.00 | 89.77 | 29.74 | 0.00 | 0.63 | 0.00 |
| 35–38 | $BL + T_4Q_3$ | 0.00 | 16.07 | 40.29 | 0.00 | 0.78 | 0.00 |
| 38–54 | $BL + T_1Q_0$ | 0.00 | 87.78 | 26.73 | 0.00 | 0.56 | 0.00 |
| 54–58 | $BL + T_4Q_3$ | 0.00 | 12.62 | 40.51 | 0.00 | 0.82 | 0.00 |
| 58–70 | $BL + T_1Q_0$ | 0.00 | 84.51 | 29.91 | 0.00 | 0.56 | 0.00 |
| 70–75 | $BL + T_4Q_3$ | 0.00 | 9.57 | 40.29 | 0.00 | 0.81 | 0.00 |
| 75–86 | $BL + T_1Q_0$ | 0.00 | 84.78 | 29.96 | 0.00 | 0.58 | 0.00 |

to the one achieved by the non-adaptive flow, which always is been transmitted with all *SVC* layers (i.e. to the maximum bit rate). Adaptation was also successfully performed in the following three peaks. Although at the start time of the bandwidth peaks (time instant *t* = 15, 35, 54 and 70 s) the rate adaptive scheme of our framework have a conservative behaviour since they perform a slow increasing of the bit rate, this behaviour helps to avoid a buffer overload when an abrupt increasing of the available bandwidth is occurred. On the other hand, regarding the duration of the peaks, a significant influence has not been found in this scenario since the algorithms of the *SVCEval-RA Traffic Agent* allowed the video source to adapt its bit rate during the shorter as well as the longer peaks.

Similarly to the scenario 1, the accurate adaptation of the data rate using *Adaptive SVC* scheme avoid congestion during the intervals in which the available bandwidth is more restrictive (see Fig. 12b). Therefore, the packet loss remains equal to 0 in the case of the *Adaptive SVC* flow and the average *Y-PSNR* in theses intervals is about 29 dB. On the contrary, when the non-adaptive scheme is used the high percentage of packet loss (about 90 % during the intervals of lower available bandwidth) significantly impact the number of decodable frames and therefore, received video cannot be successfully decoded (it is illustrated as *Y-PSNR* and *SSIM* equal to cero in Fig. 12c and d).

### 6.3 Scenario 3: "Negative" peaks in the available bandwidth

The third scenario was intended to test the behaviour of the adaptation algorithms under low bandwidth-peaks of different durations. Figure 13 illustrated the behaviour
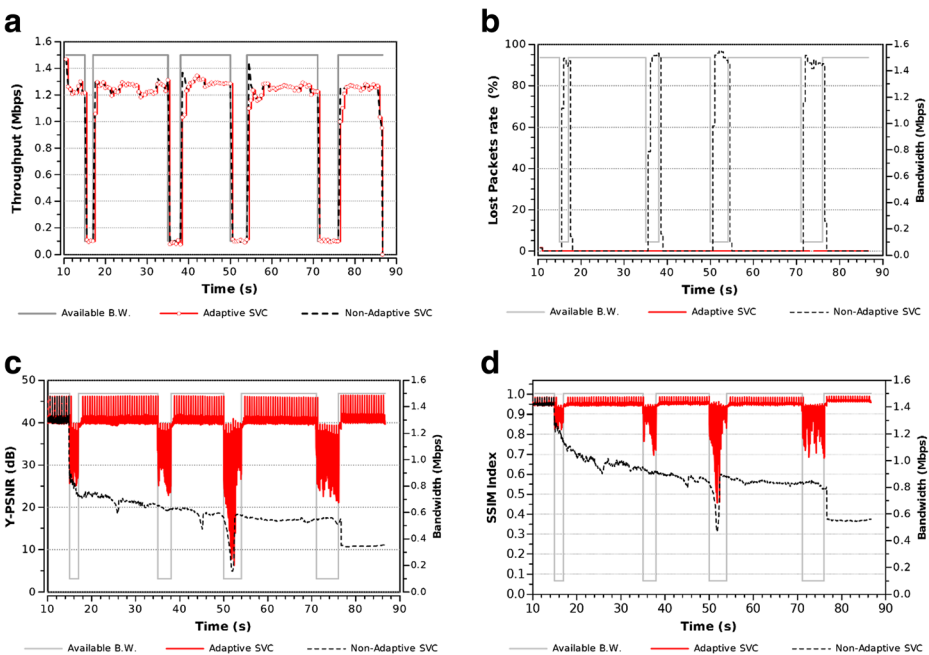


**Fig. 13** Throughput of the video flows (**a**), Lost Packet Rate (**b**), Y-PSNR (**c**) and SSIM Index (**d**) for Scenario 3 ("Negative" peaks in the available bandwidth)

of the *adaptive SVC scheme* of *SVCEval-RA* and the Table 6 summarizes the mean values of the results as well as the layers sent for each segment in which has been performed a variation of the available bandwidth.

The results of the scenario 3 are consistent with the results obtained in the previous network scenarios. In all cases, when a variation of the available bandwidth was performed, the rate adaptive algorithms of the *SVCEval-RA Traffic Agent* allow the video source to adjust its bit rate transmitting only the *SVC* layers that can be supported avoiding network congestion. In Table 6 we can see that when the available bandwidth is high (about 1.2 Mbps) the layers transmitted are the base layer and the $T_4Q_3$ layer whereas when the available bandwidth decreases to 0.1 Mbps only the base and $T_1Q_0$ layers are transmitted. This adaptation allows us to get an important decrease in packet loss (Fig. 13b) while it is makes more efficient use of the available bandwidth. Therefore, the quality of the received video is better than the quality achieved when the *Non-Adaptive SVC* scheme is used. The average *Y-PSNR* of the received video is 35.1 dB using adaptive techniques, about 15 dB of difference that using non adaptive scheme. Similarly, an improvement of 30 % is obtained in the perceived video quality of the adaptive stream with an average value for *SSIM* of 0.9. In fact, using *Non-Adaptive SVC*, the high rate of dropped packets during the first trough (with a duration of only 2 s) drastically affect the final decoding of the video (see Fig. 13c). Regarding the effects of the duration of the troughs, has not been found an influence over the efficacy of the algorithms implemented in the *Traffic Agent* of *SVCEval-RA*. In all case, the *Traffic Agent* provides a rapid response to adapt the transmission rate to abrupt decrease of the available bandwidth even though the duration of the troughs is short, such as is the case of the first "negative" peak.

In general terms, the obtained results prove that our framework provides efficient network-adaptive algorithms to perform realistic network simulations that involve adaptive video streaming supporting SVC coding. The *Adaptive SVC scheme* presented in our proposed framework allows the traffic source to transmit only the *SVC* layers that can be efficiently supported by network. This fact provides better conditions to video streaming as demonstrated by the *Y-PSNR* obtained in the simulation study. Using our *SVCEval-RA* platform a feedback about the current network status is provided to the source application in order to set the layers that can be transmitted. Without this information, the video may not be adapted, causing congestion in the network and a large number of dropped packets.

**Table 6** Average values of the results for the scenario 3

| Time interval (s) | SVC layer using Adaptive SVC | Lost packet rate (%) | | Y-PSNR (dB) | | SSIM index | |
|---|---|---|---|---|---|---|---|
| | | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC | Adaptive SVC | Non-adaptive SVC |
| 10–15 | $BL + T_4Q_3$ | 0.2 | 0.2 | 40.7 | 40.5 | 0.95 | 0.95 |
| 15–17 | $BL + T_1Q_0$ | 0.0 | 63.0 | 30.5 | 25.2 | 0.88 | 0.81 |
| 17–35 | $BL + T_4Q_3$ | 0.0 | 2.8 | 40.7 | 21.7 | 0.95 | 0.67 |
| 35–38 | $BL + T_1Q_0$ | 0.0 | 70.8 | 29.4 | 19.7 | 0.84 | 0.61 |
| 38–50 | $BL + T_4Q_3$ | 0.0 | 4.2 | 40.7 | 18.7 | 0.95 | 0.58 |
| 50–54 | $BL + T_1Q_0$ | 0.0 | 78.8 | 23.8 | 13.8 | 0.79 | 0.51 |
| 54–71 | $BL + T_4Q_3$ | 0.0 | 2.8 | 40.6 | 17.1 | 0.95 | 0.56 |
| 71–76 | $BL + T_1Q_0$ | 0.0 | 80.0 | 28.5 | 17.2 | 0.85 | 0.55 |
| 76–86 | $BL + T_4Q_3$ | 0.0 | 4.6 | 40.6 | 11.2 | 0.96 | 0.38 |

# 7 Conclusions

In this paper we have introduced *SVCEval-RA*, a novel framework to perform network simulations that involve rate-adaptive video transmission using scalable video coding (*SVC*).

We have explained the main components of the *SVCEval-RA* framework as well as its interactions with the NS-2 simulator. Basically, our framework works by generating two video traces from a real *H.264/SVC* encoded video. The traces are used as input to the network simulator in order to provide information about the packetization and transmission of the video frames. Then, the output of the simulation is utilized to produce the received video stream.

In order to demonstrate the usefulness of our proposed framework, we performed a simulation study where two video streaming strategies were evaluated: *Adaptive SVC* and *Non-Adaptive SVC*. This study was carried out over three network scenarios with the aim of explain how our approach can be used as an evaluation platform for evaluation of video transmission.

The obtained results prove that the algorithms implemented in *SVCEval-RA* allow the video source to dynamically adjust its data rate by selecting, from the original video, a set of *SVC* layers to be transmitted based on the estimation of the available bandwidth. This adaptive scheme avoids an increase in the percentage of packet loss due to network congestion. In addition, the performance evaluation of the *Adaptive SVC scheme* implemented in the *SVCEval-RA Traffic Agent* reveals that it has a rapid reaction when abrupt fluctuations of the available bandwidth occur.

In terms of video quality, the measured *PSNR* and SSIM values of the test videos show that the *Adaptive SVC* technique is able to increase the quality of the received stream compared to *Non-Adaptive SVC* scheme. Moreover, the resulting video quality has been significantly better in those scenarios where the available bandwidth was more constrained.

More advance usage of *SVCEval-RA* includes the transmission of video sequences in more realistic network topologies using other network-adaptive media technologies, such as *DASH*, *TFRC* or *RTMP*.

The videos obtained in the present validation study of *SVCEval-RA* over the three network scenarios can be displayed from the website *http://www.comm.upv.es/svceval_ra/svceval_ra.htm* and the latest version of *SVCEVal-RA* is available for free download at [4].

# References

1. Akhshabi S, Begen AC, Dovrolis C (2011) An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In: Proceedings of the second annual ACM conference on Multimedia systems. ACM, pp 157–168
2. Alabdulkarim MN, Rikli N-E (2012) QoS Provisioning for H.264/SVC Streams over Ad-Hoc ZigBee Networks Using Cross-Layer Design. In: 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM). pp 1–8
3. Birkos K, Tselios C, Dagiuklas T, Kotsopoulos S (2013) Peer selection and scheduling of H. 264 SVC video over wireless networks. In: Wireless Communications and Networking Conference (WCNC), 2013 IEEE. pp 1633–1638
4. Castellanos W (2014) SVCEval-RA - An Evaluation Framework for Adaptive Scalable Video Streaming. In: SourceForge Project. http://sourceforge.net/projects/svceval-ra/. Accessed 1 May 2015

5. Castellanos W, Guerri JC, Arce P (2015) A QoS-aware routing protocol with adaptive feedback scheme for video streaming for mobile networks. Comput Commun. http://dx.doi.org/10.1016/j.comcom.2015.08.012

6. Castellanos W, Arce P, Acelas P, Guerri JC (2012) Route Recovery Algorithm for QoS-Aware Routing in MANETs. Springer Berlin Heidelberg, Bilbao, pp. 81–93

7. Chikkerur S, Sundaram V, Reisslein M, Karam LJ (2011) Objective video quality assessment methods: A classification, review, and performance comparison. Broadcast, IEEE Trans on 57:165–182

8. Choupani R, Wong S, Tolun M (2014) Multiple description coding for SNR scalable video transmission over unreliable networks. Multimed Tools Appl 69:843–858. doi:10.1007/s11042-012-1150-9

9. CISCO Corp. (2014) Cisco Visual Networking Index Forecast and Methodology. In: White Paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf.

10. Dai M, Zhang Y, Loguinov D (2009) A unified traffic model for MPEG-4 and H. 264 video traces. IEEE Trans Multimedia 11:1010–1023

11. Detti A, Bianchi G, Pisa C, et al. (2009) SVEF: an open-source experimental evaluation framework for H.264 scalable video streaming. In: IEEE Symposium on Computers and Communications. pp 36–41

12. Espina F, Morato D, Izal M, Magaña E (2014) Analytical model for MPEG video frame loss rates and playback interruptions on packet networks. Multimed Tools Appl 72:361–383. doi:10.1007/s11042-012-1344-1

13. Fiems D, Steyaert B, Bruneel H (2012) A genetic approach to Markovian characterisation of H.264 scalable video. Multimedia Tools Appl 58:125–146

14. Floyd S, Handley M, Kohler E Datagram Congestion Control Protocol (DCCP). http://tools.ietf.org/html/rfc4340. Accessed 17 Feb 2014

15. Floyd S, Padhye J, Widmer J TCP Friendly Rate Control (TFRC): Protocol Specification. http://tools.ietf.org/html/rfc5348. Accessed 17 Feb 2014

16. Fraz M, Malkani YA, Elahi MA (2009) Design and implementation of real time video streaming and ROI transmission system using RTP on an embedded digital signal processing (DSP) platform. In: 2nd International Conference on Computer, Control and Communication, 2009. IC4 2009. pp 1–6

17. ISO/IEC (2014) Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats.

18. ITU-T (2013) Rec. H.264 & ISO/IEC 14496-10 AVC. Advanced Video Coding for Generic Audiovisual Services.

19. Ivrlač MT, Choi LU, Steinbach E, Nossek JA (2009) Models and analysis of streaming video transmission over wireless fading channels. Signal Process Image Commun 24:651–665. doi:10.1016/j.image.2009.04.005

20. Karki R, Seenivasan T, Claypool M, Kinicki R (2010) Performance Analysis of Home Streaming Video Using Orb. In: Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video. ACM, New York, NY, USA, pp 111–116

21. Ke C-H (2012) myEvalSVC-an Integrated Simulation Framework for Evaluation of H. 264/SVC Transmission. KSII Trans Internet Inf Syst (TIIS) 6:377–392. doi:10.3837/tiis.2012.01.021

22. Ke C-H, Shieh C-K, Hwang W-S, Ziviani A (2008) An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission. J Inf Sci Eng 24:425–440

23. Klaue J, Rathke B, Wolisz A (2003) Evalvid–A framework for video transmission and quality evaluation. In: Computer Performance Evaluation. Modelling Techniques and Tools. Springer, pp 255–272

24. Le TA, Nguyen H (2014) End-to-end transmission of scalable video contents: performance evaluation over EvalSVC—a new open-source evaluation platform. Multimed Tools Appl 72: 1239–1256. doi:10.1007/s11042-013-1444-6

25. Lie A, Klaue J (2008) Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video. Multimedia Systems 14:33–50. doi:10.1007/s00530-007-0110-0

26. Moving Pictures Experts Group and ITU-T Video Coding Experts Group (2011) H. 264/SVC reference software (JSVM 9.19.14) and Manual.

27. Nightingale J, Wang Q, Grecos C (2014) Empirical evaluation of H.264/SVC streaming in resource-constrained multihomed mobile networks. Multimed Tools Appl 70:2011–2035. doi:10.1007/s11042-012-1219-5

28. Parmar H, Thornburgh M (2012) Real-Time Messaging Protocol (RTMP) Specification. Adobe

29. Politis I, Dounis L, Dagiuklas T (2012) H. 264/SVC vs. H. 264/AVC video quality comparison under QoE-driven seamless handoff. Signal Process Image Commun 27:814–826

30. Pozueco L, Pañeda XG, García R, et al. (2013) Adaptable system based on Scalable Video Coding for high-quality video service. Comput Electr Eng 39:775–789. doi:10.1016/j.compeleceng.2013.01.015

31. Pozueco L, Pañeda XG, García R, et al. (2014) Adaptation engine for a streaming service based on MPEG-DASH. Multimed Tools Appl 1–20. doi:10.1007/s11042-014-2034-y
32. Schwarz H, Marpe D, Wiegand T (2007) Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Trans Circ Syst Video Technol 17:1103–1120. doi:10.1109/TCSVT.2007.905532
33. Seo H-Y (2013) An Efficient Transmission Scheme of MPEG2-TS over RTP for a Hybrid DMB System. ETRI J 35:655–665. doi:10.4218/etrij.13.0112.0124
34. Sohn H, Yoo H, De Neve W, et al. (2010) Full-Reference Video Quality Metric for Fully Scalable and Mobile SVC Content. IEEE Trans Broadcast 56:269–280. doi:10.1109/TBC.2010.2050628
35. Sousa-Vieira M-E (2011) Suitability of the M/G/∞ process for modeling scalable H.264 video traffic. In: Analytical and Stochastic Modeling Techniques and Applications. Springer, pp 149–158
36. Tanwir S, Perros H (2013) A Survey of VBR Video Traffic Models. IEEE Commun Surv Tutor 15:1778–1802. doi:10.1109/SURV.2013.010413.00071
37. Tanwir S, Perros HG (2014) VBR Video Traffic Models. Wiley, Hoboken
38. The Network Simulator (NS-2). http://www.isi.edu/nsnam/ns. Accessed 6 Feb 2015
39. Unanue I, Urteaga I, Husemann R, et al. (2011) A Tutorial on H. 264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance. Recent Advances on Video Coding 1–24.
40. Van der Auwera G, David PT, Reisslein M, Karam LJ (2008) Traffic and quality characterization of the H. 264/AVC scalable video coding extension. Adv Multimedia 2008:1
41. Wang Y, Claypool M (2005) RealTracer—Tools for Measuring the Performance of RealVideo on the Internet. Multimed Tools Appl 27:411–430. doi:10.1007/s11042-005-3757-6
42. Wang Z, Lu L, Bovik AC (2004) Video quality assessment based on structural distortion measurement. Signal Process Image Commun 19:121–132. doi: 10.1016/S0923-5965(03)00076–6
43. Wien M, Schwarz H, Oelbaum T (2007) Performance Analysis of SVC. IEEE Trans Circ Syst for Video Technol 17:1194–1203. doi:10.1109/TCSVT.2007.905530
44. YUV video repository. ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/. Accessed 10 Jan 2013

**Wilder E. Castellanos** received his Electronic Engineering degree from Industrial University of Santander, Colombia, in 2002 and the M.Sc. degree in Telematics Engineering from the Universitat Politècnica de València (UPV), Spain, in 2009. He obtained his Ph.D. in Telecommunications in 2015 from the UPV. Since 2009 he has been with the Multimedia Communications research group (COMM) of the Institute of Telecommunications and Multimedia Applications (iTEAM). His current research interests include QoS-aware architectures, multimedia streaming in mesh/ad hoc networks and network performance evaluation.

**Juan Carlos Guerri** received his M.S. and Ph. D. (Dr. Ing.) degrees, both in Telecommunication Engineering, from the Polytechnic University of Valencia, in 1993 and 1997, respectively. He is a professor in the E.T.S. Telecommunications Engineering at the Polytechnic University of Valencia, and he leads Multimedia Communications research group (COMM) of the iTEAM (Institute of Telecommunications and Multimedia Applications). He is currently in research and development projects about multimedia QoS/QoE, wireless ad hoc networks and video performance evaluation.



**Pau Arce** was born in Valencia, Spain. He received the Telecommunications Engineering and Ph. D. (Dr. Ing.) degree, both in Telecommunication Engineering, from the Universitat Politècnica de València (UPV), Valencia, Spain, in 2005 and 2014, respectively. Currently, he works as a researcher at the Institute of Telecommunications and Multimedia Applications (iTEAM) UPV. His research interests include multimedia QoS, routing on wireless ad hoc networks and performance evaluation of computer systems