

Using MPEG-21 for cross-layer multimedia content adaptation

Ingo Kofler · Joachim Seidl · Christian Timmerer ·
Hermann Hellwagner · Ismail Djama · Toufik Ahmed

Received: 2 June 2008 / Revised: 22 August 2008 / Accepted: 12 September 2008 / Published online: 21 October 2008
© Springer-Verlag London Limited 2008

Abstract This paper presents a cross-layer model—formulated using interoperable description formats—for the adaptation of scalable H.264/MPEG-4 AVC (i.e., SVC) content in a video streaming system operating on a Wireless LAN access network without QoS mechanisms. SVC content adaptation on the server takes place on the application layer using an adaptation process compliant with the MPEG-21 Digital Item Adaptation (DIA) standard, based on input comprised of MPEG-21 DIA descriptions of content and usage environment parameters. The latter descriptions integrate information from different layers, e.g., device characteristics and packet loss rate, in an attempt to increase the interoperability of this cross-layer model, thus making it applicable to other models. For the sake of deriving model parameters, performance measurements from two wireless access point models were taken in account. Throughout the investigation it emerged that the behavior of the system strongly depends on the access point. Therefore, we investigated the use of end-to-end-based rate control algorithms for steering the content

adaptation. Simulations of rate adaptation algorithms were subsequently performed, leading to the conclusion that a TFRC-based adaptation technique (TCP-Friendly Rate Control) performs quite well in adapting to limited bandwidth and varying network conditions. In the paper we demonstrate how TFRC-based content adaptation can be realized using MPEG-21 tools.

Keywords Multimedia content adaptation · Cross-layer design · MPEG-21 Digital Item Adaptation · Rate control

1 Introduction

Cross-layer designs are gaining more and more momentum, partially in an attempt to enable end users to access multimedia services anywhere, anytime and on any kind of device, i.e., supporting the vision of Universal Multimedia Access (UMA). The aim of these cross-layer interactions is to address the tough requirements derived from such a scenario—specifically in wireless networks—in coping with different device characteristics (e.g., display capabilities) and network conditions (e.g., bandwidth, delay and packet loss).

The rigidity of the Internet Protocol suite with its well-defined layers and interfaces is one obstacle that hampers the deployment of multimedia services in wireless networks. Furthermore, the wide deployment of wireless networks has created demand for enhancing the performance of multimedia applications over such links. Cross-layer designs are currently an active research topic with the aim of increasing the Quality of Service/Experience (QoS/QoE) by performing coordinated actions across the network layers and, thus, in effect violating the protocol hierarchy and isolation model. As a result, a variety of different approaches have emerged over recent years [20, 43]. In the majority of these approaches,

I. Kofler (✉) · J. Seidl · C. Timmerer · H. Hellwagner
Institute of Information Technology (ITEC),
Klagenfurt University, Klagenfurt, Austria
e-mail: ingo.kofler@itec.uni-klu.ac.at

J. Seidl
e-mail: joachim.seidl@itec.uni-klu.ac.at

C. Timmerer
e-mail: christian.timmerer@itec.uni-klu.ac.at

H. Hellwagner
e-mail: hermann.hellwagner@itec.uni-klu.ac.at

I. Djama · T. Ahmed
CNRS-LaBRI Lab, University of Bordeaux-1,
Talence, France
e-mail: djama@labri.fr

T. Ahmed
e-mail: tad@labri.fr

the cross-layer interactions take place in either a bottom-up or a top-down fashion, where lower layers influence upper layers or vice versa. More recently, efforts have focused on tackling the problem by jointly optimizing the parameters of different layers [4].

However, regardless of the techniques used by the various cross-layer designs, they all share the common property of compromising interoperability in favor of performance. In this paper, we present an approach which aims to increase interoperability by investigating how interoperable description formats can be applied to cross-layer designs, specifically in the context of the adaptation of multimedia content. The MPEG-21 multimedia framework, which enables the transparent and augmented use of multimedia resources across a wide range of networks, devices, and user preferences, is a standard worthy of consideration [3]. In particular, Part 7 of MPEG-21, i.e., Digital Item Adaptation (DIA), provides a specification of normative description formats (known as tools in DIA) for describing both the properties of the multimedia content and the context of its usage [44].

Our approach is based on exchanging these description formats (which follow a well-defined cross-layer model) and using them to steer the continuous adaptation of the multimedia content to potentially varying usage environment conditions. The context properties are comprised of information from different networking layers whereas the content description provides the means to express the relationship between the context (i.e., its constraints) and possible adaptation operations—also influencing parameters at different networking layers—satisfying these constraints, such that good quality of service can be provided.

In order to validate our approach we implemented a prototypical MPEG-21-based streaming system and performed experiments utilizing our cross-layer model over a wireless network. As the systems' behavior strongly depends on the access points' capabilities, several end-to-end-based rate control algorithms for steering the content adaptation were investigated. In anticipation of the result we selected a TCP-Friendly Rate Control-based (TFRC) adaptation technique which is the basis for our cross-layer model. TFRC was chosen because it performs well in terms of TCP friendliness, responsiveness, smoothness, and also interoperability. To that end, we developed a cross-layer model for MPEG-4 Scalable Video Coding (SVC) that natively supports scalability in the spatial, temporal, and signal-to-noise ratio (SNR) domains [36]. In a second step, we show how such a cross-layer model can be formulated by utilizing interoperable description formats as standardized within MPEG-21 DIA. Finally, the third step is comprised of finding an optimal solution for the optimization problem at hand using a generic metadata-driven Adaptation Decision-Taking Engine (ADTE).

This paper is organized as follows. Section 2 briefly introduces the relevant tools of the MPEG-21 standard that are

utilized in the project. The current state of the art in cross-layer designs and work related to our approach is given in Sect. 3. In Sect. 4 our approach of using MPEG-21 for performing cross-layer content adaptation is presented. Furthermore, the architecture of our streaming environment that was used for our experiments is explained in detail. Section 5 discusses the streaming experiments with two different access points and the conclusions drawn from the results. These conclusions lead us to consider the use of rate control in the context of MPEG-21 in order to adapt the behavior of the sender according to network feedback. These considerations as well as some recent rate control algorithms are explained in Sect. 6. In order to evaluate the modifications to our approach, we implemented a rate control algorithm by using MPEG-21 tools and performed simulations. The details of the algorithm can be found in Sect. 7. The simulation setup and the results we obtained from the two different scenarios are discussed in Sect. 8. The results and findings of our investigations are summarized in Sect. 9, which concludes this paper.

2 MPEG-21 Tools for multimedia content adaptation

The aim of the MPEG-21 standard [3], the so-called multimedia framework, is to enable transparent and augmented use of multimedia resources across a wide range of networks, devices, user preferences and even natural environments. MPEG-21 introduces the concept of a Digital Item (DI) which is a structured digital object with a standard representation and metadata. Digital Items build the fundamental unit of all transactions and content distribution within the MPEG-21 framework. A vital and comprehensive part within MPEG-21 is Part 7 of the standard, referred to as DIA, which specifies normative description tools to assist the adaptation of Digital Items. A tool within MPEG-21 is defined as the XML-based syntax of a description format and its corresponding semantics written in natural language. In particular, the DIA standard permits the construction of device and coding format independent adaptation engines. It is important to note that only the tools used to guide the adaptation process are specified by DIA, the realization of the actual adaptation engines is left open to industry competition. In the following, the MPEG-21 tools relevant to this paper are briefly introduced.

2.1 Generic bitstream syntax description

The generic Bitstream Syntax Description tool (gBSD) can be used to adapt scalable multimedia content independently of its actual coding format. By using gBSDs, i.e., metadata, the structure or the layers of scalable content can be described on an abstract level. Parts of the bitstream (e.g., certain layers) can be described by gBSD units in a hierarchical manner including information about their position within the

bitstream and certain layer information. In the case of SVC, for instance, gBSD can be used to describe the frames and their corresponding Network Abstraction Layer (NAL) units within the bitstream. The actual adaptation of the scalable content by gBSD is performed in two steps. The first step is performed in the metadata domain. In fact, the gBSD that describes the bitstream is adapted by removing the parts that describe those parts of the bitstream that have to be removed, for example, the NAL units that belong to a certain temporal layer. The standard does not specify how this adaptation of the metadata should be performed. The implementations that can be found in the literature range from an XSLT approach [48] that transforms the initial gBSD using a parameterized style sheet, to stream-oriented processing paradigms (STX) [38], and to regular expression processing [31]. The second step of the generic adaptation is the modification of the actual bitstream according to the transformed gBSD. This step is performed within the normative gBSDtoBin process. The input to this process is both the bitstream and the transformed gBSD. The output is an adapted bitstream that reflects the changes made in the metadata description. Importantly, in the case of video streaming, the adaptation is not performed on the whole bitstream at once but on a per picture, NAL unit or packet basis.

The combination of using scalable video coding (like the most recent scalable extension to H.264/AVC) and metadata-driven gBSD-based adaptation allows for cheap and flexible multimedia adaptation [47]. Compared to traditional approaches in video signal processing, this form of adaptation can be performed without decoding and re-encoding, or transcoding techniques. By shifting a degree of complexity into both the encoder and decoder, the processing of the video signal that is necessary for adaptation is reduced to the simple removal of video layers. This processing is computationally cheap and can even be performed by off-the-shelf network devices [22]. The gBSD-based adaptation was used as a basis for our work since it was already intensively evaluated and deployed as part of various EC-funded projects (DANA E [7, 19], ENTHRONE [9, 41]).

In addition to performing adaptation according to different terminal capabilities (e.g., video resolutions and bit rates), the gBSD-based approach can also be used to annotate the video stream on a semantic level. Different scenes of a video can be annotated off-line with keywords or concepts taken from an ontology. Based on the different preferences of a user, the gBSD-based adaptation can then be used to compose semantically adapted videos that are tailored to the user's needs [50]. Although this kind of adaptation is feasible with the technology used in this paper, the actual focus of this work remains on adapting video content according to the network conditions.

While the gBSD tool is basically used for codec-agnostic adaptation of scalable content, the MPEG-21 DIA standard

provides three further tools that can be used for steering the adaptation process, i.e., for adaptation decision-taking.

2.2 Adaptation QoS

The Adaptation QoS tool can be used to describe the possible adaptations and parameters of the content and introduces two concepts for this purpose. The first concept is that of IOPins which are used to represent content properties, adaptation parameters or resulting quality. An IOPin can be seen as a variable that is identified by a unique name and has a discrete or continuous domain. In the context of an SVC stream, IOPins could be used for representing the temporal identifier (TID), quality identifier (QID) and dependency identifier (DID) adaptation parameters, the resulting video bit rate or the achieved quality in terms of peak signal-to-noise ratio (PSNR). The second concept that addresses the inter-relationship between IOPins is that of a module which can be interpreted as a mathematical function. Within the DIA standard three different types of modules are specified: look-up tables, utility functions and stack functions. While the first and the second are used to define functional dependencies by explicitly listing the function values for discrete function arguments, the third type allows the formulation of algebraic expressions in postfix notation. As an example, a look-up table could be used to describe the functional dependency between the temporal layer (TID) and the corresponding frame rate in frames per second (fps). IOPins can be distinguished into two disjoint sets, based on their usage within the modules. While the values of dependent IOPins are determined by a functional dependency, the values of free IOPins can be chosen arbitrarily. For the SVC example, this means that the IOPins representing the TID, QID and DID operating points are free IOPins, while the resulting bit rate is a dependent IOPin.

2.3 Usage Environment Description

The tool that provides device independence is generally referred to as usage environment description (UED). The UED provides fundamental input to any adaptation engine and includes means for describing terminal capabilities and network characteristics as well as user characteristics and the characteristics of the natural environment. For a detailed overview of this tool, the interested reader is referred to [3].

2.4 Universal Constraints Description

The universal constraints description (UCD) tool enables users and providers to further constrain the usage of a Digital Item. With this tool, it is possible to describe two types of constraints that impact the adaptation process. Limitation constraints can be used to constrain the solution space

for the adaptation decision-taking, for example, by preventing the resulting bit rate of a resource from being set higher than the network's nominal bandwidth. Limitation constraints are formulated as Boolean expressions that have to be satisfied for a valid adaptation decision. Typically, limitation constraints are specified by referencing values of both the UED (e.g., screen resolution) and the IOPins within the Adaptation QoS description. Additionally, several optimization criteria can be specified to guide the adaptation decision-taking. In the MPEG-21 DIA terminology, these optimization criteria are called optimization constraints and are also represented as expressions in postfix notation using the stack function.

2.5 Adaptation Decision-Taking

The decision-taking process can be performed by solving an optimization problem [27] that is normally derived from metadata. The goal is to find an assignment of values for the IOPins that does not violate the given limitation constraints and is optimal concerning the specified optimization constraints. However, finding an optimal assignment of IOPins is limited to the free IOPins as the dependent IOPins are calculated based on the values assigned to the free ones. The generic design of Adaptation QoS and UCD spans a mixed-variable multi-criteria optimization problem with general constraints. In the literature [23,27], different algorithms were proposed for solving such optimization problems. However, it turns out that for scalable bitstreams it is sufficient to focus on optimization problems containing only discrete IOPins, since the layered encoding only allows for adapting the content in discrete steps. This allows generic and highly efficient software implementations to derive (by parsing metadata) and solve optimization problems. Such a component is generally referred to as an adaptation decision-taking engine (ADTE). The advantage of this approach is that the actual control logic for the adaptation is defined by metadata (Adaptation QoS, UCD) while the software component that interprets the metadata—the Adaptation Decision Taking-Engine—remains generic.

3 Related work

Cross-layer (XL) designs have been promoted in wireless communications as an important paradigm for optimizing the utilization of scarce wireless bandwidth. They address situations where different OSI layers may cooperate to improve the ability of applications to achieve certain objectives such as QoS guarantees, power saving, customization according to user preferences, etc. Most of the work on cross-layer optimization has focused on MAC and PHY layer interactions in wireless environments. Few studies have considered higher-level interactions such as the translation of user/terminal/application-level QoS requirements into effective

QoS mechanisms. Nevertheless, cross-layer designs may be established by either integrating the functionality of different layers into a single protocol or by tight cooperation between adjacent (or non-adjacent) layers [5].

A classification of the different cross-layer designs that have emerged over the last decade can be found in [43]. Based on the direction of the information flow between the layers in the protocol stack, the approaches can be classified as either bottom-up or top-down. In a bottom-up approach, information from lower layers (e.g., the physical or the MAC layer) is provided to a higher layer (e.g., the application layer) to optimize the parameter selection on that layer. In contrast to that, a top-down approach passes information about the application's QoS requirements to lower layers to optimize the actual wireless transmission. A prominent example for a top-down approach is to mark packets according to their importance at the application layer and to employ different scheduling strategies at lower layers. According to the classification, a combination of both bottom-up and top-down approach is referred to as an integrated approach. In addition, the authors distinguish between application-centric and MAC-centric approaches, depending on the layer where the actual control of the optimization takes place. Most importantly for our work, in [43] the authors also formalize the cross-layer design problem as an optimization problem and discuss the problems that arise in context of solving this problem. According to this classification scheme for cross-layer designs, the approach presented in this paper can be classified as application-centric and bottom-up since the optimization takes place at the application layer and uses information from lower layers.

A more cautionary view on the emergence of cross-layer approaches is given in [20]. The authors claim that cross-layer design can be counter productive and longer-term architectural principles and may have negative impacts as well. These concerns about the applicability of such approaches are corroborated by two examples that clearly demonstrate that optimizations on different layers can also degrade the overall performance of a system. The paper points out the importance of architectures and the use of standards in order to foster the proliferation and the long-term success of the approach. This is aligned with our approach to use standard MPEG-21 descriptions and components to build a cross-layer architecture.

Among the many different wireless network technologies that are subject to cross-layer optimizations our approach clearly focuses on 802.11-based networks. Since the initial design of 802.11 did not support any Quality of Service mechanisms, a variety of proposals and algorithms for improving the performance have emerged in the literature, e.g., [6,28]. As a consequence, the MAC layer was recently extended in 802.11e to support QoS by differentiating between packet priorities. Also a lot of research work has been

carried out to utilize this new functionality in the MAC layer [14,26]. A common property of all these approaches is that they require modifying or deploy software on the wireless access point. Our approach, however, tries to improve the performance of video streaming over wireless networks by modifying only the end points of the communication but also taking the behavior of the access points (APs) into consideration. This end-to-end approach and the use of interoperable metadata should alleviate issues raised the complexity of deployment in a real-world scenario.

Multimedia traffic is very rigid with respect to transmission over wireless links and subsequently a lot of work has been focused on improving the performance of audio-visual streams. Dynamic adaptation of the video and intelligent packet scheduling by taking the rate-distortion performance into consideration can be seen as very promising techniques [1,4,21,30]. In addition, adaptive forward error correction (FEC) and packet fragmenting schemes have been proven to increase the quality of the transmitted video [5,24,30]. These ideas are therefore considered in our architecture.

4 MPEG-21-based cross-layer streaming approach

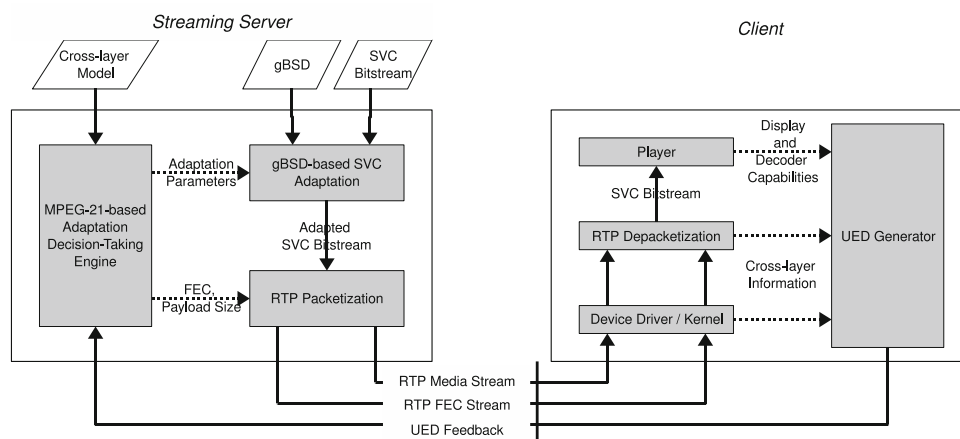
In our approach, we focus on the streaming of scalable video content to a wireless terminal. The content is delivered from a streaming server which is directly connected to a wired core network. At the content consumer’s side of the transmission chain, a mobile terminal is connected to the core network through an access network. For our further investigations, an 802.11g-based network is used as the access network. The core network is assumed to provide QoS mechanisms to ensure the transmission of real-time multimedia traffic according to pre-negotiated network parameters (e.g., delay, jitter, packet loss) between providers (i.e., Service Level Agreements). Additionally, an admission control algorithm is employed to reject video streams that would exceed the actual capacities of the core network. Therefore, in

this QoS-enabled core network the QoS parameters are statistically engineered. This network topology and the QoS assumptions correspond with the general architecture and achievements of the ENTHRONE project [9] in which this work was conducted.

Within the access network, an 802.11g connection does not provide any of the above-mentioned guarantees since packet loss is unavoidable due to the nature of wireless communication, i.e., it is always subject to a degree of interference, channel fading and signal attenuation. These effects lead to unreliable networking conditions without any QoS guarantees concerning the available bandwidth and packet loss. In order to provide a smooth video playback at the wireless terminal, the video stream has to be adapted dynamically according to the changing conditions experienced within the wireless access network. The focus of our research was to steer and perform cross-layer media adaptation based on MPEG-21-based description tools. The layers that are considered to be under the control of the adaptation process are the application layer (video adaptation) and the transport layer (packetizing). On the other hand, we obtained statistics from the application layer (player capabilities), transport layer (packet statistics) as well as from the link and physical layers (802.11-related statistics).

The architecture of both the streaming server and the wireless terminal is depicted in Fig. 1. The adaptation of the SVC video bitstream is performed at the streaming server using a generic MPEG-21-based adaptation engine [40]. The normative generic Bitstream Syntax Description tool is used to describe the frames and their corresponding NAL units in the bitstream. The description describes the offset and the length of each NAL unit and conveys the layer information. The gBSD marker attribute therefore contains the TID, DID, and QID values which signal to which temporal, spatial, and SNR scalability layer the NAL unit belongs to. Based on such a description, a codec-agnostic media adaptation can be performed in two steps, as described in Sect. 2.

Fig. 1 Architecture of the MPEG-21-based streaming server and client



The adapted SVC bitstream is then streamed over the network using the real-time transport protocol (RTP) protocol [35]. The NAL units of the scalable video bitstream are packetized into RTP packets according to the recent IETF SVC payload format draft [45]. Depending on the actual size of the NAL units, the packetizer can aggregate multiple NAL units into one RTP packet (single-time aggregation packets) or fragment one NAL unit into several packets (fragmentation units). The packetizer can also be dynamically configured to generate packets with a given maximum payload size. The RTP packetizer can optionally generate a second RTP stream that carries FEC packets according to the payload format for generic FEC [34]. The FEC stream can be used to recover lost media RTP packets by reconstructing them based on the other original media and FEC packets, thus increasing the error resilience of the video transmission in the case of bad link conditions. Typically the packetizer calculates the FEC packets using a linear block code. The amount of FEC packets that are generated can be configured dynamically by specifying the (n, k) parameters for the block code. This means that $(n - k)$ FEC packets will be used to protect a block of k media packets, essentially providing resilience against a maximum packet loss rate of $p = (n - k)/n$ when considering that also FEC packets themselves are subject to loss.

At the mobile terminal, both the SVC RTP packets and the optional FEC stream are depacketized and the resulting SVC bitstream is delivered to the player. The UED generator component is responsible for collecting information from the different layers and for creating an appropriate UED description. The information provided by the player includes the maximum display resolution that can be used for the video playback. Additionally, the player (which includes the SVC decoder) specifies a maximum bit rate of the video that it is able to decode in real-time on the given terminal. The statistics obtained from the depacketizer include the packet loss and the jitter that is encountered on the RTP level based upon the sequence numbers and packet timestamps. Low-level information from the wireless link is obtained directly from the corresponding operating system components and delivered to the UED generator. Basically, the actual signal strength and the current physical rate can be monitored on a per-frame basis. The UED generator aggregates all this information and transmits the UED description, effectively a representation of the actual usage context, back to the streaming server.

Assuming such an architecture, our focus moved to deriving a model for controlling the adaptation at the streaming server in order to cope with the dynamics of the network link. Since it considers both parameters and information from different layers of the protocol stack, it may be referred to as a cross-layer model (XLM). Following the generic approach of the MPEG-21 framework, this control logic is represented by Adaptation QoS and UCD descriptions. This model,

together with the usage context description (UED) that is provided by the client, serve as input to the MPEG-21-based Adaptation Decision-Taking Engine in the streaming server. Video adaptation and packeting is subsequently based on the adaptation decision that is obtained by solving the optimization problem as explained in Sect. 2. While it is quite straightforward to implement very rudimentary cross-layer models by considering only the packet loss, our goal was to develop a sophisticated model that takes more of the cross-layer information into consideration. Therefore, some initial experiments were performed in order to develop an idea of the real-world performance of our approach and to construct a model based on the results obtained.

5 Experimental setup and results

Based on the architecture introduced above, we implemented a prototype of both the streaming server and a corresponding client. The implementation of the server is based on Apple's Darwin Streaming Server (DSS) [8], which is available as open source software. The DSS supports the RTSP protocol for creating and removing video sessions and permits the integration of third-party modules. By utilizing this modular concept, we implemented a module for adapting and streaming SVC content. For the two-step gBSD-based adaptation, we used the libxslt library [25] to transform the gBSD and implement the normative gBSDtoBin process. The actual packeting of the NAL units of the SVC bitstream was performed by using a slightly modified implementation of the H.264/AVC RTP packetizer available in the GPAC library [15].

The RTSP client that is shipped with the DSS was used as the basis for the implementation of the client. The depacketized SVC stream was delivered to the player which, in our prototype, did not decode the content in real-time because of computational constraints. Instead, the bitstream was dumped to a file which allowed for off-line decoding and analysis of the content received at the terminal. The statistics of the wireless link were obtained by extending the Linux 802.11 protocol stack with some monitoring hooks for user-space applications. The new 802.11 stack was introduced into Kernel 2.6.22 and provides common functionality that can be used by the different device drivers for the various chipsets. According to the architecture, the UED generator component collects the statistics and capabilities from the components and submits them in the form of a UED to the server. Since MPEG-21 does not specify by which protocol the XML descriptions should be exchanged, we decided to transmit the UED using the RTSP SET_PARAMETER method. The method allows the client to send user-defined content to the server and can be seen as similar to the well-known POST method of the HTTP protocol. The SET_PARAMETER

method is used once during the session setup phase and continuously during the streaming of the video.

For the experiments, the streaming server was deployed on a Linux desktop PC while the client was installed on a portable notebook computer. The desktop was connected to a wireless access point (AP) via a 100Mbps Fast Ethernet cable, which represented the core network. Since the nominal bandwidth of the Fast Ethernet link exceeds the bit rate of the video content by far, no packet loss was expected on that link. This was also observed during the course of the experiments. The notebook was connected to the AP via its internal WiFi network interface (Intel 3945ABG Chipset, iwlmwifi Linux driver). In order to figure out the impact of the AP on the communication chain, we performed the experiments with two different devices for the AP. The first AP evaluated was a Linksys WRT54GL using the openWrt firmware (whiterussian 0.9). In the rest of this paper this device is referred to as the Linksys AP. The second AP was a D-Link DIR-635 with firmware version 1.09W, which we will subsequently refer to as the Dlink AP. Automatic rate selection and retransmissions at the link layer were enabled at both access points. On the other hand, fragmentation at the link layer was disabled for both APs.

The goal of the experiment was to stream scalable video content from the server to the client while the client moved around in an office environment. A video stream with 25 frames per second and a constant bit rate of 750kbps was selected as the test video content. For this first experiment, all the layers of the stream were transmitted to the client, which means no adaptation was performed. At the client the signal strength (in dBm) as reported by the driver, the physical rate that was used by the AP for transmission and the delay was monitored for each link layer frame received. Importantly, there was a one-to-one mapping between the link layer frame and RTP packet since fragmentation at link layer was disabled and the RTP packetization was done such that no fragmentation occurred on the IP layer. Delay was determined by comparing the actual receipt time of the frame with the time when frame was expected to be received by the client. Additionally, the loss of packets was monitored by comparing the sequence numbers in the RTP headers. Rather than focusing on an average loss rate, in the experiment considered the number of subsequent packets that were lost, as the typical size of burst losses is relevant for selecting appropriate parameters for the FEC since larger burst losses require higher block sizes for their correction.

The experiment was designed to last 50 s with a procedure as follows:

- Initially ($t = 0$), the client is located near the access point (< 1 m) and begins to receive the streamed content. During this phase the client experiences very good link quality.
- A short time after the streaming has started, the client moves away from the access point at pedestrian speed, leaves the office room where the AP is deployed and moves further away along the corridor outside the office. This movement was completed within the time interval $0 < t < 20$ s.
- The movement stops at a certain position away from the access point and the client remains at this position for a while. This occurs in the interval $20 < t < 40$ s.
- Then, at $t = 40$ s the client starts to move back towards the AP for about 10 s until the experiment ends. It should be pointed out that, at the end of the experiment, the client has not returned back to its initial start position.

The purpose of the experiment was to obtain information about the behavior of the automatic rate selection algorithm at the AP when signal degradation was detected. This was based on the assumption that a reduction in the physical transmission rate by the AP produces a decrease in throughput over the wireless link which might require the adaptation of the video content at the streaming server. Furthermore, we wanted to figure out the impact of the signal strength on the packet loss to determine appropriate heuristics for dynamically adding FEC packets to the RTP stream. In order to obtain significant results, we performed the experiment several times both with the Dlink and the Linksys APs and collected the monitoring information from each run. It turned out that, although the results were never exactly identical due to the unpredictable nature of the wireless link (fading, etc.), the trend observed in each experiment was the same.

A comparison between the Linksys and Dlink APs can be made based on Figs. 2 and 3. Both plots contain the signal strength of each frame received at the client and the corresponding physical rate. It can be seen that, as a consequence of the movement away from the AP, the signal strength gradually decreases during the first 20 s. This causes both access points to reduce the physical rate in favor of a more robust modulation and coding scheme. However, since the rate selection algorithm is not normative, both APs adapt the physical rate in a different fashion. While the Linksys AP tries to increase the physical rate very aggressively, the behavior of the Dlink AP can be characterized as more defensive. The conclusion that can be drawn is that although the signal strength is quite similar in both experiments, the APs achieve different throughput based on their rate selection algorithms/strategies.

In addition, the delay and the packet loss during the experiment were monitored and evaluated for both access points. The corresponding plots can be found in Figs. 4 and 5. The line shows the delay of the incoming packets, whereas the dots represent losses at that moment in time, in particular they indicate how many subsequent packets were lost. It can be seen that after $t > 20$ s, where the AP is operating at the

Fig. 2 Signal strength and physical rate—Linksys AP

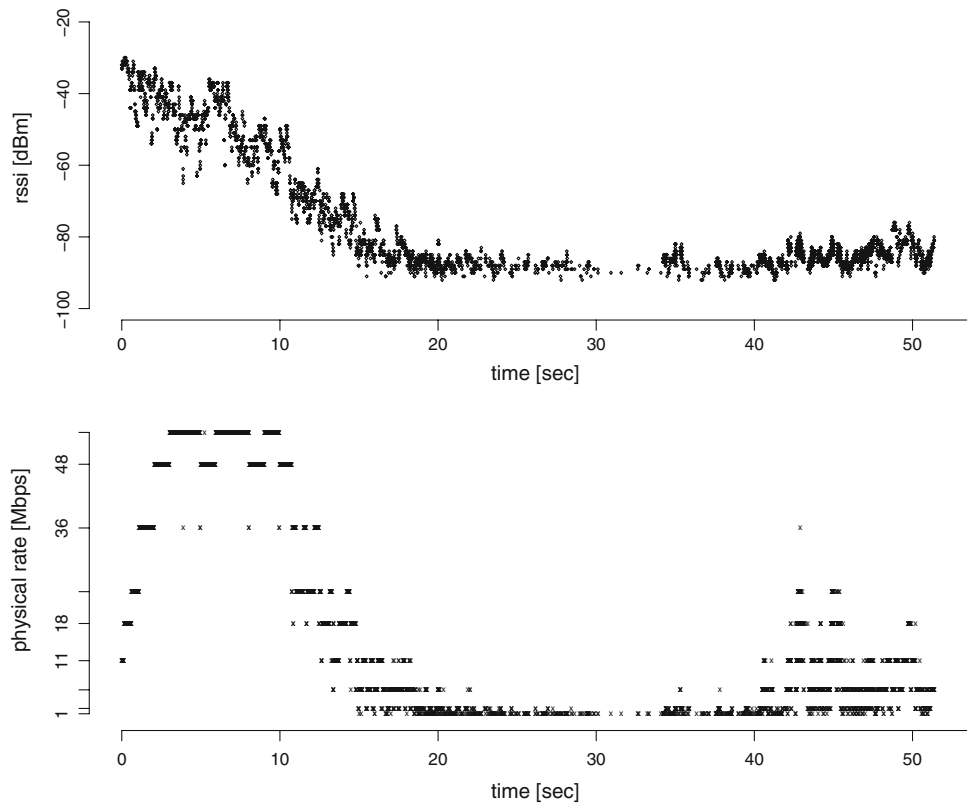
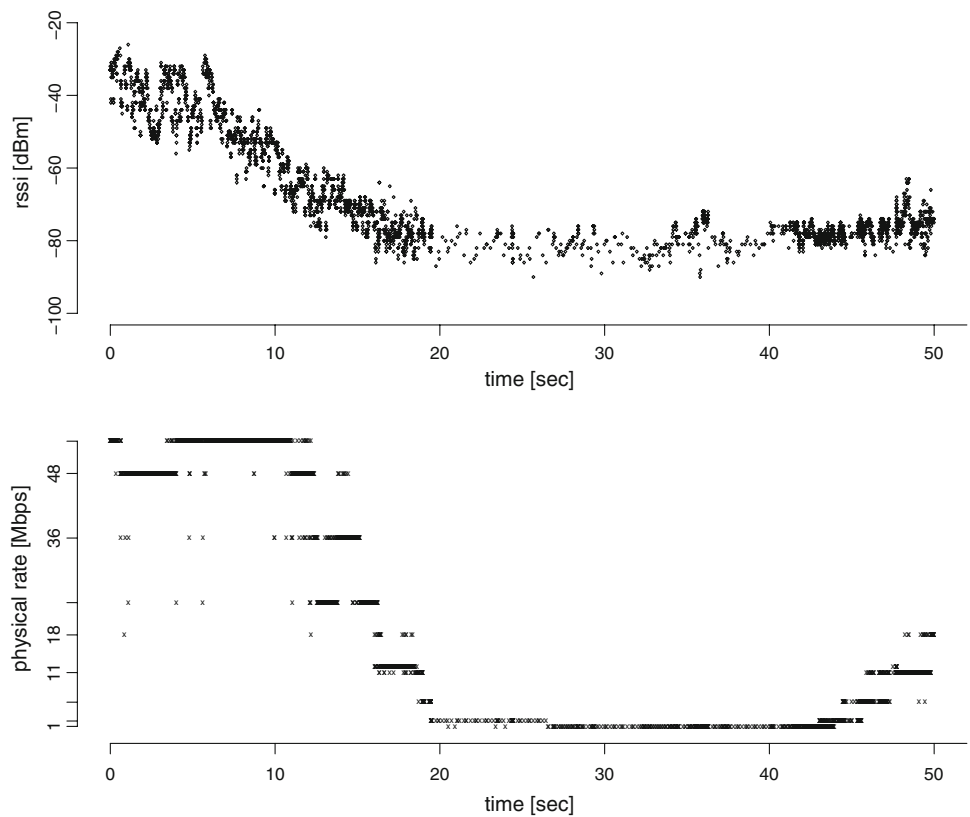


Fig. 3 Signal strength and physical rate—Dlink AP



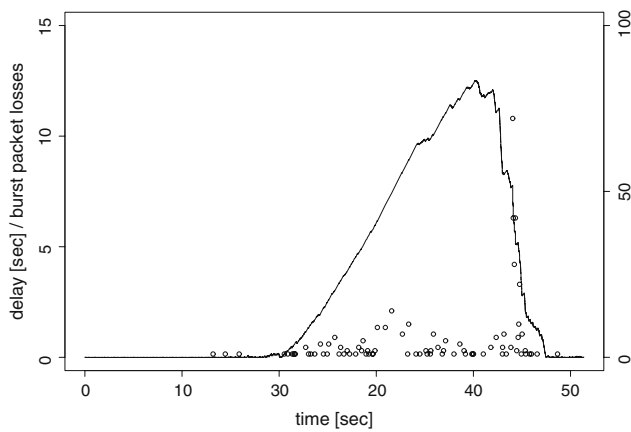


Fig. 4 Delay and burst losses—Linksys AP

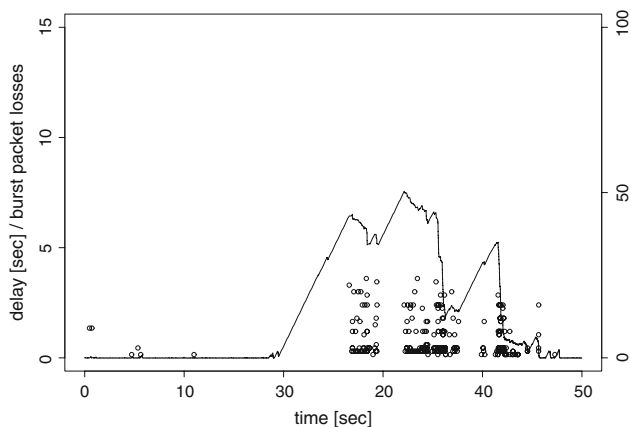


Fig. 5 Delay and burst losses—Dlink AP

lowest physical rate, the delay experienced by the packets begins to increase. This can be explained by the fact that the AP retransmits frames at the link layer which further decreases the throughput of the link. Since the video is streamed at 750 kbps, the packets are enqueued at the AP. The increasing queue size causes a delay which can be as long as several seconds. The different plots indicate that the queue size and the delay introduced differ between APs. While some of the packets at the Linksys AP suffer from a delay of more than 10 s, the maximum delay observed at the Dlink AP was less than 8 s. Besides this, both APs also demonstrate different behavior concerning packet loss. While the Linksys AP tended to produce smaller burst losses, the Dlink AP did not experience a single packet loss during the interval 20–27 s and instead encountered a higher number of burst losses.

The observations can be summarized as follows:

- The physical rate chosen by the AP drops very fast as signal strength decreases. In both cases there is a sharp transition between good and bad link conditions: the rate falls from 54 to 1 Mbps within 5 s. This can be seen as an indication that feedback from the client concerning

QoS or media adaptation should be sent very frequently in order to cope with the rapidly changing link conditions.

- The rate selection algorithm and the buffer size at the AP significantly influence the behavior of the wireless streaming. The aggressiveness of rate selection at the AP has an impact on the throughput achieved via the wireless link. However, this rate selection algorithm is non-normative and can be implemented differently by each AP vendor. This behavior is detrimental for our client/server-based approach since there is no means for the client to figure out the algorithm employed by the AP and signaling this information to the streaming server.
- The most interesting finding is that most of the packets did not get lost on the wireless link itself, but were dropped at the AP because its queue was full. In case of bad wireless conditions, the throughput of the link decreases due to low physical rates and retransmissions and the queue at the AP grows until no more incoming packets can be enqueued. As a consequence, the control logic of media adaptation should be aware of congestion and reduce the sending rate appropriately. Therefore, it must not only consider packet loss but also delay as an indicator for congestion. As one can see from Fig. 5, it is also necessary to react to increasing delay (rather than just packet loss), since once packet loss is detected it is already too late for adaptation.

As a consequence of these conclusions, it was considered necessary to investigate the applicability of rate and congestion control in the context of MPEG-21-based adaptation.

6 Rate and congestion control

Over recent decades, a lot of research has been conducted in the context of rate and congestion control in particular for the streaming of multimedia content. The success of the widely deployed TCP protocol in the Internet was a major motivation for research in this area. While TCP is in general considered as unsuitable for the real-time transmission of multimedia content, it has some properties that are in fact desirable for multimedia traffic. First, TCP is an end-to-end protocol that does not require any knowledge of the network topology and the available capacity of the network links. Although it is not aware of the topology, its congestion control algorithm is able to utilize the available bandwidth in the network in a very adaptive and efficient way. Additionally, the congestion control algorithm provides fairness in the case that multiple TCP streams are sharing a single network link and prevents the collapse of the network due to congestion.

In the context of congestion control for streaming media, the desirable characteristics can be summarized as follows:

- *TCP friendliness*. A data flow is considered TCP-friendly if its long-term throughput is similar to the throughput of a conforming TCP connection under the same conditions [10].
- *Responsiveness*. The amount of time an algorithm needs to decrease its sending rate in response to severe network congestion. If this time interval is long, an algorithm is considered as slow to respond. If the interval is short, an algorithm is considered as being highly responsive to congestion [2,49].
- *Smoothness*. Smoothness denotes the variation experienced by the sending rate for a particular flow. If the sending rate changes very rapidly over short time intervals, an algorithm is not considered as behaving smoothly. Smooth algorithms, i.e., with slow changes of the sending rates, are in general better suited to streaming applications and are also more resilient to network noise [49].

A variety of different protocols and algorithms for congestion control can be found in the literature. In order to find an appropriate algorithm that is both suitable for our adaptation scenario and its MPEG-21 realization, an extensive amount of literature was examined.

6.1 Rate Adaptation Protocol

The Rate Adaptation Protocol (RAP) [33] is a TCP-friendly congestion control mechanism for end-to-end unicast communication. It is intended to be used by real-time applications which transfer Voice over IP data, audio or video streams over a network. Such streaming applications could include Internet telephony tools or other entertainment and collaboration services such as video communication via Web cameras. RAP's primary goal is to use a fair share of bandwidth for its streams and to behave in a TCP-friendly manner. Another goal is to separate the congestion control performed by RAP from the error control performed by the application. A comparison of TCP congestion control mechanisms [46] shows that RAP behaves fairly when competing with other TCP connections as long as it experiences no or few timeouts. This is because RAP reacts to the arrival of three duplicate ACKs in almost the same fashion as TCP does. However, when the network is under heavy load and TCP experiences many timeouts, RAP acts more aggressively by taking up more bandwidth.

Another performance comparison between two rate-based congestion control mechanisms and the congestion control of TCP is presented in [18]. In that study, the ns-2 network simulator [42] served as the simulation environment. RAP and the TCP-Friendly Rate Control protocol (TFRC), which is explained in detail in Sect. 6.3, were evaluated against different implementations of TCP's congestion control. The

goal was to find out which rate-based approach behaves in a more TCP-friendly fashion in periods of network congestion. Furthermore, two different router queuing disciplines, namely Drop tail and Random Early Detection (RED) [13] were investigated regarding their impact on congestion control. The authors concluded that both RAP and TFRC were able to achieve similar throughput as a TCP connection would achieve when traversing the same network path under the same conditions. Interestingly, their experiments revealed that TFRC behaves in a more TCP-friendly fashion and more robustly than RAP. A second outcome of the simulation was that both mechanisms performed better when the routers implemented the RED queuing discipline [18].

6.2 Enhanced loss-delay-based adaptation algorithm

The loss-delay based adaptation algorithm (LDA) was first proposed in 1998 [37] and extended in 2000 [39]. The extension is indicated by the name change from LDA to LDA+. The enhanced loss-delay based adaptation algorithm (LDA+) is a rate-based congestion control algorithm for unicast applications suitable for multimedia streaming. One of its main goals is to be TCP-friendly which means that the algorithm is expected to request a fair share of bandwidth in an environment with many competing TCP connections. However, LDA+ employs an additive increase multiple decrease (AIMD) scheme, emulating TCP's congestion control. Importantly, rate-based adaptation algorithms often come with a special feedback mechanism which provides loss and delay information to the sender. This feedback algorithm is usually not normative but part of the proposed congestion mechanism. In this context, LDA+ has a notable advantage over comparable algorithms, because it utilizes RTCP messages for feedback transmission. A performance evaluation shows that LDA+ behaves fairly when competing with TCP flows [39]. Unfortunately, the AIMD scheme causes throughput variations that are rather undesirable for multimedia streams. Moreover, a weakness of the RTCP feedback mechanism is the infrequent report interval which leads to LDA+ responding slower to changing network conditions [46]. Another aspect concerns the limited precision of loss rate reports within standard RTCP. Only 8 bits are available in a receiver report for representing the loss rate (a value between 0.0 and 1.0). With $2^8 = 256$ possible steps, the lowest possible loss rate that can be signaled is about 0.0039. Lower loss rates are rounded to zero which makes an LDA+ flow request more than the appropriate bandwidth share [39].

6.3 TCP-friendly Rate Control Protocol

Another well-known protocol is the TCP-friendly Rate Control Protocol (TFRC) which is suitable for unicast applications. It also belongs to the category of rate-based

approaches and was developed for the transmission of audio/video content in best-effort networks. TFRC was first proposed in 2000 [10] and later published as an IETF protocol specification in 2003 [16]. Minor adjustments and continual improvements are being made, preparing the protocol for deployment in large-scale best-effort networks such as the internet. An updated draft with many helpful clarifications can be found in [17]. Other congestion control algorithms like RAP are based on the AIMD principle and therefore show a typical sawtooth-like throughput graph. As already mentioned earlier, this is undesirable for streaming applications. A possible solution is to introduce buffering at the receiver to compensate the variations [32], but the big advantage of TFRC is that it prevents high throughput variations from rapidly changing [11]. A performance evaluation of RAP and TFRC was conducted in [18], with the conclusion that TFRC is better suited to multimedia streaming applications.

TFRC does not follow an AIMD-based congestion mechanism, but instead uses a model of TCP's throughput to adapt its sending rate accordingly. The model of choice can be expressed as a simplified version of the TCP Reno Throughput equation [29]:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{\text{RTO}} \left(3\sqrt{\frac{3p}{8}} \right) p(1 + 32p^2)} \quad (1)$$

This equation is also called TCP response function because it models the response of a TCP connection to certain network conditions in terms of throughput. The throughput T in bytes/s is modeled as a function of the segment size s in bytes, a round-trip time estimate R in seconds, the loss event rate p as a fraction between 0.0 and 1.0 and a TCP retransmission timeout value t_{RTO} in seconds. An application using TFRC adapts its sending rate according to the newly calculated rate T_{new} . If the current rate is higher than T_{new} , the rate is reduced, if it is lower than T_{new} , the rate is increased [11]. The segment size remains constant because applications using TFRC are expected to use a fixed segment size and only vary the frequency of packet transmissions upon congestion. A further simplification was made by setting the retransmission timeout value to $t_{\text{RTO}} = 4R$. Since t_{RTO} is not used for scheduling retransmissions, slight inaccuracies do not have a negative impact on the behavior of TFRC (the original calculation of the retransmission timeout in TCP would require the round-trip time and its variance value). The remaining variables R and p need to be periodically reported by the receiver so that a TFRC sender can call the response function and eventually adapt the sending rate to the current network conditions.

According to the evaluations that can be found in the literature, TFRC can be seen as the most promising algorithm for rate and congestion control. Furthermore, it is to the best

of our knowledge the only algorithm that was standardized within the IETF and can be therefore considered as likely to coexist with the other protocols deployed on the Internet. Therefore, the TFRC algorithm was selected to be implemented and evaluated over the course of the development of our MPEG-21-based cross-layer adaptation system.

7 TFRC-based cross-layer model

7.1 Adaptation QoS description

The implementation of the TFRC approach in the cross-layer model was accomplished as follows: The parameters that are used for steering the adaptation of the video and its packeting are

- the temporal id (TID) for the temporal layer,
- the dependency id (DID) for the spatial layer,
- the quality id (QID) for specifying the quality layer, and
- the packet size for the RTP packetizer.

All of these parameters are represented by an IOPin in the Adaptation QoS description. The selection of these parameters also has an influence on the actual properties of the video such as

- the vertical and horizontal resolution,
- the bit rate, and
- the frame rate.

These properties also correspond to IOPins. The functional dependencies between the parameters (TID, DID, QID) and the properties are modeled by look-up tables.

The fundamental part of the model, however, is the implementation of the TFRC throughput equation (1). It takes the RTT estimation, the packet size and the packet loss event rate as inputs and determines an appropriate transmission rate. As the packet size is also a parameter that can be selected, we used a pre-configured size of 1,450 bytes. The other two inputs to the equation are part of the usage context that is delivered by the client, as explained in more detail below. The throughput equation is realized by a stack function which uses references to the usage context as arguments and produces output in the form of an IOPin representing the transmission rate.

7.2 Usage Environment Description feedback

The feedback that is produced by the client's UED generator is delivered to the server using the RTSP SET_PARAMETER method. The most relevant parts of the UED with respect to the TFRC model are the round-trip time estimate and the loss

event rate. For the round-trip time estimate, the packetTwoWay delay attribute was used that represents the RTT in milliseconds. Although the semantics are slightly different, we decided to signal the loss event rate by using the packetLossRate attribute. An example of such a network-related fragment of the UED can be found in Listing 1. In addition to the transport layer related parts, the UED contains further information about the video decoder's maximum bit rate and the resolution of the client's display. The feedback was sent to the server once per round-trip time, as required by the TFRC specification. The RTT estimation was smoothed by using an exponential moving average with a weight of $1/8$ for the actual sample.

Listing 1 Example UED for the TFRC approach

```
<NetworkCharacteristic
  xsi:type="NetworkConditionType">
  <Delay packetTwoWay="155"/>
  <Error packetLossRate="0.000242"/>
</NetworkCharacteristic>
```

7.3 Universal Constraints Description

The constraints that are contained in the UCDs were formulated as follows for our cross-layer model. Firstly, a limit constraint prevents the ADTE from choosing TID, DID, and QID parameters that result in a video bit rate that exceeds the calculated TFRC transmit rate. Furthermore, the maximum video bit rate of the decoder forms another constraint on the video bit rate of the adapted video stream. Additionally, limit constraints restrict the video resolution to be less or equal than the display size of the client. Ultimately, however, an optimization constraint is used to select the final adaptation parameters from the feasible TID, DID, and QID combinations. The maximization constraint facilitates the best possible bandwidth utilization by maximizing the bit rate. This means that adaptation parameters are selected such that the

bit rate is as close as possible to but still below the transmit rate as calculated by the TFRC equation.

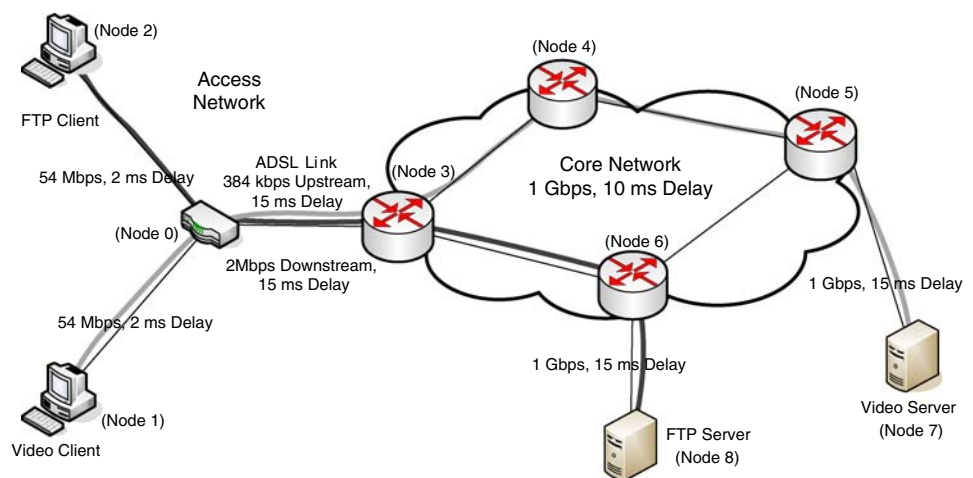
8 Simulation setup and results

The MPEG-21-based implementation of the model was evaluated using the ns-2 simulator. Since an ideal adaptation mechanism should adapt to the available bandwidth while being fair to other TCP streams, two scenarios were evaluated. In the first scenario, the capability of our approach to adapt to the given maximum capacity of a bottleneck link was investigated. The aim of this evaluation was to demonstrate how fast the approach can adapt to available bandwidth and how stable its behavior is (e.g., oscillations, etc.). In the second scenario, the fairness of our cross-layer approaches reaction to a second TCP stream was investigated. The purpose of this evaluation was to find out if the available bandwidth is shared in a fair way or if one of the streams suffers from starvation. The traffic pattern of the second stream is that of an FTP download of large file from a server.

8.1 Network topology

Figure 6 illustrates the network topology that was used for the simulation including the network characteristics in terms of bandwidth capacity and delay. The topology was defined with the network architecture of the ENTHRONE project in mind. It therefore consists of a core network where QoS can be guaranteed and an access network with no QoS mechanisms applied. For the sake of simplicity, the error-free transmission in the core network is guaranteed by over-provisioning, while in the scope of the ENTHRONE project this is done by the QoS mechanisms in the core. The access network that is illustrated in the left half of the figure is made up of a router and two client nodes. The core network shown in the right half includes the core routers and two server nodes. Access

Fig. 6 Simulation topology



and core networks are interconnected via an asynchronous digital subscriber line (ADSL).

In more detail, the network consists of:

- Two server nodes, each connected to a core network router with a bandwidth capacity of 1 Gbps and 15 ms delay. Node 7 represents a server running a video streaming application while Node 8 is used to generate the second TCP stream (background traffic).
- A core network comprised of four nodes with interconnections with 1 Gbps bandwidth capacity and 10 ms delay. The corresponding nodes Node 4, Node 5 and Node 6 represent routers that are forwarding packets. Node 3 is also part of the core network but has a special task as described below.
- An access network comprised of a router which connects two client nodes with 54 Mbps bandwidth capacity and a low delay of 2 ms to properly represent the capabilities of a local area network.
- Node 3 is the ingress/egress router to/from the core network, Node 1 represents a client terminal that runs a video player and consumes the multimedia content received from the streaming application.
- Node 2 is used as sink for the competing TCP traffic, which is an FTP client in the second scenario.
- A bottleneck link connecting one of the core nodes (Node 3) to the router in the access network (Node 0). The simulated bandwidth and delay values were chosen appropriately to comply with typical characteristics of an asynchronous digital subscriber line (ADSL), i.e., 384 kbps upstream, 2 Mbps downstream and a delay for the up/downlink of 15 ms as determined via a ping tool.

For both scenarios, a scalable video stream was chosen that can be adapted to bit rate between 90 and 3,000 kbps. Each simulation run lasted 120 s which turned out to be a sufficiently long period to get representative results. The first-in first-out (FIFO) queue of the router interconnecting core and access networks (Node 3) was configured with a drop tail policy and a maximum queue size of 50 packets. As described above, the bottleneck link between the core and access networks limited the available bandwidth for the video stream to a maximum of 2 Mbps. Figure 6 depicts the data paths of flows in the simulated network. The light grey path represents the multimedia content streamed from Node 7 to Node 1, while the dark grey path represents the competing TCP flow in the second scenario.

8.2 Scenario 1: bandwidth adaptation

This scenario was developed to find out what impact the bottleneck link has on the performance of the rate control approaches under investigation. Since the highest possible bit

rate of the video stream is 3 Mbps and the available bandwidth between the video server and the client is limited to 2 Mbps, adaptation has to be performed to avoid or minimize packet loss but to still efficiently utilize the network link.

Our proposed MPEG-21-based TFRC approach utilizes an Adaptation QoS description based on the TCP throughput model to calculate the transmit rate. As one can see in Fig. 7, our approach clearly demonstrates its ability to successfully adapt the transmission rate to the bandwidth limit. At the beginning of the simulation, the available bandwidth is probed very aggressively. The streaming rate increases rapidly until it exceeds the 2 Mbps limit by far and causes an initial 20% packet loss. However, after this initial peak, the packet loss rate remains low. A trade-off is that the system requires about 63 s until the bandwidth is fully utilized. With the feedback interval set to the current RTT estimate, the streaming rate is adapted more often so that many small-scale oscillations occur, as illustrated in the corresponding throughput graph. In this scenario, the oscillations vary between 200 and 300 kbps and therefore do not have a major impact on the video streaming process.

8.3 Scenario 2: competing FTP download

In this scenario, the video stream competes with a single FTP connection for the available bandwidth. This is the case in an environment in which one user is downloading a file from the Internet while another user in the same access network concurrently consumes a video stream. The FTP protocol was chosen to serve as background traffic, because it employs TCP as the underlying transport protocol. The behavior of a single, long-lived TCP connection is mainly characterized by the AIMD mechanism rather than by the initial slow-start phase. Therefore, this scenario is an ideal demonstration of how each rate control approach responds to the characteristics of TCP's AIMD scheme. The simulated FTP traffic in this scenario uses the NewReno TCP implementation [12] and a segment size of 1,450 bytes for the transmitted data packets.

As can be seen in Fig. 8, the TFRC-based approach demonstrates TCP-friendliness when competing with an FTP download. After a short, but aggressive initial phase, the video is streamed with approximately 650 kbps, i.e., 32% of the available bandwidth. The FTP download initiated after 10 s has a slight influence on the streaming rate, before the throughput of the two flows start to converge. However, 50 s elapses until the bandwidth is fully utilized. Presumably, this careful behavior is caused by the conservative RTT estimate ($2 \times$ one-way delay) which probably produces an RTT that is higher than it actually is. For the remainder of the simulation run, the throughput graph shows a symmetric and thus fair distribution of the available bandwidth among the two flows. Because of the careful nature of the approach, the RTP stream tends to remain below the 1 Mbps mark, whereas

Fig. 7 Simulation results of scenario 1

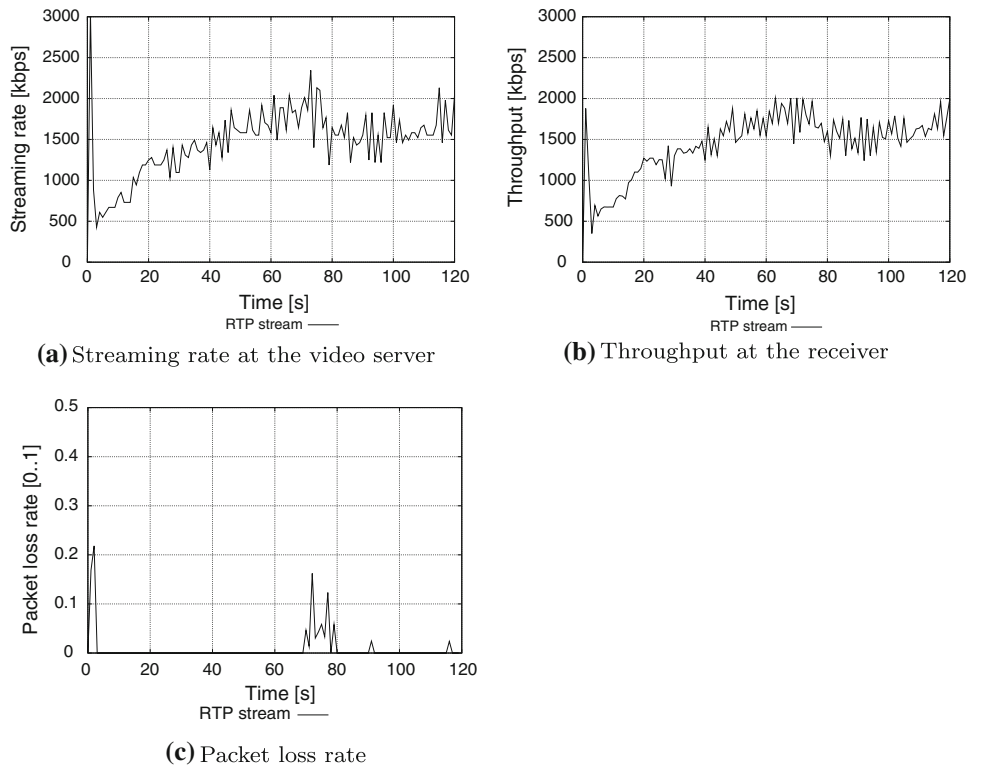
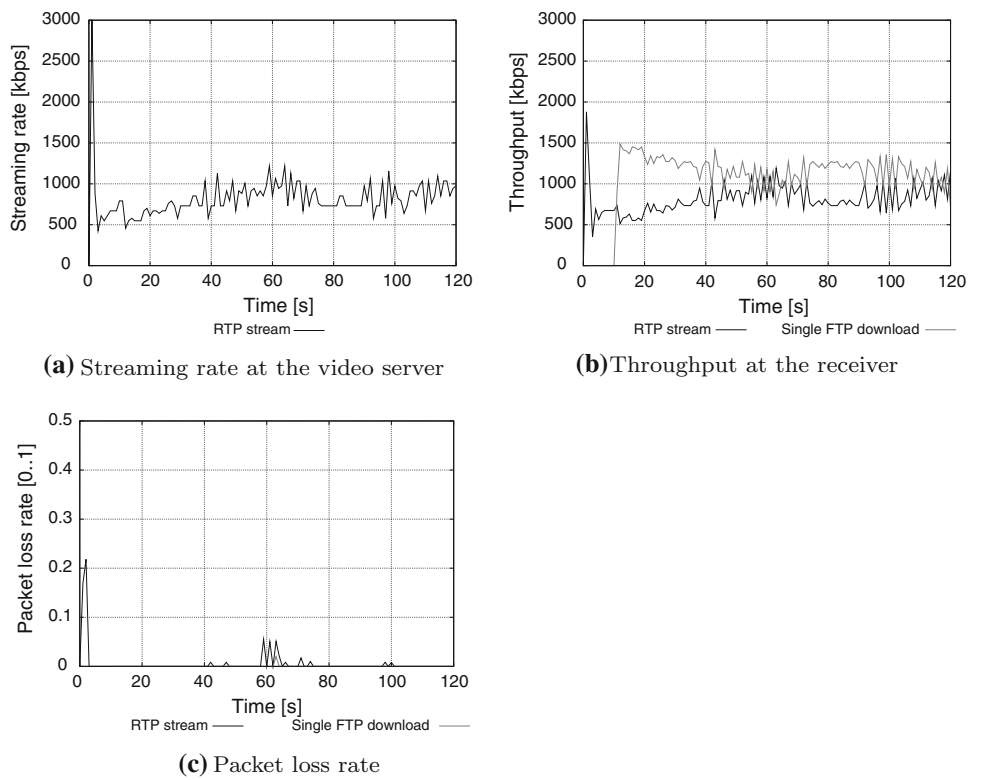


Fig. 8 Simulation results of scenario 2



the FTP flow stays above this mark. Another advantage of the TFRC-based approach in this scenario is the extremely low packet loss rate. Apart from the 22% packet loss which occurs during the first couple of seconds, the loss rates for both flows are on average far below 1%.

9 Conclusions

We presented a streaming and adaptation system for SVC video content that makes use of MPEG-21 concepts and descriptions. Adaptation is performed in a cross-layer fashion, considering both application layer characteristics (e.g., device properties) and the current network status (e.g., packet losses). Information from different layers is integrated into an MPEG-21 UED at the client and sent back to the server to dynamically adapt (layer extraction), protect (using a FEC-based scheme) and packetize the SVC content.

For validation and parameterization purposes, we measured the performance and investigated the behavior of two wireless access point (AP) models to be used in the user's access network. The major findings from these experiments were:

- The physical transmission rate chosen by an AP drops very quickly when signal strength decreases, leading to sharp transitions from good to bad link conditions and the need to send feedback (UEDs) from the client to the server frequently or spontaneously when changes occur.
- The rate selection algorithm of an AP heavily influences the behavior of wireless streaming, particularly the throughput of the wireless link. A similar observation holds for the queuing strategy and the queue size of an AP and their impact on packet delays and losses. The strategies are non-normative and differ significantly between APs. Unfortunately, it is not possible for the client to obtain information about the employed techniques and submit it to the server.
- Most importantly, it appears that most of the packets were delayed and discarded as the result of an AP's queuing strategy and not due to the wireless link itself, resulting in burst packet losses. Thus, the server should be made aware of such congestion situations and reduce the sending rate effectively, i.e., by also taking the delay as an indication of congestion into account.
- As a consequence, in such congestion situations the server should not use FEC since this would lead to an increase in the sending bit rate and would only increase congestion.

Our attention then focused on finding suitable rate control algorithms for implementation in the MPEG-21 cross-layer model. Several approaches from the literature were studied and eventually a TFRC-based rate and congestion control

algorithm was chosen and simulated. The results show that this approach can adapt to available link bandwidth effectively by using feedback provided by the client. In addition, the simulation demonstrated that the video stream behaves fairly when sharing the link with other TCP streams. This is an important issue for the large-scale deployment of adaptive streaming solutions.

Finally, the contribution of our work in the field of video streaming can be considered as twofold. First, we demonstrated that metadata-based adaptation of scalable video content can be used to adapt video streams to network conditions and thus to enhance the experience of the content consumer. As a result of layered encoding, video adaptation and processing can be performed at the server with reduced computational cost and can be done on-the-fly for each individual client. Second, the adaptation process can be controlled using MPEG-21 metadata which is provided by the client and describes the dynamically changing usage context. Since this approach only requires the deployment at the server and the client, it offers decreased deployment complexity compared with other cross-layer approaches. Additionally, the use of normative metadata provides an interoperable solution for cross-layer optimization.

Acknowledgments This work was partially supported by the European Commission in the context of the ENTHRONE project (IST-1-507637). Further information is available at <http://www.ist-enthroner.org>.

References

1. Ahmed, T., Mehaoua, A., Boutaba, R., Iraqi, Y.: Adaptive packet video streaming over IP networks: a cross-layer approach. *IEEE J Selected Areas Commun (JCS)* **23**(2), 385–401 (2005)
2. Bansal, D., Balakrishnan, H., Floyd, S., Shenker, S.: Dynamic behavior of slowly-responsive congestion control algorithms. In: *Proceedings of SIGCOMM 2001*, San Diego, pp. 263–274 (2001)
3. Burnett, I., Koenen, R., Pereira, F., Van de Walle, R. (eds.): *The MPEG-21 Book*. Wiley, London (2006)
4. Choi L., Kellerer W., Steinbach E.: On cross-layer design for streaming video delivery in multiuser wireless environments. *EURASIP J Wireless Commun Network* 1–10 (2006)
5. Djama, I., Ahmed, T., Nafaa, A., Boutaba, R.: Meet in the middle cross-layer adaptation for audiovisual content delivery. *IEEE Trans Multimed* **10**, 105–120 (2008)
6. Dunn, J., Neufeld, M., Sheth, A., Grunwald, D., Bennett, J.: A practical cross-layer mechanism for fairness in 802.11 networks. *Mobile Networks Appl* **11**(1), 37–45 (2006)
7. DANA project. <http://danae.rd.francetelecom.com>
8. Darwin Streaming Server homepage. <http://developer.apple.com>
9. ENTHRONE project. <http://www.ist-enthroner.org>
10. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicast applications. In: *Proceedings of SIGCOMM 2000*, Stockholm, pp. 43–56 (2000)
11. Floyd, S., Handley, M., Padhye, J., Widmer, J.: TCP-friendly congestion control for unicast applications. Presentation. <http://www.aciri.org/mjh/slides/tfrc-sigcomm.ps> (2000)

12. Floyd, S., Henderson, T.: The NewReno modification to TCP's fast recovery algorithm. RFC 2582. <http://www.ietf.org/rfc/rfc2582.txt> (1999)
13. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. *IEEE/ACM Trans Network* **1**(4), 397–413 (1993)
14. Foh, C., Zhang, Y., Ni, Z., Cai, J., Ngan, K.: Optimized cross-layer design for scalable video transmission over the IEEE 802.11e networks. *IEEE Trans Circuits Syst Video Technol* **17**(12), 1665–1678 (2007)
15. Gpac open source project. <http://gpac.sourceforge.net>
16. Handley, M., Floyd, S., Padhye, J., Widmer, J.: RFC 3448: TCP friendly rate control (TFRC): protocol specification. <http://www.ietf.org/rfc/rfc3448.txt> (2003)
17. Handley, M., Floyd, S., Padhye, J., Widmer, J.: Internet draft rfc3448bis-06b: TCP friendly rate control (TFRC): protocol specification. <http://www.icir.org/floyd/papers/draft-ietf-dccp-rfc3448bis-06b.txt> (2008)
18. Hassan, S., Kara, M.: Simulation-based performance comparison of TCP-friendly congestion control protocols. In: Proceedings of the 16th Annual UK Performance Engineering Workshop (UKPEW'00), Durham, pp. 199–210 (2000)
19. Hutter, A., Amon, P., Panis, G., Delfosse, E., Ransburg, M., Hellwagner, H.: Automatic adaptation of streaming multimedia content in a dynamic and distributed environment. In: Proceedings of the International Conference on Image Processing (ICIP 2005), Genova (2005)
20. Kawadia, V., Kumar, P.: A cautionary perspective on cross layer design. *IEEE Wireless Commun Mag* **12**(1), 3–11 (2005)
21. Khan, S., Peng, Y., Steinbach, E., Sgroi, M., Kellerer, W.: Application-driven cross-layer optimization for video streaming over wireless networks. *IEEE Commun Mag* **44**, 122–130 (2006)
22. Kofler, I., Prangl, M., Kuschnig, R., Hellwagner, H.: An H.264/SVC-based adaptation proxy on a WiFi router. In: Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2008), Braunschweig, pp. 63–68 (2008)
23. Kofler, I., Timmerer, C., Hellwagner, H., Hutter, A., Sanahuja, F.: Efficient MPEG-21-based adaptation decision-taking for scalable multimedia content. In: Proceedings of the 14th SPIE Annual Electronic Imaging Conference—Multimedia Computing and Networking (MMCN 2007) San Jose (2007)
24. Krishnamachari, S., van der Schaar, M., Choi, S., Xu, X.: Video streaming over wireless LANs: a cross-layer approach. In: Proceedings of the 13th International Packetvideo Workshop 2003. Nantes (2003)
25. Libxslt XSLT processor homepage. <http://xmlsoft.org/XSLT/>
26. Malli, M., Ni, Q., Turetletti, T., Barakat, C.: Adaptive fair channel allocation for QoS enhancement in IEEE 802.11 wireless LANs. In: Proceedings of the IEEE International Conference on Communications (ICC). Paris (2004)
27. Mukherjee, D., Delfosse, E., Kim, J., Wang, Y.: Optimal Adaptation Decision-Taking for Terminal and Network Quality of Service. *IEEE Trans Multimed* **7**(3), 454–462 (2005)
28. Ni, Q., Romdhani, L., Turetletti, T.: A survey on QoS enhancements for IEEE 802.11 wireless LAN. *J Wireless Commun Mobile Comput* **4**, 547–566 (2004)
29. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP throughput: a simple model and its empirical validation. In: Proceedings of the ACM SIGCOMM 1998. ACM Press, New York, pp. 303–314 (1998)
30. Qiong, L., van der Schaar, M.: Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation. *IEEE Trans Multimed* **6**(2), 278–290 (2004)
31. Ransburg, M., Gressl, H., Hellwagner, H.: Efficient transformation of MPEG-21 metadata for codec-agnostic adaptation in real-time streaming scenarios. In: Proceedings of the 9th International Workshop on Image Analysis for Interactive Multimedia Services (WIAMIS 2008), Klagenfurt (2008)
32. Rejaie, R., Handley, M., Estrin, D.: Quality adaptation for congestion controlled video playback over the internet. In: Proceedings of SIGCOMM 1999, Cambridge, pp. 189–200 (1999)
33. Rejaie, R., Handley, M., Estrin, D.: RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the internet. In: Proceedings of IEEE INFOCOM 1999, New York, vol. 3, pp. 1337–1345 (1999)
34. Rosenberg, J., Schulzrinne, H.: An RTP payload format for generic forward error correction. RFC 273. <http://www.ietf.org/rfc/rfc2733.txt> (1999)
35. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A transport protocol for real-time applications. RFC 3550. <http://www.ietf.org/rfc/rfc3550.txt> (2003)
36. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Trans Circuits Syst Video Technol* **17**(9), 1103–1120 (2007)
37. Sisalem D., Schulzrinne H.: The loss-delay based adjustment algorithm: a TCP-friendly adaptation scheme. In: Proceedings of NOSSDAV 1998. Cambridge (1998)
38. Streaming transformations for XML (STX). <http://stx.sourceforge.net>
39. Sisalem, D., Wolisz, A.: LDA+: TCP-friendly adaptation: a measurement and comparison study. In: Proceedings of the 10th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2000), Chapel Hill (2000)
40. Timmerer, C., Hellwagner, H.: Interoperable adaptive multimedia communication. *IEEE Multimed Mag* **12**(1), 74–79 (2005)
41. Timmerer, C., Ransburg, M., Kofler, I., Hellwagner, H., Souto, P., Andrade, M., Carvalho, P., Castro, H., Sidibe, M., Mehaoua, A., Fang, L., Lindsay, A., MacKay, M., Lugmayr, A., Feiten, B.: An integrated management supervisor for end-to-end management of heterogeneous contents, networks, and terminals enabling quality of service. In: Proceedings of the 2nd European Symposium on Mobile Media Delivery (EUMOB) 2008, Oulu (2008)
42. The network simulator ns-2: documentation. <http://www.isi.edu/nsnam/ns/ns-documentation.html>
43. van der Schaar, M., Shankar, N.S.: Cross-layer wireless multimedia transmission: challenges, principles and new paradigms. *IEEE Wireless Commun Mag* **12**(4), 50–58 (2005)
44. Vetro, A.: MPEG-21 digital item adaptation: enabling universal multimedia access. *IEEE MultiMed* **11**(1), 84–87 (2004)
45. Wenger, S., Wang, Y.K., Schierl, T., Eleftheriadis, A.: RTP payload format for SVC video. Internet draft draft-ietf-avt-rtp-svc-13. <http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-svc-13.txt> (2008)
46. Widmer, J., Denda, R., Mauve, M.: A survey on TCP-friendly congestion control. *IEEE Network* **15**(3), 28–37 (2001)
47. Wien, M., Cazoulat, R., Graffunder, A., Hutter, A., Amon, P.: Real-time system for adaptive video streaming based on SVC. *IEEE Trans Circuits Syst Video Technol* **17**(9), 1227–1237 (2007)
48. XSL transformations (XSLT) 1.0: W3C Recommendation, 16 November 1999. <http://www.w3.org/TR/1999/REC-xslt-19991116> (1999)
49. Yan, J., Katrinis, K., May, M., Plattner, B.: Media- and TCP-friendly congestion control for scalable video streams. *IEEE Trans Multimed* **8**(2), 196–206 (2006)
50. Zufferey, M., Kosch, H.: Semantic adaptation of multimedia content. In: Proceedings of the 48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications, Zadar, pp. 319–322 (2006)