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

Using MPEG-21 for Cross-Layer Multimedia Content Adaptation

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

Received: date / Accepted: date

Abstract In this paper, a cross-layer model – formulated using interoperable description formats – for adaptation of scalable H.264/MPEG-4 AVC (i.e., SVC) content in a video streaming system is presented, in which the user's access network is a Wireless LAN without QoS mechanisms. SVC content adaptation on the server takes place on the application layer using an adaptation process compliant to the MPEG-21 Digital Item Adaptation (DIA) standard, based on input comprising MPEG-21 DIA descriptions of content and usage environment properties. The latter descriptions integrate information from different layers on the client, e.g., device characteristics and packet loss rate, in an attempt to increase the interoperability of this cross-layer model and, thus, make it also applicable for other models. For the sake of deriving model parameters, performance measurements on two wireless access point models were taken. It turned out 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. Thus, simulations of rate adaptation algorithms are 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 by using MPEG-21 tools.

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

---

Ingo Kofler, Joachim Seidl, Christian Timmerer, Hermann Hellwagner
Institute of Information Technology (ITEC)
Klagenfurt University, Austria
Tel.: +43 463 2700 3600
Fax: +43 463 2700 3699
E-mail: {firstname.lastname}@itec.uni-klu.ac.at

Ismail Djama, Toufik Ahmed
CNRS-LaBRI Lab
University of Bordeaux-1
Talence, France
Tel.: +33 54000 3547
Fax.: +33 54000 3501
E-mail: {djama,tad}@labri.fr

1 Introduction

Cross-layer designs for multimedia networking 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., in 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, packet loss).

The Internet Protocol suite with its well-defined layers and interfaces is one rigid obstacle that hampers the deployment of multimedia services in wireless networks. Furthermore, the wide deployment of wireless networks created the demand for enhancing the performance of multimedia applications over these wireless links. Cross-layer designs are currently an active research topic aiming at 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 emerged in the last years [37][23]. In the majority of these approaches, 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 recent efforts are undertaken in the form of tackling the problem by jointly optimizing parameters at the different layers [11].

However, independently of the ways the different cross-layer designs perform, they all share the common property of compromising interoperability in favor of performance. In this paper, we present an approach which aims to increase the degree of interoperability by investigating how interoperable description formats can be applied to cross-layer designs, specifically in the context of adaptation of multimedia content. A standard to be considered for such an undertaking is 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 [10]. In particular, Part 7 of MPEG-21, i.e., Digital Item Adaptation (DIA), provides a specification of normative description formats (named *tools* in DIA) for describing both the multimedia content properties and the context of the usage of the content [44].

Our approach is based on exchanging these description formats that follow a well-defined cross-layer model and is used to steer the continuous adaptation of the multimedia content according to possibly varying usage environment conditions. The context properties comprise information from different networking layers whereas the content description provides 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 in 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 have been investigated. In anticipation of the result we have selected a TCP-Friendly Rate Control-based (TFRC) adaptation technique which is the basis for our cross-layer model. The reason for choosing TFRC is that 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 [39]. 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 comprises finding an optimal solution for the optimization problem at hand by means of a generic metadata-driven Adaptation Decision-Taking Engine (ADTE).

The paper is organized as follows. Section 2 briefly introduces the relevant tools of the MPEG-21 standard that are necessary for the understanding of the rest of the paper. The current state of the art in cross-layer designs and work related to our approach is given in Section 3. In Section 4 our approach of using MPEG-21 for performing cross-layer content adaptation is introduced. Furthermore, the architecture of our streaming environment that was used for experiments is explained in detail. Section 5 discusses the streaming experiments with two different access points and the conclusions we draw 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 sender according to network feedback. These considerations as well as some recent rate control algorithms are explained in Section 6. In order to evaluate the modifications of our approach we implemented a rate control algorithm by using MPEG-21 tools and performed simulations. Details about the realization can be found in Section 7. The simulation setup and the results we obtained from simulating two different scenarios are discussed in Section 8. The results and findings of our investigations are summarized in Section 9, which concludes this paper.

2 MPEG-21 Tools for Multimedia Content Adaptation

The aim of the MPEG-21 standard [10], 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 transactions and distribution of content within the MPEG-21 framework. A vital and comprehensive part within MPEG-21 is Part 7 of the standard, referred to as Digital Item Adaptation (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 specifies means for enabling the construction of device and coding format independent adaptation engines. It has to be noted that only tools used to guide the adaptation process are specified by DIA, the realization of the adaptation engines proper 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 for performing the adaptation of scalable multimedia contents independently of the actual coding format. By using gBSDs, i.e., metadata, the structure or the layers of a 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 using 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, e.g., NAL units belonging 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 based on a parameterized style sheet, to stream-oriented processing paradigms (STX) [6], and to regular expression processing [33]. 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. It should be noted that, 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 the adaptation can be performed without decoding and re-encoding, or transcoding techniques. By putting the complexity into both the encoder and decoder, the processing of the video signal that is necessary for adaptation degenerates to a simple removal of video layers. This processing is computationally cheap so that it can be even performed on off-the-shelf network devices [25]. The gBSD-based adaptation was used as a basis for our work since it was already intensively evaluated and deployed in different EC-funded projects (DANAE [1], [22], ENTHRONE [3], [43]).

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 semantical 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 for composing semantically adapted videos 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 is 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, 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 are *IOPins* that are used to represent content properties, adaptation parameters, or a 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 interrelationship between IOPins is called *module* and 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 ones are used to define functional dependencies by explicitly listing the function values for discrete function arguments, the third one 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 allowing for device independence is generally referred to as *Usage Environment Description* (UED). The UED provides a 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 [10].

2.4 Universal Constraints Description

To enable users and providers to further constrain the usage of a Digital Item, the *Universal Constraints Description* (UCD) tool has been specified. 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, e.g., by preventing the resulting bit rate of a resource to be higher than the network's nominal bandwidth. The 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 [29] that can be derived from the 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 [29] [26], different algorithms are proposed for solving the optimization problem. However, it turned 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 for implementing efficient software components for deriving the optimization problem by parsing the metadata and solving the optimization problem. Such a component is generally referred to as *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 particularly put forward in wireless communications as an important paradigm to optimize the scarce wireless bandwidth utilization. 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 works 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 functionality of different layers in a single protocol or simply establishing tight cooperation between adjacent (or non-adjacent) layers [12].

A classification of the different cross-layer designs that emerged in the last decade can be found in [37]. Based on the direction of 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 the information of lower layers (e.g., the physical or the MAC layer) provides information 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. Besides, the authors distinguish between application-centric and MAC-centric approaches, which mainly indicates at which layer the actual control of the optimization takes place. Most importantly for our work, in [37] 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 of 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 of the lower layers involved.

A more cautionary view on the emerging cross-layer approaches is given in [23]. The authors claim that cross-layer design can run counter to sound and longer-term architectural principles and may have negative impacts as well. The concerns about the applicability are corroborated by two examples that clearly demonstrate that optimizations on different layers can also degrade the total 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 does not support any Quality of Service mechanisms, a variety of proposals and algorithms for improving the performance emerged in the literature, e.g., [30], [13]. 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 was done to utilize these new functionality in the MAC layer [28], [18]. A common property of all these approaches is that they require a modification of, or the deployment of a certain part of software on, the wireless access point. Our approach, however, tries to improve the performance of video streaming over the wireless network 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 leverage the complexity of deployment in a real-world scenario.

Multimedia traffic is very rigid w.r.t. transmission over wireless links and also a lot of work has been done to improve the performance of audio-visual streams. Dynamic adaptation of the video and intelligent packet scheduling by taking into consideration the rate-distortion performance can be seen as very promising techniques [11], [32], [8], [24]. Besides, also adaptive Forward Error Correction (FEC) and packet fragmenting schemes have been proven to increase the quality of the transmitted video [32], [27], [12]. The ideas behind the proposed mechanisms 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 are aligned with the general architecture and achievements of the ENTHRONE project [3] in which this work was done.

In 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., properties such as interference, channel fading, and signal at-

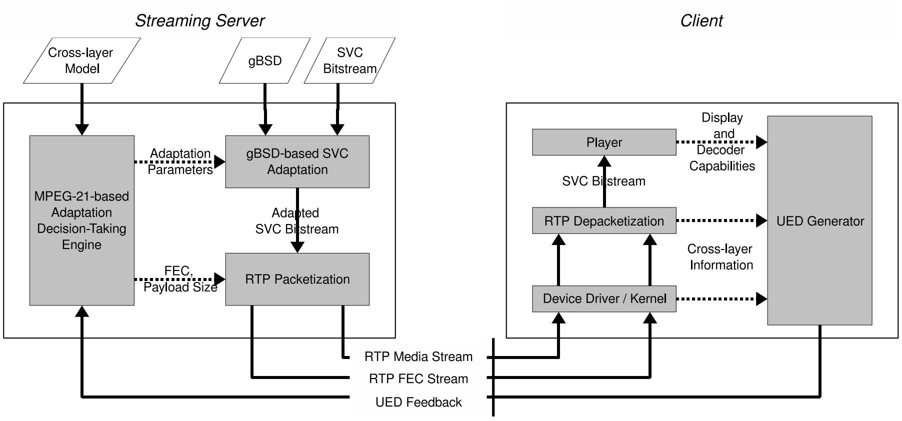


Fig. 1 Architecture of the MPEG-21-based streaming server and client

tenuation. 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 of 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 were considered to be under the control of the adaptation process are the application layer (video adaptation) and the transport layer (packeting). On the other hand, we obtained statistics from the application layer (player capabilities), transport layer (packet statistics) as well as link and physical layer (802.11-related statistics).

The architecture of both the streaming server and the wireless terminal is depicted in Figure 1. The adaptation of the SVC video bitstream is performed at the streaming server using a generic MPEG-21-based adaptation engine [42]. 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 which is accomplished in two steps, as described in Section 2.

The adapted SVC bitstream is then streamed over the network using the real-time transport protocol (RTP) protocol [38]. 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 be dynamically configured to generate packets with a given maximum payload size. The RTP packetizer can optionally generate a second RTP stream that carries Forward Error Correction (FEC) packets according to the payload format for generic FEC [36]. The FEC stream can be used to recover from lost media RTP packets by reconstructing them based on the other original media and FEC packets. This measure should increase the error resilience of the video transmission in the case of bad link conditions. 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. This basically provides a resiliency against a maximum packet loss rate of $p = (n - k)/n$ when considering that also FEC packets are affected by loss.

At the mobile terminal, both the SVC RTP packets and the optional FEC stream are depacketized and the SVC bitstream is delivered to the player. The UED generator component is responsible for collecting information of 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 actual terminal. The statistics obtained from the depacketizer include the packet loss and the jitter that is encountered on the RTP level by the inspection of 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, that represents the actual usage context, back to the streaming server.

Based on that architecture our focus was on 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 is denoted as 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 and the usage context description (UED), that is provided by the client, is input to the MPEG-21-based Adaptation Decision-Taking Engine in the streaming server. Based on the adaptation decision that is obtained by solving the optimization problem as explained in Section 2, both the video adaptation and packeting is controlled. 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, we first performed some experiments in order to get an idea of the real-world performance of our approach and to develop a model based on the results we obtain.

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) [2], which is available as open source software. The DSS supports the RTSP protocol for creating and removing video sessions and allows for integrating third-party modules. By utilizing this module concept, we implemented a module for adapting and streaming SVC content. For the two-step gBSD-based adaptation we used the libxslt library [5] for the gBSD transformation and implementation of 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 [4].

For the implementation of the client, the RTSP client that is shipped with the DSS was used as a basis. The depacketized SVC stream was delivered to the player which in our prototype did not perform a real-time decoding of the content because of computational constraints. Instead, the bitstream was dumped into 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 in 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 perform the signaling of the UED by 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 at session setup time 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 100 Mbps 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, iwlwifi 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 denoted as Linksys AP. The second AP was a D-Link DIR-635 with firmware version 1.09W, which we will refer to as Dlink AP in the rest of this paper. At both access points, the automatic rate selection algorithm and retransmissions at the link layer were enabled. On the other hand, fragmentation at the link layer was disabled for both APs.

The goal of the experiment was to stream a scalable video content from the server to the client while the client moves around in an office environment. As video content we used a video stream with 25 frames per second and a constant bit rate of 750 kbps. For this first experiment, all the layers of the stream were transmitted to the client, which means no adaptation was performed. At the client we monitored for each link layer frame received the signal strength of each frame in dBm as reported by the driver, the physical rate that was used by the AP for transmitting the frame, and the delay of the frame. It is noted that for our considerations there was a one-to-one mapping between link layer frame and RTP packet since fragmentation at link layer was disabled and the RTP packetization was done in a way so that no fragmentation on the IP layer was performed. The 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. For our observations we did not focus on an average loss rate but we were rather interested in the number of subsequent packets that were lost. The typical size of burst losses are 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 seconds with a procedure executed as follows.

- At the beginning ($t = 0$), the client is located near the access point ($< 1m$) and initiates the streaming of the content. During that phase the client benefits from a very good link quality.
- 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 done in the time interval $0 < t < 20$ seconds.
- The movement stops at a certain distance away from the access point and the client remains at this position for a while. This is done in the interval $20 < t < 40$ seconds.
- Then, at $t = 40$ seconds the client starts moving back towards the AP for about 10 seconds 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.

Based on the design of the experiment, we hoped to obtain information about the behavior of the automatic rate selection algorithm at the AP when a degradation of the signal strength is detected. A reduction of the physical rate by the AP comes along with a decreased throughput over the wireless link which might require an 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 in order to find heuristics to dynamically add FEC to the RTP stream. In order to get significant results, we performed the experiment several times both with the Dlink and the Linksys APs and collected the monitoring information for 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 Figure 2 and Figure 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 is gradually decreasing during the first 20 seconds. This causes both access points to reduce the physical rate in favor of a more robust modulation and coding rate. 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.

For both access points, also the delay and the packet loss during the experiment were monitored and evaluated. The corresponding plots can be found in Figure 4 and Figure 5. The line shows the delay of the incoming packets; the dots are representing losses at that moment in time, in particular they signal how many subsequent packets were lost. It can be seen that after $t > 20$ seconds, where the AP is operating at the

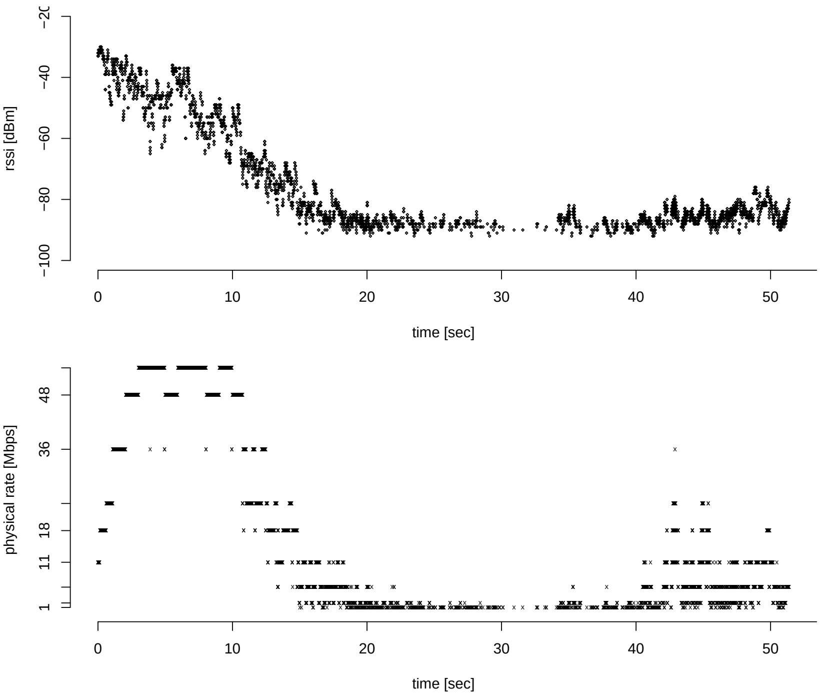


Fig. 2 Signal strength and phyiscal rate – Linksys AP

lowest physical rate, the delay of the packets is increasing. 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 of the packets which can be up to several seconds. The different plots indicate that the queue size and the delay introduced on the packets differ between APs. While some of the packets at the Linksys AP suffer from a delay of more than 10 seconds, the maximum delay observed at the Dlink AP was less than 8 seconds. Besides, both APs show also a different behavior concerning packet loss. While the Linksys AP tends to produce smaller burst losses, the Dlink AP did not cause a single packet loss during the interval 20 to 27 seconds. However, the encountered burst losses are subsequently higher at the Dlink AP.

The observations can be summarized as follows:

- The physical rate chosen by the AP drops very fast when signal strength decreases. In both cases there is a sharp transition between good and bad link conditions: the rate falls from 54 Mbps to 1 Mbps within 5 seconds. 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 fast 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

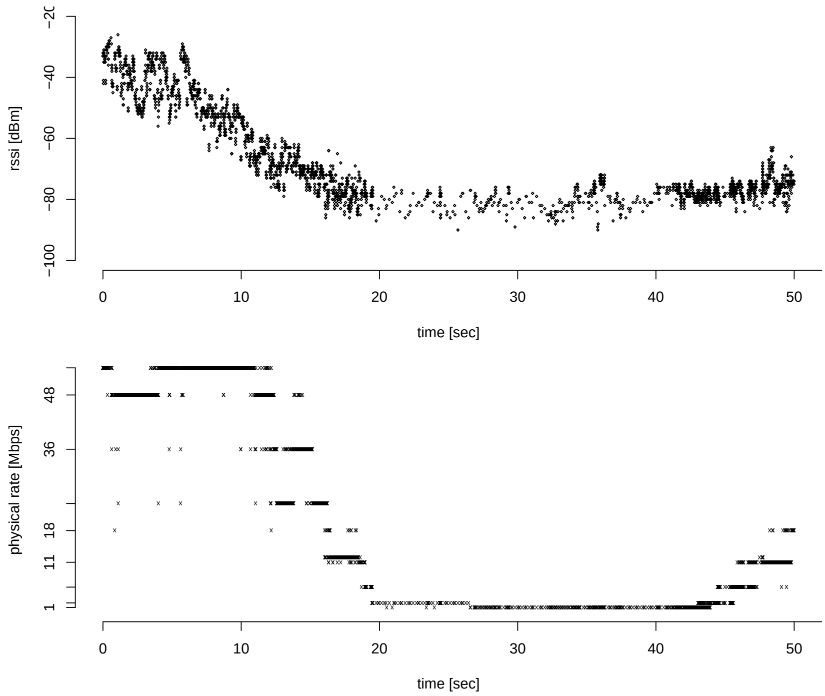


Fig. 3 Signal strength and phyiscal rate – Dlink AP

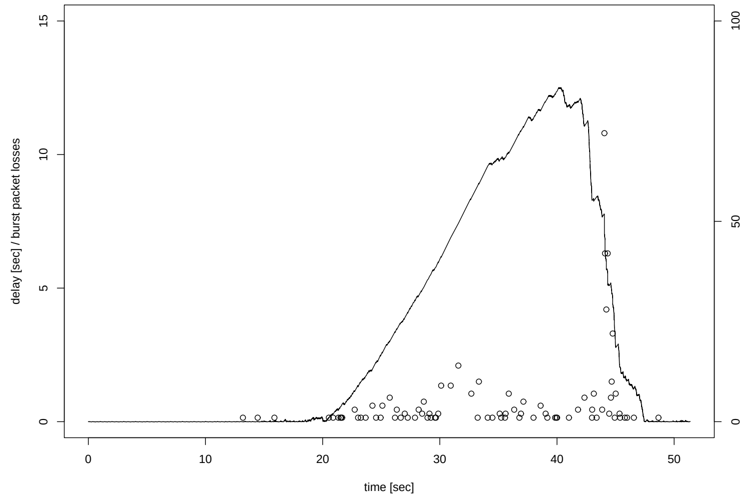


Fig. 4 Delay and burst losses – Linksys AP

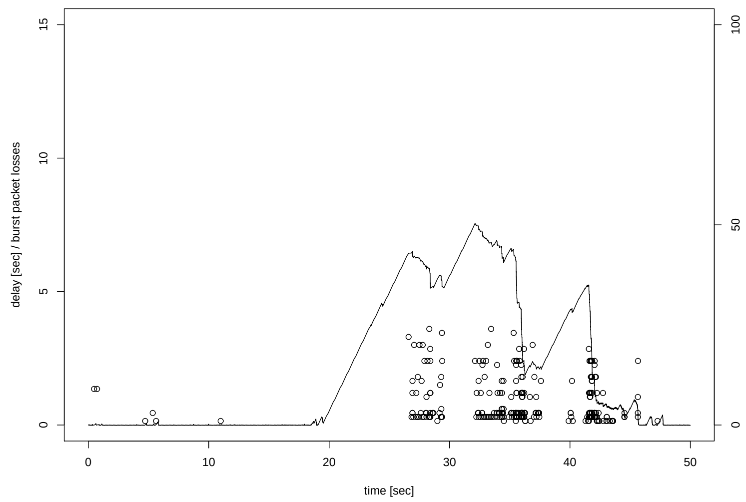


Fig. 5 Delay and burst losses – Dlink AP

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 signalling 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 queue in the AP. Since in case of bad wireless conditions the throughput of the link decreases due to low physical rates and retransmissions, 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 has to consider not only the packet loss but also the delay as an indicator for congestion. As one can see from Figure 5, it is necessary to react on an increasing delay rather than only on packet loss, since when the packet loss is detected it is already too late for adaptation.

Therefore, 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

In the last decades, a lot of research has been performed in the context of rate and congestion control in particular for the streaming of multimedia content. The success of the TCP protocol that is deployed on a large-scale in the Internet was a major motivation to perform research in this area. While TCP is in general considered as not suitable for real-time transmission of multimedia content, it conveys some properties that are desirable also 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 when multiple TCP streams are sharing a single link on the network and prevents from a congestion-caused collapse of the network.

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 [14].
- *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 slow-responsive. If the interval is short, an algorithm is considered to respond fast to congestion situations [49] [9].
- *Smoothness.* Smoothness denotes the variation of the sending rate for a particular flow. If the sending rate changes very rapidly in short time intervals, an algorithm is not considered to behave smoothly. Smooth algorithms, i.e. with slow changes of the sending rates, are in general better suited for streaming applications and are also more resilient to network noise [49].

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

6.1 Rate Adaptation Protocol

The *Rate Adaptation Protocol (RAP)* [35] 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 the network. Such streaming applications could be Internet telephony tools and 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 congestion control performed by RAP from error control performed by the application. A comparison of TCP congestion control mechanisms [46] shows that RAP behaves fairly when competing with TCP connections as long as TCP experiences no or few timeouts. This is because RAP reacts to the arrival of three duplicate ACKs almost the same way 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 [21]. The ns-2 network simulator [7] served as simulation environment. RAP and the TCP-friendly Rate Control Protocol (TFRC), which is explained in detail in Section 6.3, were evaluated against different implementations of TCP's congestion control. The goal was to find out which rate-based approach behaves more TCP-friendly in times of network congestion. Furthermore, two different router queuing disciplines, namely Drop Tail and Random Early Detection (RED) [17] 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 more TCP-friendly and more robustly than RAP. A second outcome of the simulation was that both mechanisms performed better when the routers implemented RED as queuing discipline [21].

6.2 Enhanced Loss-Delay-based Adaptation Algorithm

The *loss-delay based adaptation algorithm (LDA)* was first proposed in 1998 [40] and extended in 2000 [41]. 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 main goal is to be TCP-friendly which means that the algorithm is expected to request a fair bandwidth share in an environment with many competing TCP connections. However, LDA+ employs an additive increase multiple decrease (AIMD) scheme, emulating TCP's congestion control. 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 [41]. 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 limitation of how precise the loss rate can be reported with standard RTCP. In a receiver report only an 8 bit field is used to represent 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 [41].

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 [14] and later published as an IETF protocol specification in 2003 [19]. 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 [20]. 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 [34], but the big advantage of

TFRC is that it prevents high throughput variations from evolving [15]. A performance evaluation of RAP and TFRC was conducted in [21], concluding that TFRC is better suited for multimedia streaming applications.

TFRC does not follow an AIMD-based congestion mechanism, but 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 [31]:

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

This equation is also called *TCP Response Function* because it models how a TCP connection would respond to certain network conditions in terms of throughput. The throughput T in bytes/sec 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 [15]. 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 $t_{RTO} = 4R$. Since t_{RTO} is not used for scheduling retransmissions, slight inaccuracies do not impact the behavior of TFRC negatively (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 well co-existing with the existing protocols deployed in the Internet. Therefore, the TFRC algorithm was selected to be implemented and evaluated in the course of our MPEG-21-based cross-layer adaptation approach.

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

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

These properties have a corresponding IOPin as well. The functional dependencies between the parameter selection (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 the transmit rate to be used. As the packet size is also a parameter that can be selected, we used a pre-configured size of 1450 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 as arguments references to the usage context and stores the output in an IOPin representing the transmit rate.

7.2 Usage Environment Description Feedback

The feedback that is produced at the client's UED generator is delivered to the server by the RTSP SET_PARAMETER method. The most relevant parts of the UED w.r.t. 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 is 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,

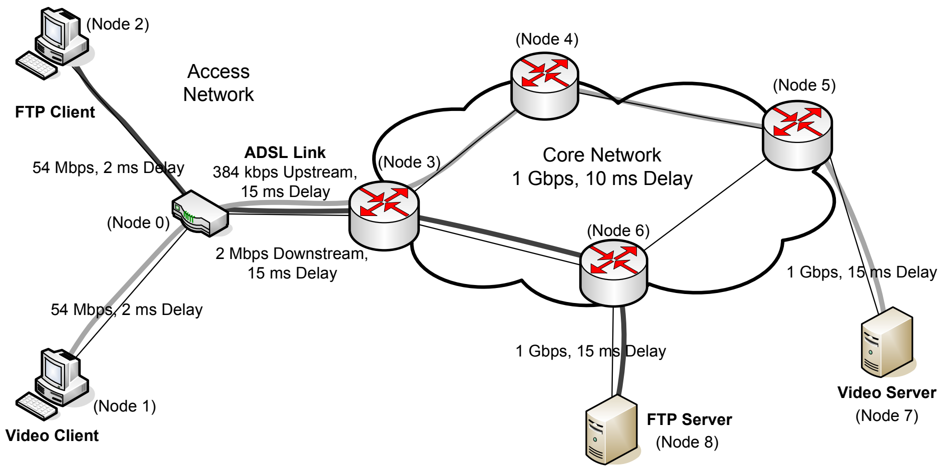


Fig. 6 Simulation topology

limit constraints restrict the video resolution to be less or equal than the display size of the client. For selecting the final adaptation parameters among the feasible TID, DID, and QID combinations, an optimization constraint is used. The maximization constraint facilitates the best bandwidth utilization by maximizing the bit rate. This means that those adaptation parameters are finally selected which lead to a bit rate that is closest but still below the transmit rate as calculated by the TFRC equation.

8 Simulation Setup and Results

Based on the MPEG-21-based implementation of the model, an evaluation of the model by using the ns-2 simulator was performed. Since an ideal adaptation mechanism should be adaptive concerning the available bandwidth while being fair to other TCP streams, two scenarios were evaluated. In the first scenario, the capability of our approach in adapting to a given maximum capacity of a bottleneck link was investigated. This evaluation should demonstrate how fast the approach can adapt to available bandwidth and how stable is its behavior (e.g., oscillations etc.). In the second scenario, the fairness of our cross-layer approach against 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 suffer 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 of the simulation, 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 part of the figure is made up of a router and two attached client nodes. The core network shown in the right part includes the core routers and two server nodes attached to it. Access and core networks are interconnected via an asynchronous digital subscriber line (ADSL).

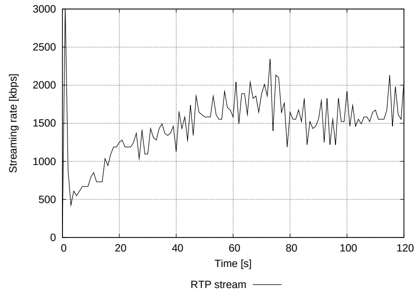
In more detail, the network consists of:

- *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* comprising four nodes with 1 Gbps bandwidth capacity and 10 ms delay among each other. 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* comprising 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 which was determined via the ping tool.

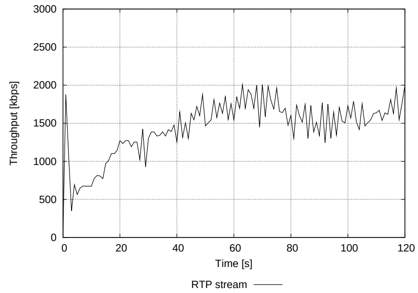
For both scenarios, a scalable video stream was used which bit rate can be adapted in the range of 90 to 3000 kbps. Each simulation run lasts 120 seconds which turned out to be a sufficiently long period to get representative results. The first-in first-out (FIFO) queue of the router interconnecting core and access networks (Node 3) is configured with a drop tail policy and a maximum queue size of 50 packets. As described above, the bottleneck link between core and access networks limits the available bandwidth for the video stream to a maximum of 2 Mbps. As depicted in Figure 6, two lines indicate 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 flow of the competing TCP flow in the second scenario.

8.2 Scenario 1 - Bandwidth Adaptation

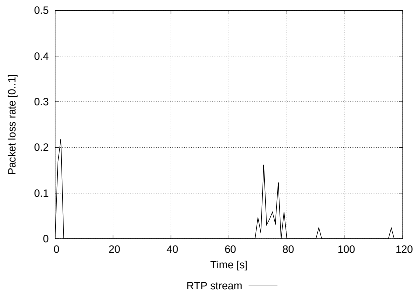
This scenario was developed to find out how the limited bottleneck link impacts the performance of the rate control approaches under investigation. Since the highest possible bit rate of the video stream is 3 Mbps and the available bandwidth between the video server and the client is limited to 2 Mbps, an adaptation has to be performed to avoid or minimize packet loss but to still efficiently utilize the network link.



(a) Streaming rate at the video server



(b) Throughput at the receiver



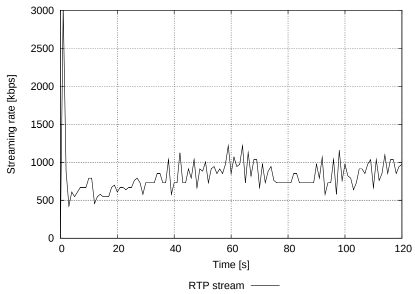
(c) Packet loss rate

Fig. 7 Simulation results of scenario 1

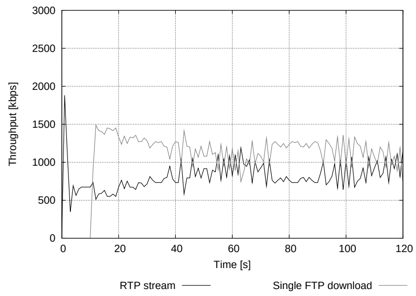
Our proposed MPEG-21-based TFRC approach utilizes an Adaptation QoS description based on the TCP throughput model to calculate the transmit rate. As one can see in Figure 7, our approach clearly shows its ability to successfully adapt the transmit rate to the bandwidth limit. At the beginning of the 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 it takes about 63 seconds 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 200-300 kbps and therefore they can be considered to do not have a major impact on the video streaming process.

8.3 Scenario 2 - Competing FTP Download

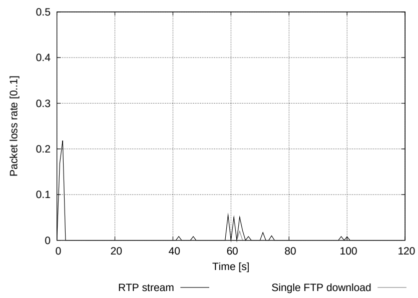
In this scenario, the video stream competes with a single FTP connection for the available bandwidth. This is the case in an environment in which one user is downloading a file from the Internet while another user in the same access network concurrently consumes a video stream. The FTP protocol was chosen 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 and not so much by the initial slow-start phase. Therefore, this scenario is ideal to demonstrate 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 [16] and a segment size of 1450 bytes for the transmitted data packets.



(a) Streaming rate at the video server



(b) Throughput at the receiver



(c) Packet loss rate

Fig. 8 Simulation results of scenario 2

According to Figure 8, the TFRC-based approach demonstrates its TCP-friendliness when competing with an FTP download. After a short, but aggressive initial phase, the video is streamed with approximately 650 kbps, that is already 32% of the available bandwidth. The FTP download initiated after 10 seconds influences the streaming rate just slightly, before the throughput of the two flows start to converge. However, it takes about 50 seconds until the bandwidth is fully utilized. Presumably, this careful behavior is caused by the conservative RTT estimate (2 $\times$ one-way delay) which indicates 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 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 being performed in a cross-layer fashion, considering both application layer characteristics (e.g., device properties) and the current network status (e.g., packet losses). On the client, the information from different layers is integrated into an MPEG-21 UED and sent back to the server for dynamic adaptation (layer extraction), FEC-based protection, and RTP packetization of 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 fast 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 on changes, respectively.
- 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 queueing 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. The client unfortunately cannot get information about the employed techniques for signalling it back to the server for adaptation.
- Most importantly, it was disclosed that most of the packets were delayed and discarded as the result of an AP's queueing strategy, not on 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 increased sending bit rate and would only increase the congestion of the link.

We thus turned our attention to 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 well adapt to available link bandwidth by using feedback provided by the client. Besides, the simulation shows that the video stream shows a fair behaviour when sharing the link with other TCP streams. This is considered as an important issue for 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 the video streaming according to network conditions and thus to enhance the experience of the content consumer. As a result of the layered encoding, the adaptation and processing of the video at the server can be performed computationally cheap and can be done on-the-fly for each individual client. Second, the actual control of the adaptation is facilitated by 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 decreases deployment complexity as compared to other cross-layer approaches. Additionally, the use of normative metadata provides an interoperable solution for cross-layer optimizations.

Acknowledgements 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-enthrone.org.

References

1. DANAE project. URL: http://danae.rd.francetelecom.com
2. Darwin Streaming Server Homepage. URL: http://developer.apple.com
3. ENTHRONE project. URL: http://www.ist-enthrone.org
4. gpac Open Source Project. URL: http://gpac.sourceforge.net
5. libxslt XSLT processor Homepage. URL: http://xmlsoft.org/XSLT/
6. Streaming Transformations for XML (STX). URL: http://stx.sourceforge.net
7. The Network Simulator ns-2: Documentation. URL: http://www.isi.edu/nsnam/ns/ns-documentation.html
8. Ahmed, T., Mehaoua, A., Boutaba, R., Iraqi, Y.: Adaptive Packet Video Streaming Over IP Networks: a Cross-Layer Approach. IEEE Journal on Selected Areas in Communications (JCAS) **23**(2), 385–401 (2005)
9. Bansal, D., Balakrishnan, H., Floyd, S., Shenker, S.: Dynamic behavior of slowly-responsive congestion control algorithms. In: Proceedings of SIGCOMM 2001, pp. 263–274. San Diego, CA, USA (2001)
10. Burnett, I., Koenen, R., Pereira, F., Van de Walle, R. (eds.): The MPEG-21 Book. Wiley (2006)
11. Choi, L., Kellerer, W., Steinbach, E.: On Cross-Layer Design for Streaming Video Delivery in Multiuser Wireless Environments. EURASIP Journal on Wireless Communications and Networking **2006**, 1–10 (2006)
12. Djama, I., Ahmed, T., Nafaa, A., Boutaba, R.: Meet in the Middle Cross-Layer Adaptation for Audiovisual Content Delivery. IEEE Transactions on Multimedia **10**, 105–120 (2008)
13. Dunn, J., Neufeld, M., Sheth, A., Grunwald, D., Bennett, J.: A practical cross-layer mechanism for fairness in 802.11 networks. Mobile Networks and Applications **11**(1), 37–45 (2006)
14. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicast applications. In: Proceedings of SIGCOMM 2000, pp. 43–56. Stockholm, Sweden (2000)
15. Floyd, S., Handley, M., Padhye, J., Widmer, J.: TCP-friendly congestion control for unicast applications. Presentation (2000). URL: http://www.aciri.org/mjh/slides/tfrc-sigcomm.ps
16. Floyd, S., Henderson, T.: The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582 (1999). URL http://www.ietf.org/rfc/rfc2582.txt
17. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ ACM Transactions on Networking **1**(4), 397–413 (1993)
18. 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 Transactions on Circuits and Systems for Video Technology **17**(12), 1665–1678 (2007)
19. Handley, M., Floyd, S., Padhye, J., Widmer, J.: RFC 3448: TCP Friendly Rate Control (TFRC): Protocol Specification (2003). URL http://www.ietf.org/rfc/rfc3448.txt
20. Handley, M., Floyd, S., Padhye, J., Widmer, J.: Internet draft rfc3448bis-06b: TCP Friendly Rate Control (TFRC): Protocol Specification (2008). URL http://www.icir.org/floyd/papers/draft-ietf-dccp-rfc3448bis-06b.txt

21. 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), pp. 199–210. Durham, UK (2000)
22. 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, Italy (2005)
23. Kawadia, V., Kumar, P.: A Cautionary Perspective on Cross Layer Design. IEEE Wireless Communications Magazine **12**(1), 3–11 (2005)
24. Khan, S., Peng, Y., Steinbach, E., Sgroi, M., Kellerer, W.: Application-driven cross-layer optimization for video streaming over wireless networks. IEEE Communications Magazine **44**, 122–130 (2006)
25. 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), pp. 63–68. Braunschweig, Germany (2008)
26. 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, CA, USA (2007)
27. 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, France (2003)
28. Malli, M., Ni, Q., Turletti, 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, France (2004)
29. Mukherjee, D., Delfosse, E., Kim, J., Wang, Y.: Optimal Adaptation Decision-Taking for Terminal and Network Quality of Service. IEEE Transactions on Multimedia **7**(3), 454–462 (2005)
30. Ni, Q., Romdhani, L., Turletti, T.: A Survey on QoS Enhancements for IEEE 802.11 Wireless LAN. Journal on Wireless Communications and Mobile Computing **4**, 547–566 (2004)
31. 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, pp. 303–314. ACM Press, New York, NY, USA (1998)
32. Qiong, L., van der Schaar, M.: Providing adaptive QoS to layered video over wireless local are networks through real-time retry limit adaptation. IEEE Transactions on Multimedia **6**(2), 278–290 (2004)
33. 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, Austria (2008)
34. Rejaie, R., Handley, M., Estrin, D.: Quality Adaptation for Congestion Controlled Video Playback over the Internet. In: Proceedings of SIGCOMM 1999, pp. 189–200. Cambridge, MA, USA (1999)
35. 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, vol. 3, pp. 1337–1345. New York, NY, USA (1999)
36. Rosenberg, J., Schulzrinne, H.: An RTP Payload Format for Generic Forward Error Correction. RFC 2733 (1999). URL http://www.ietf.org/rfc/rfc2733.txt
37. van der Schaar, M., Shankar, N.S.: Cross-layer wireless multimedia transmission: challenges, principles and new paradigms. IEEE Wireless Communications Magazine **12**(4), 50–58 (2005)
38. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (2003). URL http://www.ietf.org/rfc/rfc3550.txt
39. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology **17**(9), 1103–1120 (2007)
40. Sisalem, D., Schulzrinne, H.: The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme. In: Proceedings of NOSSDAV 1998. Cambridge, England (1998)

41. 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, NC, USA (2000)
42. Timmerer, C., Hellwagner, H.: Interoperable adaptive multimedia communication. IEEE Multimedia Magazine **12**(1), 74–79 (2005)
43. 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, Finland (2008)
44. Vetro, A.: MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access. IEEE MultiMedia **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 (2008). URL http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-svc-13.txt
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 Transactions on Circuits and Systems for Video Technology **17**(9), 1227–1237 (2007)
48. XSL Transformations (XSLT) 1.0: W3C Recommendation, 16 November 1999 URL: http://www.w3.org/TR/1999/REC-xslt-19991116
49. Yan, J., Katrinis, K., May, M., Plattner, B.: Media- and TCP-friendly congestion control for scalable video streams. IEEE Transactions on Multimedia **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, pp. 319–322. Zadar, Croatia (2006)