J. Vis. Commun. Image R. xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Design options and comparison of in-network H.264/SVC adaptation ☆

Robert Kuschnig, Ingo Kofler, Michael Ransburg, Hermann Hellwagner *

Institute of Information Technology, Klagenfurt University, Universitätsstraße 65–67, 9020 Klagenfurt, Austria

ARTICLE INFO

Article history:
Received 15 December 2007
Accepted 18 July 2008
Available online xxxx

Keywords:
Scalable video coding (H.264/SVC)
In-network adaptation
RTP/RTSP MANE
MPEG-21 Digital Item Adaptation (DIA)
Generic Bitstream Syntax Description (gBSD)

ABSTRACT

This paper explores design options and evaluates implementations of in-network, RTP/RTSP based adaptation MANEs (Media Aware Network Elements) for H.264/SVC content streaming. The obvious technique to be employed by such an adaptation MANE is to perform SVC specific bitstream extraction or truncation. Another mechanism that can be used is description (metadata) driven, coding format independent adaptation based on generic Bitstream Syntax Descriptions (gBSD), as specified within MPEG-21 Digital Item Adaptation (DIA). Adaptation MANE architectures for both approaches are developed and presented, implemented in end-to-end streaming/adaptation prototype systems, and experimentally evaluated and compared. For the gBSD based solution, open issues like the granularity of bitstream descriptions and of bitstream adaptation, metadata overhead, metadata packetization and transport options, and error resilience in case of metadata losses, are addressed. The experimental results indicate that a simple SVC specific adaptation MANE does clearly outperform the gBSD based adaptation variants. Yet, the conceptual advantages of the description driven approach, like coding format independence and flexibility, may outweigh the performance drawbacks in specific applications.

1. Introduction and motivation

Today, multimedia content is accessible on diverse end devices through a multitude of networks. Content consumers desire to retrieve content not only in the best supported quality, but also to have their personal usage preferences be taken into account. This requires content providers to offer multimedia content tailored to a wide variety of possible usage contexts, in order to maximize the Quality of Experience (QoE) of the individual content consumer. So far, content and service providers have mostly relied on the stream selection paradigm to address the variety of usage contexts. This means that multiple variations of the same content are stored in different qualities and separately offered for download or streaming. However, this is inefficient since each content variation demands for additional hard disk space. Furthermore, it is unrealistic to assume that a variation for each possible usage context can be provided. Rather, the content variations represent approximate reactions to the usage contexts which might be encountered. If a variation does not quite fit the specific usage context of a content consumer, the QoE of the user will be suboptimal.

Still, this approach works for pre-stored content, e.g., in a video on demand application; yet, it does not work well for live content, i.e., for low-delay applications. Multiple content variations usually cannot be produced (encoded) in real time and offered for consumption in various usage contexts. In this paper, the latter, more difficult case of (almost) live content streaming to heterogeneous usage contexts is assumed. An example application is a video conferencing system where the multi-point control unit has to cope with the diverse devices and networks of the participants. Another example is an application, e.g., in a soccer or tennis stadium, that serves almost instantaneous, on-site, individual requests from the personal devices of users for replays of exciting scenes of the ongoing match. While transcoding is being used in such applications, this technique has inherent problems like noticeable delay being added, particularly under heavy load, or quality degradations and mismatches being introduced for specific end devices.

In order to better serve such situations and applications, the developers of new media codecs have attempted to integrate *adaptation* support into the codecs. Such scalable media codecs enable the creation of degraded versions of an original media bitstream by simple removal of bitstream segments. Depending on which segments are removed, the adapted version can represent a lower quality in one or more scalability dimensions. For video, these dimensions are typically: temporal scalability (various frame rates), spatial scalability (various spatial resolutions), and SNR scalability (various quality levels). One prominent example of such a scalable media codec is the Scalable Video Codec (SVC) [37] which was recently standardized as an amendment of MPEG-4 Advanced Video Codec (AVC)/ITU-T H.264 by the Joint Video Team (JVT).

Scalable codecs thus in general provide a good basis for efficiently adapting the media content to diverse usage contexts that may even change dynamically. However, in a streaming scenario

☆ This work was supported by the Austrian Science Fund (FWF) under project "Adaptive Streaming of Secure Scalable Wavelet-based Video (P19159)" and by the EC in the context of the ENTHRONE II project (IST-1-507637).

* Corresponding author.
E-mail address: hermann.hellwagner@uni-klu.ac.at (H. Hellwagner).

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.07.004

there are several options for actually deploying SVC and SVC-based content adaptation. The options range from simple server–client architectures to more complex delivery architectures involving several adaptation nodes located along the content delivery path [17] [20]. Such architectures aim at minimizing adaptation delay by placing adaptation nodes close to the location where dynamically changing usage environments are expected, e.g., in the wireless access networks of the end consumers. Another aim of such in-network adaptation architectures is to save bandwidth in scenarios where multiple consumers wish to consume the same content. In such scenarios, a single SVC stream is delivered to the access network of the content consumers and only there it is replicated for, and adapted to, the usage environment of each consumer, thus saving bandwidth in the core network. Due to these potential benefits, this paper investigates SVC-based video adaptation in such a *mid-network adaptation* node, e.g., in a gateway or a dedicated content adaptation node.

A standardized way of transporting multimedia data is to use the Real-time Transport Protocol (RTP) [22]. In general, adaptation of RTP streams is not achievable without an RTP translator or mixer, because missing packets might have a negative impact on the streaming process [35]. Therefore special considerations have to be made before looking at adaptation in detail. Section 3 will show how to cope with these problems and how to enable in-network adaptation based on RTP, basically proposing an adaptation MANE (Media Aware Network Element) as introduced in [33].

In order to signal the rich scalability options offered by SVC and to enable adequate content adaptation w.r.t. the actual usage context, it is advantageous to base the adaptation process on *content related metadata* describing the bitstream and its scaling facilities (bitstream syntax descriptions). Such a metadata driven approach is considered here and explored in some detail. Usually, metadata is codec specific; however, within MPEG-21, a codec agnostic way to describe multimedia content was developed. This option is based on generic Bitstream Syntax Descriptions (gBSD). This technique will be briefly presented in Section 3.3, as applied to SVC.

Metadata can be provided in-band (e.g., within the video bitstream) or external as a description of the bitstream. While the former option was taken into account in the SVC design, i.e., in the system and transport interface, transmitting the content related metadata via a separate (RTP) channel is a valid approach as well, as adopted in [17], for instance. In this paper, both approaches will be considered. The transport mechanisms for metadata and the usage for content adaptation in the two basic adaptation MANE variants will be discussed in Sections 4 and 5.

Thus, in summary, this paper aims to comprehensively investigate metadata driven in-network adaptation in the context of H.264/SVC and RTP in order to find a best practice for provisioning SVC content in dynamically changing usage environments, and to contrast the description driven adaptation approach to a simple, SVC specific one. Several design options for describing an SVC bitstream and its scalability options, for reducing metadata (gBSD) overhead, for packeting and transporting the metadata, and for performing the adaptation process, are presented and evaluated, using prototype SVC streaming and adaptation systems.

The paper provides an evaluation of the adaptation approaches under consideration and the prototype systems in three respects: a general discussion of strengths and weaknesses of the SVC specific and the gBSD metadata driven techniques in Section 6; a specific investigation of the gBSD approach and optimizations in terms of metadata overhead, metadata packetization and transport, and error resilience in case of metadata losses in Section 7; and an extensive evaluation of the runtime behavior (delay, jitter, and computational load induced) of the prototype implementations in Section 8.

The experimental results indicate that the gBSD based adaptation MANEs are notably inferior in performance (delay, jitter, computational load) to a simple SVC specific adaptation MANE, by factors of roughly three to five times higher delay introduced and three to five times higher computational load induced on the adaptation MANE.

Still, it was found that gBSD based adaptation can well be performed in real time for up to 60 Mbps of throughput (30 parallel streams of approx. 2 Mbps bit rate each) on a desktop computer. As for delay jitter in that situation, more than 99% of the frames encounter an additional delay due to gBSD based adaptation of less than two frame times.

We conclude that the additional performance penalty of description driven adaptation can be tolerated by applications that can benefit from the conceptual advantages of the gBSD approach, e.g., coding format independence, flexibility, and semantic adaptation facilities.

2. Related work

DANAE[1] (**D**ynamic and distributed **A**daptation of scalable multimedia co**N**tent in a context-**A**ware **E**nvironment) was a 30-month EU IST project. The various results of the project [17] include a codec-agnostic adaptation MANE based on MPEG-21 gBSD [20] which was delivered as a separate RTP stream as described and evaluated in [19]. Additionally, [38] details the SVC encoder implemented in DANAE and describes the application of an unequal erasure protection mechanism for improved robustness. While DANAE constitutes the foundation for the mechanisms implemented and evaluated in this paper, our current work proposes several advances: it (1) facilitates and evaluates the RTP SVC payload format [35], (2) quantitatively compares the gBSD based adaptation approach with the SVC specific adaptation approach, (3) provides detailed results w.r.t. delay and error resilience, and (4) evaluates new mechanisms for metadata organization, compression, and transport.

In [4] [5], an alternative description driven adaptation approach for SVC content is described. While this approach also relies on the DIA framework [27], the descriptions are built using the Bitstream Syntax Description Language (BSDL) and specify SVC bitstreams on a more detailed level than our gBSD descriptions. A complete, optimized framework for BSDL based processing and adaptation is proposed. Yet, the system is destined to operate on the server before content streamout, rather than performing dynamic, in-network adaptation which is the focus of our work.

Wang et al. [31] present the system interface of SVC. This work introduces, among others, the NAL and SEI mechanisms which are the foundation for the adaptation and media-specific metadata transport mechanisms introduced in our paper. Additionally, Wenger et al. [34] present the transport interface and signaling facilities of SVC, including the SVC RTP payload format [35] which our work relies on. Different types of RTP based network elements are described, including the concept of a *mixer* which is the foundation of the design of our adaptation MANE. However, that paper does not report on implementations and evaluations of such network nodes that scale SVC streams.

Several other papers in [36] deal with SVC adaptation as well, however mostly on the basis of bitstream extraction or truncation.

The work on receiver driven layered multicast [14] was the first concept for delivering scalable (layered) multimedia content to clients using RTP and relying on IP multicast, where each layer is transported in its own IP multicast group. As discussed in [34], practical constraints (NAT and firewalls) lead to the concept of a MANE, a "middlebox" in the network that aggregates for each client one or more layers into a single RTP stream tailored to the client's requirements. The MANE architecture of our paper represents

[1] DANAE, http://danae.rd.francetelecom.com

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

such a *mixer* device, yet with the difference that we do not assume IP multicast facilities or IP multicast requirements; rather, for simplicity, since the focus of the paper is on exploring the options for the actual adaptation process on the MANE, a unicast scenario or application-layer multicast is assumed.

In [10], we describe a concrete SVC adaptation MANE implementation (an application-layer proxy) on a Wireless LAN access point, e.g., for deployment in a home scenario. Only SVC specific adaptation is performed on that device.

3. RTSP/RTP-based in-network SVC content adaptation

RTP challenges in-network adaptation in many different ways. We will provide an overview of how in-network SVC adaptation can be realized using RTSP/RTP [22] [23]. The main component is an RTSP signaling-aware RTP mixer [22], similar to the MANE concept defined in [33]. An RTP mixer, as shown in Fig. 1, acts as endpoint for the incoming RTP streams and creates new outbound RTP streams with a different Synchronization Source (SSRC). Due to this decoupling of the RTP streams, the processing/adaptation on the RTP mixer does not lead to inconsistent RTCP sender and receiver reports for both sides (server–mixer, mixer–client) and does not introduce gaps in the RTP packet sequence numbers.

The requirements on such an RTSP signaling-aware RTP mixer (MANE) can be summarized as follows:

- *Endpoint functionality.* The MANE acts as an endpoint to the server and creates new RTP streams for the client.
- *Signaling awareness.* The MANE needs to listen to the RTSP communication to identify which RTP streams contain adaptable content or metadata.
- *Stateful operation.* State has to be associated with the RTSP communication/sessions and is needed for the processing of the RTP streams.
- *Security context.* The MANE has to be in the security context, otherwise it will not be able to listen to the RTSP signaling.

A detailed explanation of the MANE architecture designed for the investigations in this paper is given in the following subsection.

3.1. Adaptation MANE architecture

Our SVC adaptation MANE (Fig. 2) acts as an RTP mixer, which receives and delivers the video data in a single unicast RTP stream. It receives the RTSP request from the client and creates a new RTSP request for the actual RTSP/RTP server. The server returns a description of the RTP streams utilizing the SDP protocol [7]. Based on the RTSP session and SDP information, new state is created on the MANE, which is used in the RTP mixing process. The mixing includes full de-packeting of the incoming RTP streams and processing/adaptation on bitstream level. After adaptation the bitstream is packed with a new SSRC and delivered to the client. Thus, the actual processing/adaptation is performed on the application layer, not on the network layer. In the following, the components of the architecture are introduced.

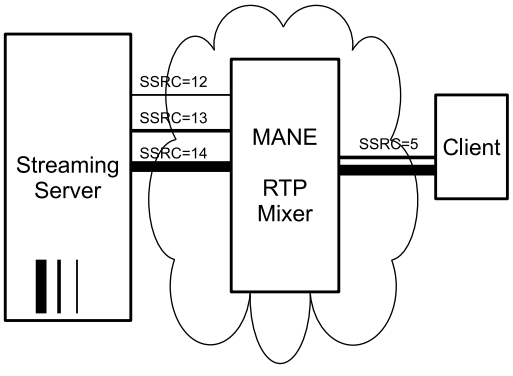


Fig. 1. Simple RTP mixer.

The *bitstream level adaptation* component is exchangeable and enables easy replacement of the adaptation mechanism. It is steered by the *adaptation decision taking engine (ADTE)*, which supplies the information how to actually adapt the media bitstream. Adaptation decision taking will not be considered in this paper, but in general it takes the user preferences, network conditions, terminal capabilities and natural environment descriptions into account. Previous work by the authors on ADT can be found in [11]. Similarly to the ADTE, discussions on session setup and control using RTSP and RTCP as well as on collecting and transmitting usage environment information are beyond the scope of this paper. Some of these aspects are addressed in [10].

The *RTP de-packetizer and packetizer* are standard RTP components for handling aggregation or fragmentation of the transferred content. In the RTP payload format for H.264/SVC [35], single- (STAP) and multi-time aggregation packets (MTAP) as well as fragmentation units (FU) are defined. STAP or MTAP is needed if a NAL unit is much smaller than the maximum transmission unit (MTU). This would result in small RTP packets and cause significant overhead, because the packet header is large in comparison to the transferred payload data. By aggregating several NAL units into a single RTP packet, we can mitigate this problem. When a NAL unit does not fit into a single RTP packet, FUs are used to split the NAL unit into several parts each fitting into a single RTP packet. Because of the full de-packeting/packeting inside our RTP mixer (MANE), the adaptation process can be unaware of H.264/SVC RTP aggregation/fragmentation modes. This simplifies the implementation and cancels out the adaptation restrictions incurred by the aggregation modes (i.e., STAP RTP packets may restrict adaptation options to only temporal ones), but introduces processing delay and additional load on the MANE.

The *access unit (AU) aggregator and fragmenter* are needed to be compliant to the RTP marker bit semantic. Only the last packet of an AU should have the marker bit set, so in general it has to be updated after adaptation. Before packeting the adapted AU, the AU has to be fragmented into pieces that the packetizer understands. (In case of H.264, these are NAL units.)

3.2. SVC-specific adaptation

In this subsection, we will provide a brief introduction of SVC with a focus on its adaptation features. SVC-specific adaptation mainly relies on the Network Abstraction Layer Unit (NALU) header which co-serves as the header of the SVC RTP payload format. For an in-depth discussion of SVC, the reader is referred to [36].

3.2.1. Video coding layer

Similarly to other video codecs, a video encoded with SVC consists of a sequence of pictures, i.e., access units (AUs). Each AU can be further divided into coded slices, e.g., for scalability reasons as explained below. Each AU therefore contains all data which is necessary to decode exactly one picture. There are basically three different types of pictures. SVC adopts the concepts of intra-, predictively- and bi-predictively-coded pictures from AVC including hierarchical B pictures as introduced in [24], which enables temporal scalability by dropping the leaf bi-predictively-coded pictures.

In addition to the temporal dimension, SVC content can also be scaled in the spatial dimension. That is, different spatial resolutions can be embedded in the same bitstream, e.g., Common Intermedi-

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

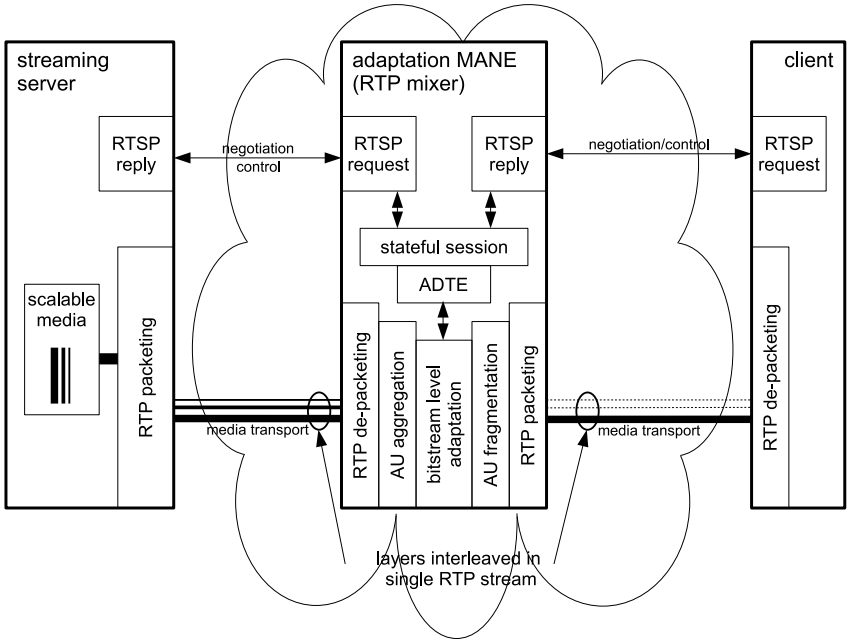


Fig. 2. Adaptation-enabled MANE based on RTSP/RTP.

ate Format (CIF, 352 × 288 pixels) and 4× Common Intermediate Format (4CIF, 704 × 576 pixels). This is achieved by encoding pictures as multiple coded slices, e.g., in our example above, the first coded slice contains all information to decode the picture at CIF resolution and the second coded slice contains the additional information needed in order to decode the picture at 4CIF resolution. This enables to easily reduce the spatial resolution by simply disregarding all coded slices belonging to the 4CIF layer. The same mechanism can be used to achieve scalability in the quality dimension. However, in this case the additional information of the second coded slice is not used for upsampling the picture to 4CIF resolution, but rather to enhance the visual quality (i.e., reduce the number of visual artifacts) for the CIF resolution. This type of scalability is referred to as Coarse Grained Scalability (CGS).

All three scalability dimensions have in common that between any two switching pictures[2] only complete layers can be dropped, i.e., the number of layers to be removed may only be changed at those switching pictures. For scalability in the quality dimension, a more fine granular way of scalability was desired. Therefore, Medium Grained Scalability (MGS) was introduced. MGS is performed in the same manner as CGS, however with the difference that MGS coded slices can be individually removed, i.e., it is not needed to remove the complete layer.

However, in order to selectively decide which coded slice belongs to which temporal, spatial, or quality layer, this information needs to be added to the coded slice, which is the aim of the Network Abstraction Layer (NAL), as discussed in Section 3.2.3.

3.2.2. Parameter sets and supplemental enhancement information

Parameter Sets (PSs) and Supplemental Enhancement Information (SEI) messages do not contain coded video data. A PS contains information which applies to a large number of coded slices of a specific layer, where it would be inefficient to encode this information for each coded slice. The spatial resolution of a video segment of a specific layer is an example of information which is included in a PS.

SEI messages provide supplemental data which is not necessary for the decoding process, but which may be helpful for the processing of the bitstream, like timing information for the playout at the client. Scalability SEI messages carry layer boundary information which indicates the highest values of temporal level, quality level, and dependency id (see Section 3.2.3) for all coded slices of the media stream. Additionally, they contain bit rate information for each layer of the scalable stream. There are special SEI messages defined in the H.264/SVC standard for carrying *user data*. For large scale deployments it is possible to register these specific SEI messages globally [37].

3.2.3. Network abstraction layer

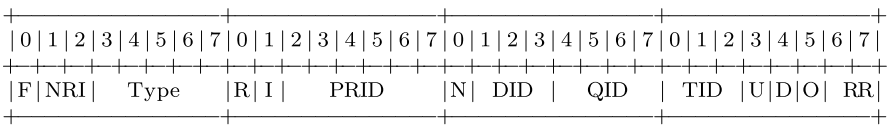
In order to—among other things—selectively decide which coded slices belong to which layer, the so called Network Abstraction Layer Unit (NALU) header is added in front of each coded slice. The resulting coded slices are called Network Abstraction Layer Units (NAL Units or NALUs). The NALU header with SVC-specific extensions is shown in Listing 1.

In the following, we focus on those fields of the header which are important regarding adaptation. These are the priority id (PRID), temporal id (TID), dependency id (DID), quality id (QID), and the discardable flag (D):

- *Priority id* is a 6 bit field which provides an application-specific priority setting.
- *Dependency id* is a 3 bit field which provides the inter-layer dependency for CGS and spatial scalability. NALUs with a higher DID can depend on NALUs with a lower DID, but never the other way around.
- *Quality id* is a 4 bit field which provides the quality level of an MGS NALU. Similar to above, MGS NALUs of a higher level depend on MGS NALUs of a lower level, thus the highest level(s) can be removed for quality scalability.
- *Temporal id* is a 3 bit field which provides the temporal level of the current NALU. The same rules as above are valid, i.e., the highest level(s) are to be removed first for temporal scalability.

[2] Switching pictures can be I pictures or P pictures which are specially encoded to allow layer switching. For more information the interested reader is referred to [9].

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004



Listing 1. NALU header with SVC-specific extensions.

- *Discardable flag* is a 1 bit flag which indicates whether the current NALU is needed for decoding NAL units of the current picture. Additionally, if set, this NALU is not needed by any other NALU in subsequent pictures which have a greater DID than the current NALU, i.e., such NAL units can be discarded without risking the integrity of higher layers with greater DID.

3.3. gBSD-based adaptation

In this subsection, we will provide a brief introduction to MPEG-21 based, description driven adaptation with a focus on enabling coding format independence. For an in-depth introduction into this topic, the reader is referred to [2] [16] [29] [30].

MPEG-21 Digital Item Adaptation [27] provides a number of normative description formats, among which the so called *generic Bitstream Syntax Description (gBSD)* specification is relevant here. A gBSD is an XML document which describes a (scalable) multimedia bitstream enabling its adaptation in a codec agnostic way. Only the high-level bitstream structure is described, i.e., how it is organized in terms of packets, headers, or layers. The level of detail of this description depends on the scalability characteristics of the bitstream and the application requirements. Listing 2 shows a gBSD for an SVC access unit (frame). Each NALU of the SVC content is described by a *gBSDUnit*, which provides the NALU's *length* in bytes. (Addressing of the NALUs is done consecutively in this example, starting from the *start* position of the frame.) Additionally, the *marker* attribute indicates the TID, DID, and QID values as described above. This gBSD also provides timing information for the gBSD itself and the described SVC access unit, which is used to synchronize them in dynamic and distributed adaptation scenarios [18].

In the course of content adaptation, the gBSD of a media bitstream is transformed first, followed by the generation of the adapted bitstream from the original one, guided by the transformed gBSD. Adaptation will mainly comprise simple remove operations (of gBSDUnits and bitstream syntax elements) as well as some update operations (of address information, for instance) to keep the adapted bitstream standard compliant. The gBSD transformation can be performed by an XSLT style sheet [39], which is provided during session setup. Thus, the adaptation process is moved into the domain of the codec agnostic gBSD, which enables the actual adaptation to be independent from the coding format. That is, an existing adaptation MANE could accommodate any current or future scalable coding format if a valid gBSD is transmitted with the bitstream.

The gBSD-based adaptation mechanism was originally intended for static, server-based adaptation but was recently extended to support dynamic and distributed adaptation scenarios [17] [20]. As such it will be evaluated in this paper as an alternative to the codec-specific adaptation approach introduced in Section 3.2.

4. SVC-specific adaptation MANE

Codec specific adaptation is easy to implement and features in general high performance and scalability. For our investigations, we use a simple codec specific adaptation mechanism, which is common for in-network adaptation. The adaptation is done on bitstream level (NAL units) and does not restrict the adaptation facilities. It also enables a fair comparison of the different adaptation concepts.

Our approach to adapt H.264/SVC content extends the architecture of the adaptation MANE shown in Fig. 2 by using bitstream level adaptation based on NAL units, as shown in Fig. 3. The NAL unit stream derived from the de-packetizer is aggregated into access units (AUs). After aggregation, each NAL unit of the AU is adapted with the help of the adaptation decision, which steers the adaptation process. This adaptation decision consists of several SVC-specific parameters describing which parts of the bitstream should be kept and which should be filtered out. By simply matching these param-

```
<gBSDUnit xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns="
    urn:mpeg:mpeg21:2003:01-DIA-gBSD-NS" xmlns:bs1="
    urn:mpeg:mpeg21:2003:01-DIA-BSDL1-NS" xmlns:xsi="http://www.w3.org
    /2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
    XMLSchema" xmlns:si="urn:mpeg:mpeg21:2003:01-DIA-XSI-NS" xmlns:msi="
    urn:mpeg:mpeg21:2003:01-DIA-MSI-NS" addressUnit="byte" addressMode="
    Absolute" bs1:bitstreamURI="v.264" msi:timeScale="90000"
    msi:dtsDelta="3000" si:timeScale="90000">
    <gBSDUnit start="24355" length="14590" marker="Frame" si:pts="24000"
        >
        <gBSDUnit length="9" marker="T0D0Q0"/>
        <gBSDUnit length="846" marker="T0D0Q0"/>
        <gBSDUnit length="669" marker="T0D1Q0"/>
        <gBSDUnit length="442" marker="T0D2Q0"/>
        <gBSDUnit length="1839" marker="T0D3Q0"/>
        <gBSDUnit length="1265" marker="T0D4Q0"/>
        <gBSDUnit length="2997" marker="T0D5Q0"/>
        <gBSDUnit length="1442" marker="T0D6Q0"/>
        <gBSDUnit length="5081" marker="T0D7Q0"/>
    </gBSDUnit>
</gBSDUnit>
```

Listing 2. Generic Bitstream Syntax Description example.

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

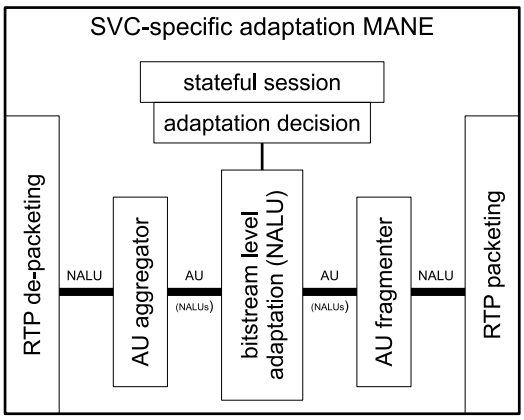


Fig. 3. SVC-specific adaptation based on NAL units.

eters with the fields of the NAL unit header, it is possible to adapt the bitstream. NAL units not matching the parameters of the adaptation decision are consecutively removed. After adaptation, the AU fragmenter analyzes the remaining NAL units in such a manner that the integrity of the marker bit is preserved. The resulting NAL units are forwarded to the packetizer and sent to the client.

This solution does not require any adaptation specific metadata and relies only on the metadata provided by the NAL unit header, which results in little processing overhead.

5. gBSD-based adaptation MANE

Using separate metadata affects the architecture of the adaptation MANE. The metadata has to be streamed and synchronized with the media data. We will now discuss how to enable gBSD-based in-network adaptation (see Section 3.3), which uses separate metadata to describe the H.264/SVC content. The main concern of this section is how to transport, signal, and synchronize this metadata with the multimedia content. The adaptation mechanism itself complies to the MPEG-21 Digital Item Adaptation standard.

We consider two different gBSD-based adaptation MANEs, which mainly differ in terms of metadata transport. Yet, initially the common metadata handling mechanisms, which are used in both architectures, are discussed.

5.1. Common gBSD metadata handling

In case of H.264/SVC, the gBSD metadata "naturally" describes access units (AUs); see Listing 2 for an example. Fig. 4 illustrates, however, that the metadata can be structured into so-called *process units (PUs)* [18], each describing either a single AU or, in an aggregation mode, multiple AUs forming a group of pictures (GOP). The metadata includes all necessary timing information needed for synchronization. As shown in Fig. 4, each timestamp (TS) of an AU is specified relative to the timestamp of the first AU the metadata describes.

When utilizing GOP-level metadata for adaptation, it is possible to split a GOP-level gBSD PU into smaller gBSD PUs, each describing only one AU. On the one hand, GOP-level aggregation of the gBSD metadata has a positive impact on compression since it reduces the metadata overhead before transmission. On the other hand, synchronization of the media data with the metadata on an AU basis and corresponding fine grained adaptation is facilitated. There is also an advantage for the error resilience of the gBSD based adaptation systems, because the synchronization and error recovery can be realized on the AU level instead of the GOP level. Detailed results on compression and error resilience performance are given in Section 7.2.

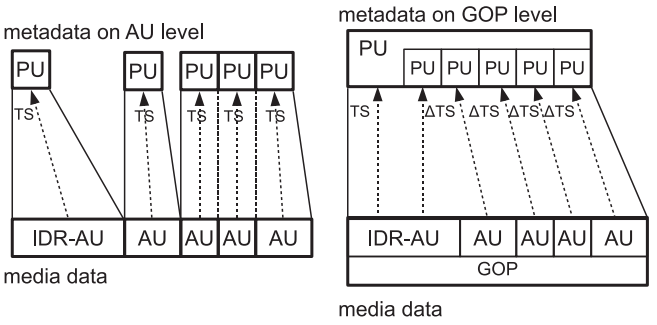


Fig. 4. Different types of metadata organization.

5.2. gBSD two-stream solution

The use of a second RTP stream is a codec-agnostic way to transport metadata. This straightforward method is easy to implement, but has some non-obvious problems which are discussed in detail later on. The architecture as shown in Fig. 5 is similar to the implementation in the DANAE project (see Section 2), but has some improvements for error resilience and runtime performance of the *bitstream level adaptation* component.

After session negotiation and setup, the server sends two separate RTP streams to the MANE, one containing the video data and the other the metadata. Metadata packetization and transport via RTP are covered in Section 7.3. The dependencies between the streams can be described by means of the SDP protocol decoding dependency extension [21]. The synchronization of the video stream and the metadata is done via the timestamps of the RTP header. This packet-accurate synchronization is compulsory for the gBSD adaptation process.

Each gBSD process unit (metadata) is split—in case of GOP process units—into several AU process units. The timestamps of the AU process units are used as input for the AU aggregation process, which aggregates NAL units with this timestamp, to be used later for the adaptation of the access unit. The gBSD-based adaptation mechanism transforms the AU process unit (gBSD) under the constraints of the adaptation decision and adapts the AU according to the standardized gBSDtoBin process [30]. After adaptation, the AU is fragmented into NAL units, the packetizer handles the NAL units, and the packets are sent to the client.

Compared to the SVC-specific adaptation based only on NAL units as described before, the overhead introduced by this solution is mainly the second RTP channel, the additional gBSD metadata, and their processing (additional channel and metadata). More details will be presented in Section 8.4.

5.3. gBSD single-stream solution

Delivering metadata inside the transported media content is not always possible. Either the multimedia codec or the RTP payload format have to enable this feature. In case of H.264/SVC, this is possible via SEI messages which were introduced in Section 3.2.2. For easy prototyping, an SEI message with a custom SEI payload type can be defined and used as a metadata container. This special SEI message will be either removed after adaptation by the MANE or ignored by the receiver/decoder in case the video stream reaches a client without passing an adaptation MANE. For large scale deployments, we would advise to use *user data SEI messages*, which can be registered globally.

Concerning transport, there is no separate handling in the RTP payload format to consider. Because the SEI message is a standard NAL unit, it will be injected into the NAL unit stream of the content and handled like a part of the video. The synchronization is done

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

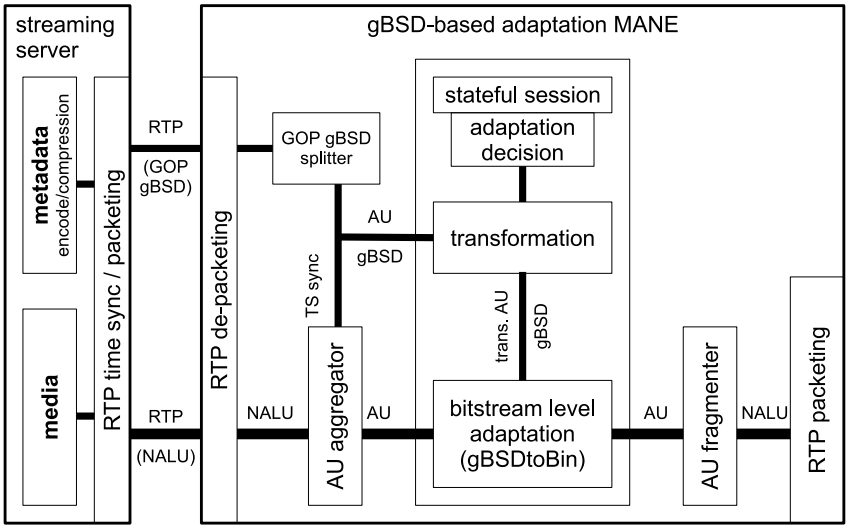


Fig. 5. gBSD two-stream solution.

in-band, which means that the SEI message receives the same timestamp as the first AU it describes and precedes this AU in the NAL unit stream. More or less, the SEI message is part of this AU and inherits all its specifics. Therefore this solution is compatible with any standard RTP implementation.

The architecture of this gBSD single-stream adaptation MANE, as shown in Fig. 6, uses the same components as the gBSD two-stream solution presented in Section 5.2. Only the metadata transport differs. The video data and the metadata can be transmitted over a single RTP channel. The adaptation MANE has only to extract the metadata out of the NAL unit stream provided by the de-packetizer. The remaining NAL units form the input of the access unit aggregator. The parts in Fig. 6 marked by a hatched area are exactly the same as in the gBSD two-stream solution. The metadata is split and transformed, the NAL units aggregated, and fed into the gBSD adaptation engine. After the fragmentation of the AU, the NAL units are packed and sent. For more details on the common functionality see Section 5.2.

This architecture inherits the problems of the two-stream solution, namely gBSD processing and metadata overhead, but simplifies the synchronization between metadata and the video data by using in-band metadata. Moreover, this approach reduces processing load by utilizing only one RTP channel.

6. Comparison of SVC-specific and gBSD-based adaptation

Adaptation of scalable multimedia content is in principle a straightforward process. (Clearly, complex tasks like content analyses enabling semantic adaptation, summarization, or personalization are beyond the scope of this paper.) For the actual adaptation process, the main issue is to choose the right adaptation method from the options available. A direct, SVC specific adaptation technique as well as a more general, gBSD metadata based approach were introduced so far. While it is possible to compare most of the properties of these methods quantitatively (see Sections 7 and 8), some of their features cannot be measured and are discussed qualitatively in this section. Such features include:

- *Flexibility*. Does metadata based abstraction of scalability features result in more or more flexible ways to adapt content?
- *Capability*. Is it possible to do adaptation based on codec unrelated or semantic parameters (e.g., violence levels of scenes)?
- *Adaptability*. How does the system/architecture react to small/evolutionary changes of media codecs/formats?
- *Extensibility*. How easily can the system/architecture be adapted to new media codecs/formats?

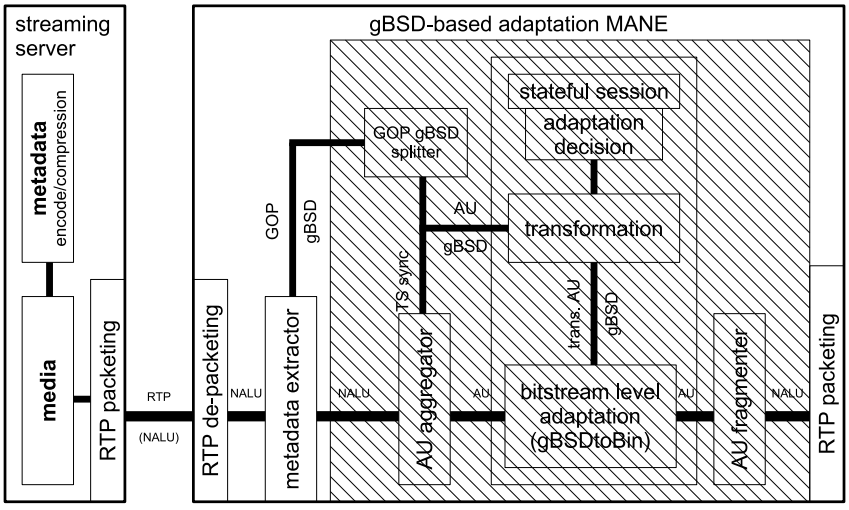


Fig. 6. gBSD single-stream solution.

- *Interoperability*. Can the metadata also be used in or combined with other systems (like MPEG-7 [8])?

The major qualitative strengths and weaknesses of the two adaptation approaches considered in this paper are listed as follows:

- *SVC specific adaptation.*
  Strengths:
  - Simple and fast adaptation process, inducing low delays only.
  - Notion of NAL units allows handling any content in a common fashion.

  Weakness:
  - Obviously only works for H.264/SVC.

- *gBSD based adaptation.*
  Strengths:
  - Codec-agnostic, i.e., works for any scalable media content properly described by metadata.
  - Semantic annotation/adaptation of content possible (e.g., violent scenes).
  - Signaling of when and how to adapt is possible, such that QoE is maximized (e.g., IDR or switching pictures).

  Weaknesses:
  - Induces metadata and processing overhead due to XML and its processing.
  - Separate synchronization of media data and metadata needed.

Apart from simple SVC specific adaptation, there is the possibility in H.264/SVC to utilize SEI messages, which can signal any custom information needed by a possibly more advanced SVC specific adaptation process. But it is obvious that such a solution suffers from the same problems as the codec agnostic approach (gBSD based adaptation). However, it has no standards basis. Therefore, such advanced SVC specific adaptation will not be discussed in this paper.

In summary, it can be concluded from these initial qualitative considerations that gBSD based adaptation has advantages in terms of flexibility, capability and functionality in general, while it can be assumed that the simpler, SVC specific adaptation system will excel in performance. The specific requirements of the application and the concrete overhead incurred by the gBSD metadata and its processing will mainly determine which adaptation approach to deploy.

7. Metadata overhead and transport analysis

In this section, several options of granularity, organization, and transport of gBSD descriptions for SVC streams are presented and compared quantitatively, since the organization of gBSD metadata has a great impact on bit rate overhead and error resilience and thus will directly affect the architecture of the gBSD based adaptation MANEs. The aims of this initial, static analysis are to identify feasible solutions for description driven adaptation and to explore ways how to deploy and improve them for in-network adaptation. These solutions will in a next step be evaluated in terms of their dynamic behavior and compared to the simple SVC specific adaptation approach (Section 8).

7.1. Test video streams

Four different well-known clips (*foreman, harbour, city, deadline*) were used as a basis for the metadata analysis and were encoded using different layer configurations and GOP sizes. The clips were generated using the Joint Scalable Video Model (JSVM) [15] 9.8 software, which offers a variety of settings that have a great influence on both the quality and the bit rate. Since our investigations focus on the adaptation mechanisms that are unaware of the quality of the actual video, we decided to configure the encoder in a way to get reasonable bit rates in the range of 800 to 2200 kbps by using constant quantization parameters.

The different layer configurations are summarized in Table 1. All of the configurations consist of eight layers which enable a broad spectrum of adaptation possibilities. For the encoding of the quality refinement layers, both coarse-grained and medium-grained scalability were used.

In addition to the different layer configurations, we encoded the clips using different GOP sizes. In the context of this paper, a GOP size is considered to be the distance between two IDR frames, e.g., a GOP size of 16 means that every 16th frame is encoded as an IDR frame. For our evaluation we considered three different GOP sizes (16, 32 and 48) and also the possibility of using only a single IDR frame at the beginning of the sequence (1 GOP). Although the usage of a single IDR frame has no practical relevance in a streaming scenario, it was regarded to be useful to measure the overhead that is imposed by a certain GOP size.

7.2. Metadata overhead

In the first step of our investigation, we focused on measuring the overhead that is introduced by the metadata. For the quantitative evaluation, we generated the gBSD for each of the 96 different coded video sequences and investigated two different fragmentation modes for the gBSD. Fragmentation is necessary since in a streaming scenario the gBSD is too large to be transmitted as a whole but has to be fragmented into process units (PUs). In the case of SVC, two granularities of fragmentation are obvious: one PU describes either a single access unit (AU) or a complete GOP. In the AU fragmentation mode, one gBSD PU describes exactly one AU with its corresponding NAL units. In the GOP fragmentation mode, all AUs that belong to the same GOP are described by a single gBSD PU.

Firstly, we investigated the bit rate of the metadata stream. It soon turned out that using plain-text encoding of the gBSD leads to a significant bit rate of the metadata stream. The AU fragmentation mode resulted in an average bit rate of approx. 260 kbps, while the GOP fragmentation led to roughly 115 kbps. The reduction in bit rate is based on the fact that the header of each gBSD PU contains verbose XML namespace declarations which cause a significant overhead when transmitted for each access unit. Another observation is that the bit rate of the metadata stream does not directly depend on the bit rate of the video stream but on the number of layers.

As a consequence of the high bit rates, the impact of compressing the metadata was investigated in more detail. The bzip2 algorithm [25] was selected for further experiments since it achieves very competitive compression ratios for gBSD metadata [28]. A comparison between the actual video bit rate and the resulting

Table 1
Layer configurations used for encoding of test video streams

Encoding set (acronym)	Base layer (quality enh.)	1st spat. enh. layer (quality enh.)	2nd spat. enh. layer (quality enh.)
cgs0	CIF@30 Hz (7 CGS)		
mgs0	CIF@30 Hz (7 MGS)		
cgs	QCIF@15 Hz (3 CGS)	CIF@30 Hz (3 CGS)	
mgs	QCIF@15 Hz (3 MGS)	CIF@30 Hz (3 MGS)	
cgs2	CIF@30 Hz (3 CGS)	4CIF@30 Hz (3 CGS)	
mgs2	CIF@30 Hz (3 MGS)	4CIF@30 Hz (3 MGS)	
cgs3	QCIF@15 Hz (1 CGS)	CIF@30 Hz (2 CGS)	4CIF@30 Hz (2 CGS)
mgs3	QCIF@15 Hz (1 MGS)	CIF@30 Hz (2 MGS)	4CIF@30 Hz (2 MGS)

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

metadata bit rate for some selected video sequences is given in Table 2. It clearly shows that the metadata overhead can be reduced significantly by bzip2 compression. The achieved compression factors for the gBSD PUs are illustrated in Fig. 7. The general observation that can be made is that the average compression factor increases with the size of the process unit. The resulting average compression factors for GOP sizes of 16, 32 and 48 were 8.1, 11.6 and 14, respectively. The right most bar in the figure represents a compression factor of about 26 that can be achieved when considering the whole sequence as one GOP and compressing the gBSD without fragmenting it. It can be concluded that the actual bit rate of the metadata can be reduced to around 10 kbps when using fragmentation on GOP granularity and compression.

An alternative method to encode the metadata is the Binary format for Metadata (BiM) [8]. When compressing the PUs with BiM, compression factors of around 10 for the AU fragmented PUs and 11 for the GOP fragmented PUs can be achieved. This relative constant result can be explained by the way an XML document is encoded using BiM. BiM compresses by encoding XML elements, attributes, and values efficiently, e.g., by using binary representations of numbers or employing variable length codes, but it does not remove redundancy within a document. Therefore, also the GOP size has no longer influence on the compression factor as it was the case for bzip2.

7.3. Metadata packetization and transport

In a next step, the overhead and the implications when transmitting the gBSD metadata over the network were investigated. For our evaluation, we considered two different approaches of transmitting the compressed gBSD PUs. The first approach is to use a separate RTP stream for the compressed metadata and to use the standard RFC 3550 RTP packeting mode [22]; this corresponds to the *gBSD two-stream solution* of Section 5.2. The second approach is to encapsulate the gBSD PUs into the SVC stream by using SEI messages; this corresponds to the *gBSD single-stream solution* above (Section 5.3). The SEI messages are then transmitted as part of the SVC bitstream and are packeted according to the IETF SVC draft [35]. In the following, both approaches will be denoted as *separate packetization* and *SEI packetization*. For all our packeting investigations, we considered an MTU of 1500 bytes as is the case in the predominant Ethernet.

The results of the evaluation can be found in Table 3. It can be seen that the GOP fragmentation clearly outperforms the AU fragmentation in terms of metadata bit rate and the resulting overhead compared to the video bitstream. Besides, it turns out that the selection of the packetization does not have a great impact on the number of additional packets that are produced during packetization. Both separate packetization and SEI packetization lead to less than 1% more packets.

Table 2
Metadata overhead of selected video sequences

Video stream (bit rates in kbps)				Metadata (bit rates in kbps)					
				AU frag.			GOP frag.		
Sequence	Encoding	GOP	Bit rate	Plain	BiM	bzip2	Plain	BiM	bzip2
City	mgs	16	806	251	24	112	106	10	14
City	mgs	32	755	251	24	112	100	9	9
City	mgs	48	742	251	24	112	98	9	7
Harbour	mgs2	16	2184	257	26	114	112	11	15
Harbour	mgs2	32	2056	257	26	114	107	10	10
Harbour	mgs2	48	2014	257	26	114	105	10	8

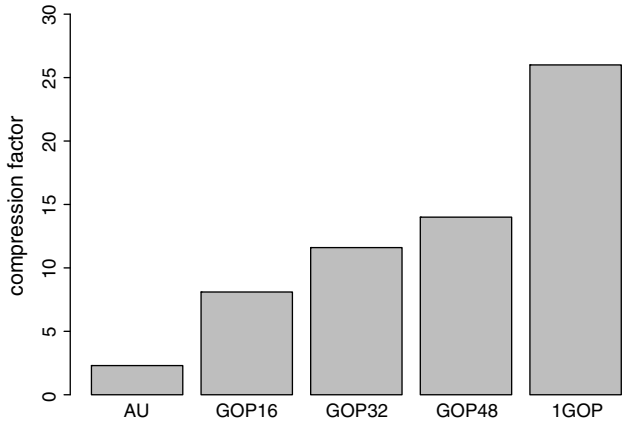


Fig. 7. Compression factors for gBSD descriptions.

7.4. Error resilience

As the GOP fragmentation mode has advantages concerning both the compression efficiency and packetization mode, it was further evaluated concerning its error resilience behavior in case of packet losses. Following the idea of the gBSD-based adaptation, a media bitstream and its corresponding gBSD description are required for the adaptation. If this adaptation is performed on a per-GOP basis, the gBSD describing the GOP and the complete bitstream of the GOP, i.e., all the AUs belonging to the GOP, have to be transmitted successfully to the adaptation MANE. For a more detailed analysis of the robustness in case of packet losses, the average number of packets per GOP was determined for each of the encoded sequences. It turned out that for the clips *foreman*, *city* and *deadline* on average 56 media packets were required to transmit a GOP consisting of 16 AUs. For larger GOP sizes of 32 and 48 AUs the average number of packets was 103 and 151, respectively. Considering the fact that for a successful adaptation more than 50 subsequent packets have to be transmitted successfully, it renders the GOP-based adaptation approach as not realizable in a real networking scenario.

In order to combine both the higher compression efficiency of the GOP fragmentation mode and the better error resilience behavior of per-AU adaptation, we propose to fragment the gBSD on a per-GOP basis, *yet* to perform adaptation on a per-AU basis. The adaptation on a per-AU basis requires that the gBSD description of the AU and all media packets belonging to the AU have to be transmitted successfully to the adaptation MANE. For investigating the error resilience of this approach, we simulated the streaming of both metadata and video content for each of the 96 sequences. For each sequence, the transmission of 50,000 AUs with packet loss ratios of 3%, 5%, 10% and 20% was investigated. The packet loss was introduced according to the loss patterns available from the ITU [32]. The probability of receiving (and adapting) a complete AU was considered as the metric for error resilience. The distribution of this probability for each of the four loss patterns is given as a box-plot in Fig. 8. As one can learn from the plot, it is still possible to adapt around 70 percent of the AUs for most of the sequences when considering a packet loss rate of 5%.

Table 3
Metadata implications on transport

Fragmentation	Metadata stream		Packet overhead	
	Bit rate (kbps)	Overhead (%)	Separate pack. (%)	SEI pack. (%)
Per AU	110	13	27	6
Per GOP	10	1–2	0.97	0.5

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

8. Performance evaluation

For a direct quantitative comparison of SVC specific and description driven in-network adaptation MANEs, the following metrics will be used:

- Transmission delays between server and clients, to a large degree determined by the delays incurred by the adaptation MANE.
- Load on, and scalability of, the adaptation MANE, in terms of CPU usage.

These metrics are directly linked to the utility of the approaches in real streaming and adaptation systems. In order to achieve results of practical relevance, we have implemented streaming and adaptation prototypes using standard technologies as described below. The implementations are affected by operating systems and network behavior, e.g., context switches and socket processing efforts, as well as by complex scheduling effects on various levels, e.g., in the streaming server or in the multimedia and networking libraries employed.

The performance results achieved are useful beyond the direct performance comparison of SVC specific and gBSD based adaptation MANEs. For instance, client jitter buffer dimensioning can make use of these results, or timed transmission of the media packets from the adaptation MANE to the clients can be additionally performed. However, such investigations are beyond the scope of the current paper.

8.1. Prototype implementations

Our prototype implementations are based on standard open source streaming technologies, namely:

- Darwin Streaming Server [1] used as an RTP/RTSP implementation.
- Live555 [13] modules used for adaptation and RTP/RTSP clients.
- GPAC [12] for server-side packeting of H.264/SVC video and metadata.
- XML processing with libxml [26], xerces-c [6] and CodeSynthesis XSD [3].

The streaming server is based on the Darwin Streaming Server and uses a custom GPAC module to packetize H.264/SVC according to the recent RTP specification [35]. It streams the metadata in plain RTP packets and uses the marker bit to enable fragmentation of the NALUs over more than one RTP packet. The adaptation MANE is also based on Darwin and uses the Live555 RTSP client to communicate with the streaming server. A Live555 module is used to perform the actual adaptation and is located between the Live555 RTP source (more or less the RTP client) and the RTP sink, which sends the adapted video stream to end-user client. The end-user client is Live555's openRTSP, a simple command line client. All implemented in-network adaptation MANEs have the same software basis, only the Live555 source–adaptation module–sink chain differs, as explained generally in Sections 4 and 5.

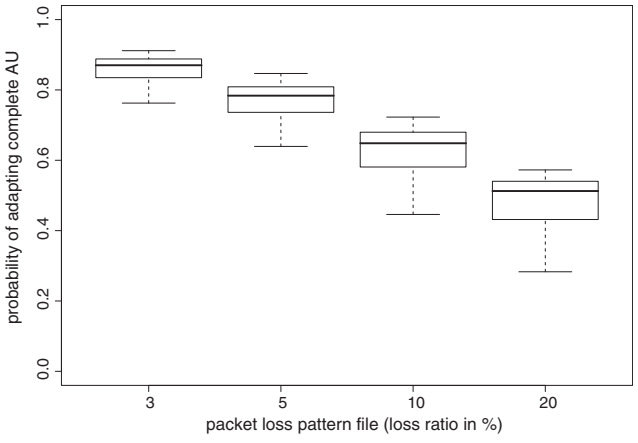


Fig. 8. Error resilience of per-AU adaptation.

8.2. Packet-forwarding MANE

In addition to the MANEs described in Sections 4 and 5, a simple packet forwarding MANE was implemented to serve as a reference. This packet forwarder simply takes the RTP payload of an incoming packet and fills this payload into an outgoing packet. Additionally, the presentation time and marker bit of the incoming packet have to be considered when sending out the packet to the client. This MANE enables us to measure how much basic processing effort the RTP de-packeting, AU aggregation, adaptation process, AU fragmentation, and RTP packeting steps will induce, which have to be performed by any adaptation MANE.

8.3. Evaluation prerequisites and test setup

As a result of the metadata analysis of the encoded video content and the possible GOP sizes, it was concluded that a GOP size of 32 frames (GOP32) has the most benefit for the proposed streaming scenarios. Due to the metadata aggregation on a GOP basis, the metadata overhead is much lower than on an AU basis; error resilience is good because the adaptation scheme can still work on an AU basis. Hence, only GOP32 variants of the test content were considered for performance evaluation, because they give the best metadata overhead–to–error resilience ratio. We reduced the evaluated encoding sets to *mgs* and *mgs2* because they provide balanced options in the temporal, spatial, and quality adaptation dimensions, which fits best in an open adaptation scenario. The video streams used for the evaluations and their bit rates are described in Table 4.

Because all implemented adaptation MANEs have the same software basis, we can safely assume that the measurements only show the overhead due to the specific adaptation mechanism. We have now to consider one packet forwarder and three adaptation MANEs, which are the following:

- Packet forwarder (*packet forward*).
- SVC specific adaptation MANE (*SVC specific*).
- gBSD with single stream carrying both video data and metadata, termed SEI packetization in Section 7 (*gBSD 1 stream*).
- gBSD with two separate streams, termed separate packetization of media data and metadata in Section 7 (*gBSD 2 stream*).

For delay measurements, we used a special method to retrieve accurate end-to-end delays. In general, this is problematic because

Table 4
Video streams used for performance evaluation

Sequence	Encoding set (max. resolution@frame rate)	GOP size	Bit rate in kbps
City	mgs (CIF@30 Hz)	32	755
Deadline	mgs (CIF@30 Hz)	32	719
Foreman	mgs (CIF@30 Hz)	32	715
Harbour	mgs (CIF@30 Hz)	32	1281
City	mgs2 (4CIF@30 Hz)	32	1247
Harbour	mgs2 (4CIF@30 Hz)	32	2056

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

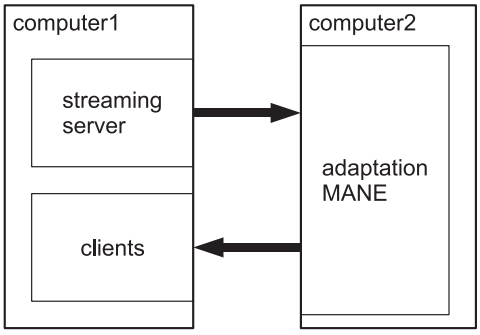


Fig. 9. Evaluation setup.

of clock synchronization issues. By simply placing the streaming server and the clients on the same computer, we overcome this synchronization problem. Also we can safely assume that the measured results will only be inferior to a solution with the server separate from the clients, because of concurrency issues. Fig. 9 shows the test setup, which consists of two DELL PowerEdge 1850 servers with two Intel Xeon 3.0 GHz EM64T processors with Hyperthreading disabled. In each computer, the main memory comprises 2 GB and the operating system is Ubuntu Linux 6.06.1 (dapper) with kernel 2.6.15 (×86_64). The servers are connected via Intel(R) PRO/1000 network cards (1 Gbps) to a Gigabit Ethernet network switch.

8.4. Evaluation results

This quantitative evaluation shows the impact of in-network adaptation on the transmission delay. In addition, the load on the adaptation MANE for multiple streams is compared for the different implementations.

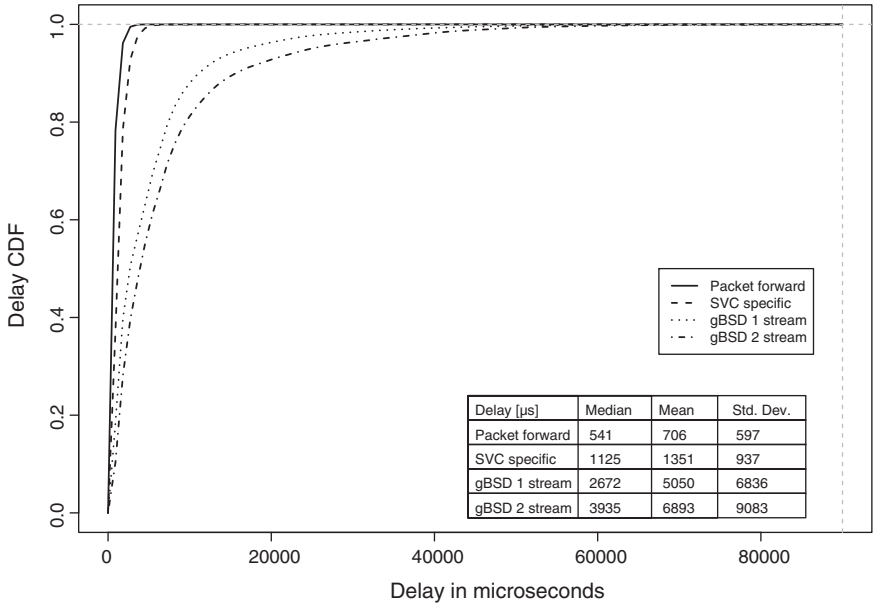
For all measurements, 30 clients were receiving the same content from the streaming server via the adaptation MANE. There was no packet loss during transmission for all clients and measurements. The first 100 seconds were removed from the result data sets, because in the startup phase the number of clients is not constant (i.e., clients are being started one after the other in this time frame) and the delay may be better than the worst case (i.e., when all clients are being served). The delay between the streaming server and the client is measured on a picture-by-picture basis. So after fully sending/receiving the last NAL unit of a picture (i.e., of an AU), the timestamp for the picture is recorded. This is done for each stream served by the streaming server and on each client. The test sequences were being played in a loop for a total of 54,000 pictures, which results in a streaming and playout time of 30 min. The adaptation process was enabled for all adaptation MANEs, but the adaptation decision was selected in such a manner that all packets are passed through. This would be the worst case scenario, because all of the data has to be handled by the adaptation MANE and transmitted to the client. In the evaluations, this "pass-through" operation for all packets is required in order to achieve correct transmission delay measurements.

To illustrate how the bit rate of the video is responsible for the processing delay of the adaptation MANEs, three representative video sequences have been selected, which are *foreman mgs*, *city mgs2*, and *harbour mgs2*. Fig. 10 shows the induced delay for the *foreman* sequence with an overall bit rate for all 30 clients of about 20 Mbps, Fig. 11 shows the delay for *city* (approx. 35 Mbps), and Fig. 12 depicts the delay for *harbour* (around 60 Mbps). In Fig. 13, the average CPU usage for each measurement run is shown.

The graphs show that each adaptation MANE architecture has a similarly formed cumulative delay distribution function, but with markedly different mean and median values and standard deviations. By comparing *packet forward*ing with the *SVC specific* adaptation solution, it becomes evident that the additional processing effort due to the RTP de-packeting, AU aggregation, adaptation, AU fragmentation, and RTP packeting steps is very low, because the adaptation engine used for *SVC specific* adaptation is very simple.

The delays of the description driven adaptation approaches, *gBSD 1 stream* and *gBSD 2 stream*, are notably higher, mainly because of the processing required for decompressing and parsing the XML text (gBSD metadata). In addition, the required synchronization of the metadata and the video data increases this delay. The delay jitter increases significantly as well, as indicated by the standard deviation results.

The gBSD two-stream solution (*gBSD 2 stream*) is inferior to the single-stream solution (*gBSD 1 stream*) because it uses an addi-



Delay [µs]	Median	Mean	Std. Dev.
Packet forward	541	706	597
SVC specific	1125	1351	937
gBSD 1 stream	2672	5050	6836
gBSD 2 stream	3935	6893	9083

Fig. 10. Delay distribution function for sequence *foreman mgs GOP32*.

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

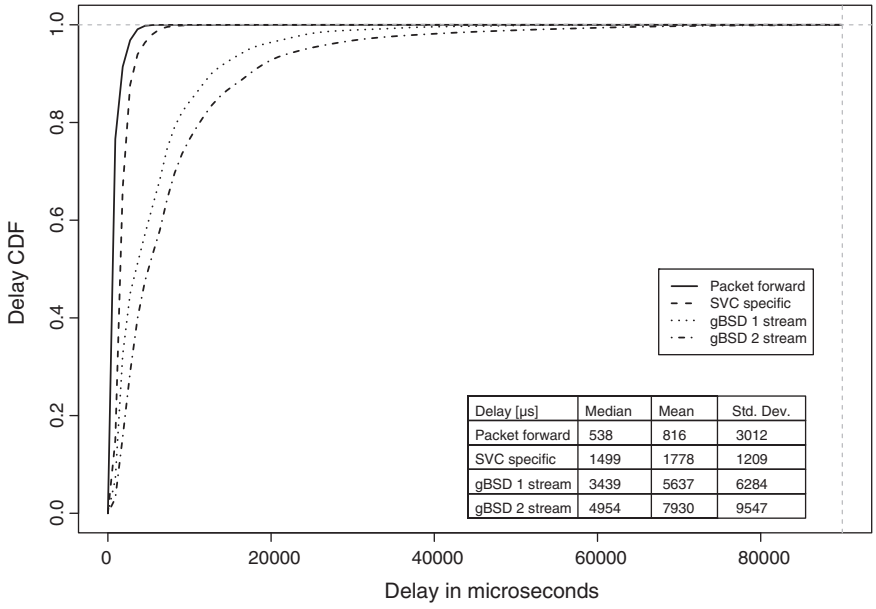


Fig. 11. Delay distribution function for sequence *city mgs2 GOP32*.

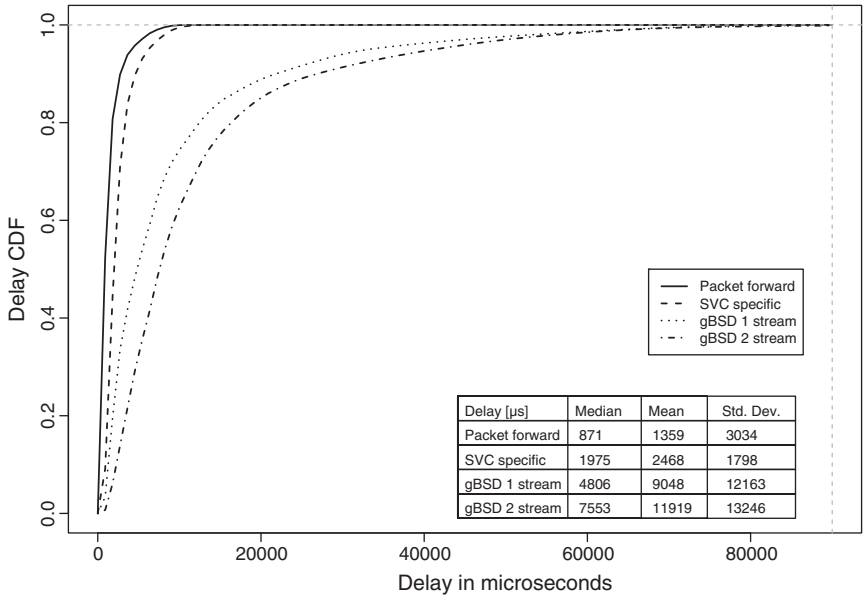


Fig. 12. Delay distribution function for sequence *harbour mgs2 GOP32*.

tional RTP channel for metadata transport. The data from this channel has to be retrieved from a network socket and synchronized with the separate video RTP channel.

Yet, the increased delays and the additional jitter of the gBSD solutions can be regarded as tolerable since more than 90% of the frames are delayed by less than one frame time of the video (1/30 s) and more than 99% of the frames by less than two frame times.

In direct comparison, Figs. 10–12 reveal that the delay distribution is also directly dependent on the bit rate of the video: higher bit rates lead to increases both of delay and jitter.

Furthermore, the CPU consumption (Fig. 13) increases with the bit rate, yet rather modestly. This increase is mainly due to the higher media data throughput; gBSD metadata size and thus metadata processing effort is almost constant across the different test streams and bit rates (at the same degree of adaptation facilities that the streams offer). CPU usage results are consistent with the delay distribution results: the lowest computational effort is needed for *packet forward*ing followed by the simple *SVC specific* adaptation, *gBSD 1 stream*, and *gBSD 2 stream*. The gBSD based adaptation solutions are notably outperformed by the simple SVC specific adaptation implementation by factors of approx. three to five. (These factors are derived by comparing the CPU usage values for the two approaches and factoring out the CPU load that both have in common, i.e., the reference load for *packet forward*ing.) While the gBSD solutions induce more computational

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

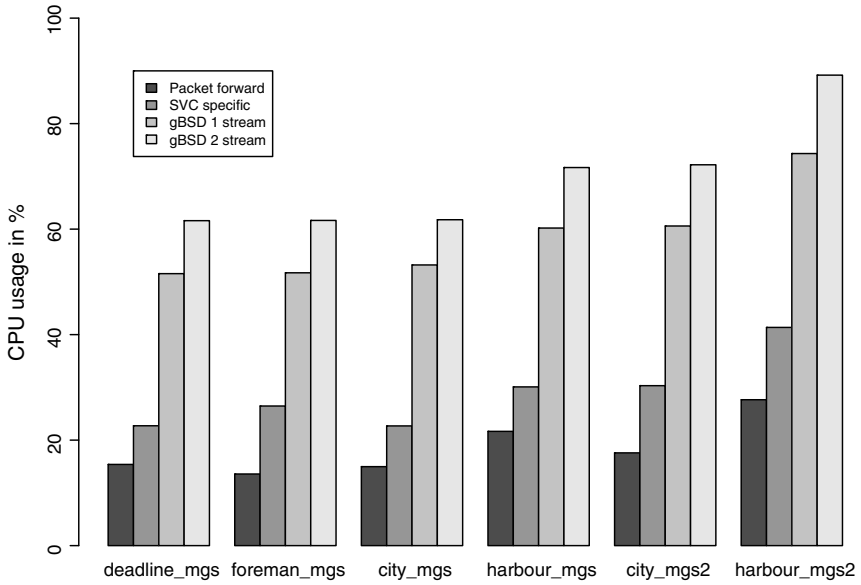


Fig. 13. Average CPU usage.

load, still a significant number of parallel streams (30, in the figures shown, representing up to 60 Mbps of throughput) can be handled in real time on a standard PC-based adaptation MANE.

9. Conclusions

In this paper, we evaluated mechanisms for adapting SVC video streams on a mid-network adaptation MANE and compared both codec aware and MPEG-21 gBSD metadata driven adaptation approaches.

The gBSD description driven adaptation technique has major conceptual advantages. It represents a codec agnostic way of (scalable) media stream adaptation, is thus easily extensible to new media coding formats, is flexible in the sense that various levels of description and adaptation granularity can be covered, and is powerful in that format independent modifications can be performed, e.g., semantic adaptations based on the violence levels of scenes. These properties come along as a result of the generic adaptation machinery that transforms the XML-based bitstream syntax descriptions and generates the adapted bitstreams associated with the modified descriptions. Only the specific gBSD descriptions, adaptation decisions, and the parameters steering the adaptation process are application dependent.

Based on full implementations of the adaptation approaches under consideration and on our evaluations, we can conclude that the gBSD based adaptation approach is a viable alternative to SVC specific adaptation, provided that a competitive implementation of the gBSD based approach is available. Our gBSD implementation is advanced in the sense that it does not rely on standard XML transformation techniques (like XSLT), but realizes a more efficient way to modify an XML based bitstream description and to generate the adapted bitstream (by directly manipulating the C++ object tree, rather than relying on XSLT processing on the DOM tree). Yet, the approach is compliant to the gBSDtoBin process as specified in the MPEG-21 DIA standard.

According to our experimental results, gBSD description driven adaptation is inferior to SVC specific adaptation by factors of approx. three to five, in terms of delay induced on the end-to-end media transmission path and in terms of computational load generated on the adaptation MANE. Yet, gBSD based adaptation can well be performed in real time for up to 60 Mbps of throughput (30 parallel streams of approx. 2 Mbps bit rate each) on a standard desktop machine.

Furthermore, the gBSD based adaptation and the packet handling at the MANE introduce additional jitter on the packetized video stream. However, the amount of jitter can be regarded as manageable since more than 90% of the frames are delayed by less than one frame time of the video (1/30 s) and more than 99% of the frames by less than two frame times in the above situation.

It was found that the metadata overhead can be heavily reduced when describing the video stream on a per-GOP basis. This aggregation and the resulting redundancy within the metadata can then be utilized for a more efficient compression. The resulting bit rate of the compressed metadata stream is around 1% of the bit rate of the video stream. For the actual adaptation, it turned out that the adaptation on a per-AU basis (combined with the description on a per-GOP basis, sent in advance of the GOP's media data) is advantageous since, qualitatively, it provides the full adaptation flexibility inherent in the SVC bitstream and, quantitatively, it both reduces the delay introduced by the adaptation MANE and increases the error resilience.

Finally, it turned out that transporting the gBSD description within the SVC media stream in the form of customized SEI messages (gBSD single-stream solution), is superior in performance to transmitting the gBSD metadata on a second RTP channel (gBSD two-stream solution). Managing and serving an additional RTP channel is expensive. In the specific case of this paper, metadata can be encapsulated in the media data (SVC) stream, thus the separate RTP channel can be avoided. For a fully generic adaptation MANE or in case gBSD metadata cannot be embedded into the media data stream, the gBSD two-stream solution must be adopted, employing the generic RTP payload format.

The results of this paper, i.e., the conceptual benefits and the performance drawbacks of the gBSD description driven adaptation mechanism as compared to the SVC specific technique, will support the decision which adaptation approach to use in a specific application.

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004

References

[1] Apple Inc. Darwin—Open Source Streaming Server, http://developer.apple.com/opensource/server/streaming/.
[2] I. Burnett, R. Koenen, F. Pereira, R. Van de Walle (Eds.), The MPEG-21 Book, Wiley, 2006.
[3] Code Synthesis Tools CC, CodeSynthesis XSD: XML Data Binding for C++. http://www.codesynthesis.com/products/xsd/.
[4] D. De Schrijver, W. De Neve, K. De Wolf, R. De Sutter, R. Van de Walle, An optimized MPEG-21 BSDL framework for the adaptation of scalable bitstreams, Journal of Visual Communication and Image Representation 18 (3) (2007) 217–239.
[5] D. De Schrijver, C. Poppe, S. Lerouge, W. De Neve, R. Van de Walle, MPEG-21 bitstream syntax descriptions for scalable video codecs, Multimedia Systems 11 (5) (2006) 403–421. June.
[6] The Apache Software Foundation, Xerces C++ Parser. http://xerces.apache.org/xerces-c/.
[7] M. Handley, V. Jacobson, C. Perkins, SDP: Session Description Protocol, RFC 4566 (2006). July.
[8] ISO/IEC 15938-1:2002, Information Technology—Multimedia content description interface—Part 1: Systems. July 2002.
[9] M. Karczewicz, R. Kurceren, The SP- and SI-frames design for H.264/AVC, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 637–644. July.
[10] I. Kofler, M. Prangl, R. Kuschnig, H. Hellwagner, An H.264/SVC-based adaptation proxy on a WiFi router, in: Proceedings of the 18th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), Braunschweig, Germany, May 2008, pp. 63–68.
[11] I. Kofler, C. Timmerer, H. Hellwagner, A. Hutter, F. Sanahuja, 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), San Jose, CA, USA, Jan./Feb. 2007.
[12] J. Le Feuvre, C. Concolato, J.-C. Moissinac, GPAC: open source multimedia framework, in: Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 2007, pp. 1009–1012.
[13] Live Networks Inc. LIVE555 Streaming Media, http://www.live555.com/liveMedia/.
[14] S. McCanne, V. Jacobson, M. Vetterli, Receiver-driven layered multicast, in: Proceedings of the 1996 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), Palo Alto, CA, USA, Aug. 1996, pp. 117–130.
[15] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Joint Scalable Video Model. Doc. JVT-X202, July 2007.
[16] F. Pereira, J. Smith, A. Vetro, Special section on MPEG-21, IEEE Transactions on Multimedia 7 (3) (2005). June.
[17] M. Ransburg, R. Cazoulat, B. Pellan, C. Concolato, S. De Zutter, R. Van de Walle, Dynamic and distributed adaptation of scalable multimedia content in a context-aware environment, in: Proceedings of the 1st European Symposium on Mobile Media Delivery (EuMob), Alghero, Italy, September 2006.
[18] M. Ransburg, S. Devillers, C. Timmerer, H. Hellwagner, Processing and delivery of multimedia metadata for multimedia content streaming, in: Proceedings of the 6th Workshop on Multimedia Semantics—The Role of Metadata, Aachen, Germany, March 2007, pp. 117–138.
[19] M. Ransburg, C. Timmerer, H. Hellwagner, Transport mechanisms for metadata-driven distributed multimedia adaptation, in: Proceedings of the 1st International Conference on Multimedia Access Networks (MSAN), Orlando, Florida, USA, June 2005, pp. 25–29.
[20] M. Ransburg, C. Timmerer, H. Hellwagner, S. Devillers, Design and evaluation of a metadata-driven adaptation node, in: Proceedings of the 8th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Santorini, Greece, June 2007, pp. 83–86.
[21] T. Schierl, S. Wenger, Signaling media decoding dependency in Session Description Protocol (SDP), Internet Draft draft-ietf-mmusic-decoding-dependency-02 (2008).
[22] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, RFC 3550 (2003).
[23] H. Schulzrinne, A. Rao, R. Lanphier, Real Time Streaming Protocol (RTSP), RFC 2326 (1998).
[24] H. Schwarz, D. Marpe, T. Wiegand, Analysis of hierarchical B pictures and MCTF, in: Proceedings of the 2006 IEEE International Conference on Multimedia and Expo (ICME), Toronto, Canada, July 2006.
[25] J. Seward, bzip2 Homepage, http://www.bzip2.org.
[26] The GNOME Project, The XML C parser and toolkit of Gnome. http://xmlsoft.org.
[27] C. Timmerer, S. Devillers, M. Ransburg (Eds)., ISO/IEC 21000-7:2007 Part 7: Digital Item Adaptation, second ed., International Standardization Organization, 2007.
[28] Timmerer, C., Kofler, I., Liegl, J., Hellwagner, H, An evaluation of existing metadata compression and encoding technologies for MPEG-21 applications, in: Proceedings of the 1st IEEE International Workshop on Multimedia Information Processing and Retrieval (MIPR), Irvine, California, USA, December 2005, pp. 534–539.
[29] A. Vetro, C. Christopoulos, T. Ebrahimi, Special issue on universal multimedia access, IEEE Signal Processing Magazine 20 (2) (2003). March.
[30] A. Vetro, C. Timmerer, Digital Item Adaptation: overview of standardization and research activities, IEEE Transactions on Multimedia 7 (3) (2005) 418–426. June.
[31] Y. Wang, M.M. Hannuksela, S. Pateux, A. Eleftheriadis, S. Wenger, System and transport interface of SVC, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1149–1163. September.
[32] S. Wenger, Error Patterns for Internet Experiments. ITU—Telecommunications Standardization Sector, Study Group 16, Video Coding Experts Group (Question 15), Document Q15-I-16r1, October 1999. http://ftp3.itu.ch/av-arch/video-site/9910_Red/q15i16r1.zip.
[33] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, D. Singer, RTP payload format for H.264 video, RFC 3984 (2005). February.
[34] S. Wenger, Y. Wang, T. Schierl, Transport and signaling of SVC in IP networks, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1164–1173. September.
[35] S. Wenger, Y. Wang, T. Schierl, A. Eleftheriadis, RTP payload format for SVC video, Internet Draft draft-ietf-avt-rtp-svc-09 (2008). May.
[36] T. Wiegand, J. Ohm, G. Sullivan, A. Luthra, Special issue on scalable video coding—standardization and beyond, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007). September.
[37] T. Wiegand, G. Sullivan, H. Schwarz, M. Wien (Eds.), ISO/IEC 14496-10:2005/Amd3: Scalable Video Coding, International Standardization Organization, 2007.
[38] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, P. Amon, Real-time system for adaptive video streaming based on SVC, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1227–1237. September.
[39] XSL Transformations (XSLT) 1.0. W3C Recommendation, 16 November 1999. URL: http://www.w3.org/TR/1999/REC-xslt-19991116.

Please cite this article in press as: R. Kuschnig et al., Design options and comparison of in-network H.264/SVC adaptation, J. Vis. Commun. (2008), doi:10.1016/j.jvcir.2008.07.004