

An Interoperable Streaming Framework for Scalable Video Coding Based on MPEG-21

Michael Eberhard*, Luca Celetto[†], Christian Timmerer*, Emanuele Quacchio[†],
Hermann Hellwagner*, and Fabrizio S. Rovati[†]

*ITEC, MMC – Klagenfurt University, Austria, <firstname.lastname>@itec.uni-klu.ac.at

[†]ST Microelectronics, Italy, <firstname.lastname>@st.com

Keywords: Scalable Video Coding, Adaptation, Video on Demand, Multicasting, MPEG-21 Digital Item Adaptation.

Abstract

This paper presents an interoperable framework for the streaming of scalable multimedia content such as Scalable Video Coding (SVC). In particular, the framework's architecture for both, Video on Demand (VoD) and multicast streaming, is presented. The architecture includes a detailed description of the adaptation engine – conforming to MPEG-21 Digital Item Adaptation – as well as the integration of the adaptation engine into VideoLAN's VLC media player, which provides the streaming server and client for the framework. Following the description of the architecture, a comparison in terms of performance of the generic MPEG-21 DIA-based adaptation approach, which is utilized by the described demo, versus an SVC-specific adaptation approach is presented and possible further improvements for both approaches are investigated.

1 Introduction

The streaming of multimedia content over heterogeneous wired and wireless networks is due to the heterogeneity of the capabilities of the users' terminals in terms of supported codecs, display resolution, processing power, energy supply, or bandwidth conditions a challenging research area. Technologies like transcoding or stream switching offer in general not the performance and flexibility which are needed to tailor the multimedia resources to the requirements of the users and their terminals [1]. Thus, scalable multimedia coding formats which allow the adaptation to a lower quality by simple removing or minor editing operations provide a good possibility to satisfy the envisaged scenarios. The Scalable Video Coding (SVC) extension of the ISO/ITU-T's Advanced Video Coding (AVC) standard [2] provides support for temporal, spatial and signal-to-noise ratio (SNR) scalability while still maintaining the superior coding efficiency of AVC.

Although scalable multimedia coding formats can be adapted by applying simple removing or minor editing operations, the adaptation process is specifically defined for each of the various scalable audio and video coding formats. To address this dependence of the adaptation process on the coding

formats, interoperable and universal adaptation engines which perform an adaptation of a scalable multimedia content independent of the coding format can be utilized. A solution for such an interoperable adaptation engine is enabled by metadata specified in Part 7 of the MPEG-21 standard, Digital Item Adaptation (DIA) [3].

In this paper, a framework including a test-bed for the interoperable streaming and adaptation of scalable multimedia contents is presented providing the possibility for both, Video on Demand (VoD) and multicasting deployed for Internet Protocol Television (IPTV) applications. A detailed description of the architecture of the VoD implementation is given in Section 2 while the architecture of the multicast implementation is presented in Section 3. Furthermore, the performance comparison of the generic adaptation approach provided by MPEG-21 DIA to an SVC-specific adaptation approach provided by the JSVM reference software [6], which has already been briefly presented in [7], is further discussed in Section 4. This comparison includes possible optimizations for both approaches. For the MPEG-21 DIA metadata-based adaptation approach, the processing of the metadata has been identified as the performance bottleneck. Thus, alternatives to the traditionally used processing libraries are presented. Furthermore, possible optimizations for the reference software by utilizing length information in Network Abstraction Layer (NAL) unit header are investigated. Section 5 concludes this paper.

2 An Interoperable Architecture for Video on Demand

The architecture of our test-bed illustrated in Figure 1 demonstrates the adaptation of scalable multimedia streams in heterogeneous networks for the VoD scenario, utilizing an MPEG-21 DIA metadata-based adaptation engine. The test-bed consists of the *MPEG-21 DIA-enabled VoD Server* and a number of heterogeneous *VoD Clients*, e.g., a notebook computer, a Personal Digital Assistant (PDA), or a Television-Set with a Set-Top-Box (STB) for SVC base layer decoding.

The implementation of the test-bed comprises two major parts: the VLC media player and streaming server¹, which provides the streaming capabilities for the test-bed, and the

¹ <http://www.videolan.org/vlc/>

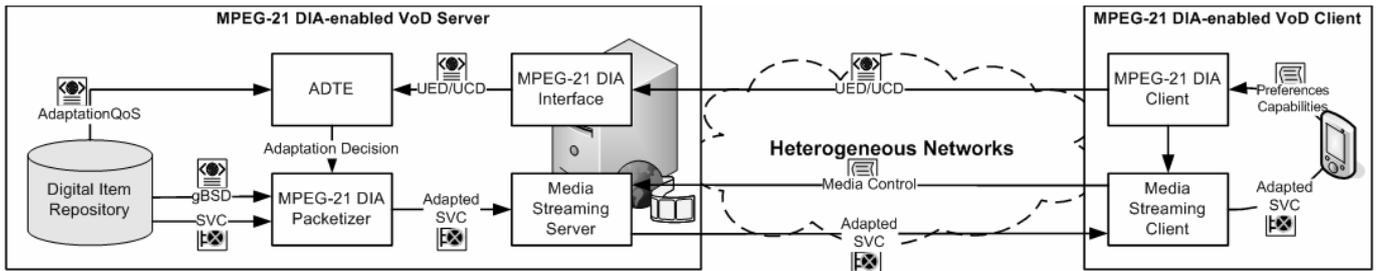


Figure 1: An Interoperable Architecture for Video on Demand.

MPEG-21 DIA tools, which provide the capabilities for the adaptation decision-taking process based on the users' preferences and the capabilities of their terminals, and for the adaptation of the scalable multimedia streams based on this decision. The walkthrough for a typical VoD streaming session with one streaming client is described in the sequel.

To setup the VoD session, the client sends an initial content request to the *MPEG-21 DIA Interface* of the server and receives a list of the available Digital Items. The terminal displays the available sequences and allows the user to choose a sequence and to set her/his viewing preferences for the three SVC scalability dimensions. Subsequently, the client transmits the user's preferences and the terminal's capabilities formatted as MPEG-21 DIA metadata, i.e., as Usage Environment Description (UED) and as Universal Constraints Description (UCD) [3], together with the chosen Digital Item identifier to the server utilizing the Hypertext Transfer Protocol (HTTP) to trigger the setup of the VoD session. The server's interface returns a Real Time Streaming Protocol (RTSP) Uniform Resource Locator (URL) to the client after the receipt of the metadata, which is subsequently utilized by the client to start the streaming of the multimedia content.

The implementation of the streaming client was performed with the intention to make it available on as many platforms as possible. Therefore, the client consists of the *Media Streaming Client*, which is provided by the VLC, as well as of an extension of the VLC, the *MPEG-21 DIA Client*. The client supports HTTP for the transmission of the MPEG-21 DIA metadata as well as RTSP for starting and controlling of the streaming session. By adding the MPEG-21 DIA functionality on top of these protocols, the existing VLC media streaming solution was extended by supporting MPEG-21 DIA without introducing proprietary protocols.

On the server side, the *MPEG-21 DIA Interface*, as well as the *Adaptation Decision-Taking Engine (ADTE)* and the *MPEG-21 DIA Packetizer* are integrated into the VLC as dynamic modules. After the receipt of the UED/UCD, the interface passes the metadata to the ADTE. Furthermore, the ADTE accesses the AdaptationQoS description, which describes the adaptation capabilities of the SVC content, from the *Digital Item Repository*. The ADTE matches the usage environment properties and the preferences described by the UED/UCD with the available SVC layers described by the AdaptationQoS description to find the optimal adaptation parameters. This matching process can be seen as mathematical optimization process and is already discussed in the literature [8]. Furthermore, the MPEG-21 DIA interface uses the Digital Item Identifier to set up the VoD session for the *Media Streaming Server*.

The adaptation parameters, which are the result of the adaptation decision-taking process, are forwarded to the *MPEG-21 DIA Packetizer*, which performs the actual adaptation process on the SVC content. The packetizer is called by the VLC main program control for each access unit during the streaming process. The packetizer performs the adaptation of the access units on a NAL unit level utilizing the adaptation parameters provided by the ADTE as well as the generic Bitstream Syntax Description (gBSD) [3]. The gBSD provides a high-level description of the structure of the bitstream. The adaptation process comprises the transformation of the gBSD based on the adaptation parameters and the actual adaptation of the SVC bitstream. The former can be performed by any XML processing/transformation tool. The latter is specified by MPEG-21 DIA and is referred to as gBSDtoBin [3] which utilizes the transformed gBSD to extract the required bitstream segments. The adapted access units are finally streamed to the client by the VLC's *Media Streaming Server*. Additionally, the test-bed allows each client to dynamically update its usage environment properties and the user's preferences during the streaming session. In order to perform such a dynamic update, the user needs to input the changed preferences/capabilities to the client and the client transmits the data wrapped in the UED/UCD to the server. At the server, the adaptation decision is taken again by the ADTE and the updated adaptation parameters are provided to the *MPEG-21 DIA Packetizer*. As the packetizer is called by the VLC for each access unit, the change of the adaptation parameters becomes effective for the adaptation of the following access unit. Thus, the update of the adaptation parameters is visible at the client as soon as the next access unit is transmitted.

3 An Interoperable Architecture for Multicast Streaming

The layered-multicast implementation intends to provide the layers of a scalable multimedia bitstream utilizing several parallel multicast Real-Time Transport Protocol (RTP) sessions. Usually channels conditions are both time- and user-dependent, and users have different network and computational resource requirements. By using the layered multicast within the streaming platform, it is possible to reduce these problems, enabling different devices to join one or more RTP sessions according to their capabilities, available bandwidth and computational power. Furthermore, it would be possible to dynamically subscribe/unsubscribe to RTP sessions in case of congestion or changes of the network

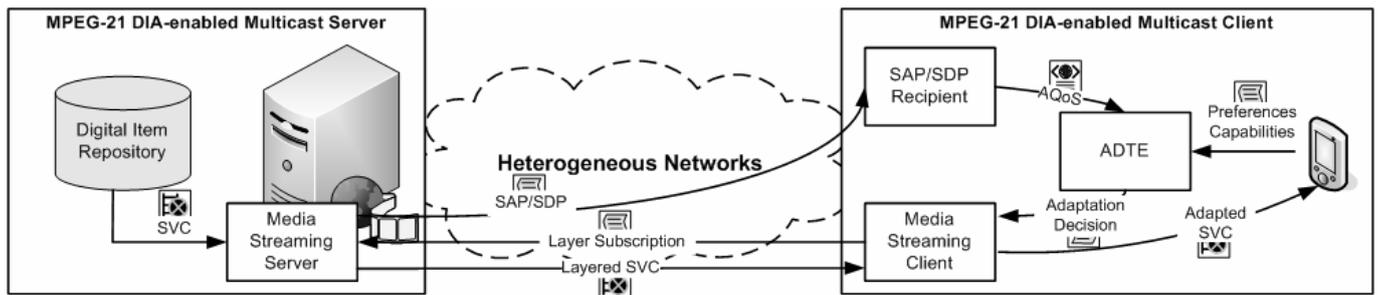


Figure 3: Interoperable Architecture for Multicast Streaming.

conditions, in order to reduce the network load and experiencing a graceful degradation of the video quality.

The layered multicast implementation for scalable bitstreams is based on packets formatted by the hint tracks [4] of the MPEG-4 file format. The hint tracks are created by parsing the bitstream offline and including the hinting information in the MP4-file. Thus, the server only needs to parse the hint tracks and not the bitstream during the real-time streaming, which instruct the server how to copy the NAL units from the scalable bitstream to the output packets of the different multicast sessions.

The multicast session information and parameters are broadcasted to the connecting clients in the SDP (Session Description Protocol)² format utilizing the Session Announcement Protocol (SAP)³. An example for such an SDP announcement is given in Figure 2.

```

1. sdp=v=0
2. o=- 1206634430919576 3 IN IP4 127.0.0.1
3. s=NONE
4. t=0 0
5. a=tool:vlc 0.8.4a
6. c=IN IP4 224.0.0.1/1
7. a=group:DDP 1 2
8. m=video 33032 RTP/AVP 96
9. a=control:trackID=3
10. a=rtpmap:96 H264/90000
11. a=mpeg4-esid:1
12. a=fmtp:96 profile-level-id=4d400b;
13.   packetization-mode=2; init-buf-time=0;
14.   sprop-parameter-sets=Z01AC5rLFicg,aN4Eag==;
15.   aEngRqA=,aGngRiA=;
16.   framewidth=176; frameheight=144;
17.   framerate=30; bitrate=400;
18. a=mid:1
19. m=video 33034 RTP/AVP 97
20. a=control:trackID=4
21. a=rtpmap:97 SVC/90000
22. a=mpeg4-esid:1
23. a=fmtp:97 profile-level-id=53000d;
24.   packetization-mode=2; init-buf-time=0;
25.   sprop-parameter-sets=
26.   Z01AC5rLFicg,Z1MADUsBBrLCwSyA,aN4Eag==,
27.   aEngRqA=,aGngRiA=;
28.   framewidth=352; frameheight=288;
29.   framerate=30; bitrate=1200;
30. a=mid:2
31. a=depend:lay 1
32. m=audio 33036 RTP/AVP 98
33. a=control:trackID=5
34. a=rtpmap:98 MPA/48000
35. a=mpeg4-esid:2

```

Figure 2: SAP/SDP Announcement.

The SDP description includes general information about the multicast session in lines 1 to 7 like the protocol version, the owner of the session or connection information. Furthermore, the description includes information about each layer, starting with the media description tag (“m”-tag) in lines 8-18 and in lines 19-35, which specifies the port where the multicast layer is transmitted (33032 and 33034). For the first layer, which describes the AVC-compatible base layer, the payload is set to dynamic by selecting a value from the dynamic range, i.e. 96, and the codec is set to AVC. The codec-specific parameters are given by the *fmtp*-attribute of the “a”-tag in lines 12-16. To describe the decoding of the layers, the *fmtp*-attribute contains the *packetization-mode* and *sprop-parameter-sets* fields that allow a proper configuration of the decoder. The *packetization-mode* field provides the possibility to exchange the order of transmission of the NAL units in respect of the decoding order. For the layered multicast, it allows reconstructing the correct re-ordering for decoding the NAL units that are sent over different scalable RTP sessions. The *sprop-parameter-sets* field contains the Sequence Parameter Sets (SPS) and Picture Parameter Sets (PPS) of the scalable bitstream encoded in base64⁴. Furthermore the SVC-specific properties which enable the *SAP/SDP Recipient* at the client to generate the *AdaptationQoS* description are included in the *fmtp*-attribute, i.e., the frame width, frame height, frame rate and bit rate. In addition to using the *fmtp*-attribute, the dependency between the SVC layers is signalled in SDP following a draft specification [5] by means of the *mid* and *lay* attributes. While the *mid*-attribute assigns the ID to the layer, the *depend-lay*-attribute contains all the layers the current layer depends on.

The architecture of the multicast test-bed is illustrated in Figure 3 and consists of the *MPEG-21 DIA-enabled Multicast Server* and a number of *MPEG-21 DIA-enabled Multicast Clients*. The different layers of the scalable content are provided to the clients utilizing a layered multicast approach [9]. In particular, the adaptation decision, which decides which layers need to be subscribed to, is taken at the client.

The streaming server continuously performs a layered multicast of all the SVC layers where each layer is transmitted in a separate RTP session. The properties of the layers are announcement as described above utilizing SAP/SDP announcements.

The client receives the announcements which are saved in the playlist of available channels. Furthermore, the client uses the provided information about the properties of the layers to

² <http://www.ietf.org/rfc/rfc2327.txt>

³ <http://www.ietf.org/rfc/rfc2974.txt>

⁴ <http://www.ietf.org/rfc/rfc3548.txt>

generate the AdaptationQoS description, which contains the adaptation capabilities for all layers. The user's preferences are again provided to the client by the user and are wrapped together with the terminal's capabilities into the UED/UCD. The AdaptationQoS description and the UED/UCD are subsequently passed to the ADTE. The ADTE takes the adaptation decision – as described in Section 2 – based on these metadata and forwards the adaptation decision to the *Media Streaming Client*. The *Media Streaming Client* maps the adaptation decision to the layer properties and subscribes to the desired layers and starts receiving the multimedia data.

4 Evaluation

A comparison of the performance of the generic MPEG-21 DIA metadata-based adaptation approach to an SVC-specific adaptation approach, i.e., the Bitstream Extractor of the JSVM reference software [6], has already been presented in [7]. The comparison has shown that the MPEG-21 DIA metadata-based implementation clearly outperforms the reference software when adaptation is desired. However, during the initial comparison, transformation of the gBSD has been identified as the bottleneck for the MPEG-21 DIA metadata-based adaptation. To further optimize the generic adaptation approach, alternatives to the usage of an Extensible Stylesheet Language for Transformations (XSLT) library for the transformation of the gBSD have been evaluated. To present a fair evaluation, the Bitstream Extractor has been optimized as well: once by adding length information to the headers of the NAL units, and once by implementing a customized, lightweight Bitstream Extractor providing the minimal functionality to extract certain SVC layers. The results of these evaluations are presented in the following.

The MPEG-21 DIA metadata-based adaptation approach consists of three major parts: the *main method*, which performs the preparations of the adaptation process and cleans up after the adaptation is finished, the *transformation* of the gBSD, which transforms the gBSD according to the adaptation parameters, and the *gBSDtoBin* process, which performs the actual adaptation of the content based on the transformed gBSD. The advantage in terms of performance for the MPEG-21 DIA metadata-based adaptation in comparison to the Bitstream Extractor of the JSVM reference software is mainly due to the gBSDtoBin process [7]. That is, while the Bitstream Extractor always has to parse and analyse the whole bitstream, as there is no length information available in the header of the NAL units, the gBSDtoBin process only has to copy those parts of the bitstream which are described by the transformed gBSD. Thus, the performance of the gBSDtoBin process greatly improves if a large number of access units are truncated, while the performance of the Bitstream Extractor remains constant independent of the adaptation parameters. However, the gBSD needs to be transformed prior to the actual adaptation utilizing the gBSDtoBin process. As this transformation introduces an additional overhead which remains rather stable for all layers, the transformation of the gBSD offers the most significant optimization potential.

	Layer 4	Layer 3	Layer 2	Layer 1	Layer 0
Resolution	4CIF	CIF	CIF	QCIF	QCIF
Framerate	15	30	15	30	15
Bitrate(kbps)	3264	1571	1228	376	281
Size (MB)	49	21.8	17	5.2	3.9

Table 1: Layer Properties of the Mariposa Sequence.

	Layer 3	Layer 2	Layer 1	Layer 0
Resolution	4CIF	4CIF	CIF	CIF
Framerate	30	15	30	15
Bitrate(kbps)	4512	3426	1005	773
Size (MB)	4.3	3.3	0.9	0.7

Table 2: Layer Properties of the Ice Sequence.

The main advantage of the usage of an XSLT library is that the codec-specific part of the adaptation process is covered by the XSLT style sheet. Thus, if the transformation needs to be tailored for a different coding format, only the XSLT style sheet needs to be changed, but there is no need to modify or recompile the adaptation engine itself. However, to improve the performance of the transformation process, also other alternatives, which are not as generically applicable as XSLT, were investigated.

To perform a significant comparison of the transformation alternatives, the transformation of a number of different gBSDs was measured in terms of execution time. The performance measurements were performed on a Dell Optiplex GX620 Desktop with an Intel Pentium D 2.8 GHz processor and 1024 MB RAM. The operating system was OpenSUSE 10.2 and the performance measurements were done with the OProfile System Profiler⁵ for Linux in version 0.93.

The performance evaluations are presented for two sequences, Mariposa and Ice; their properties are given in Table 1 and Table 2, where the layer with the highest number represents the SVC sequence in best quality. The layers with a lower number provide a lower quality and are extracted by applying adaptation. Note that each layer contains all lower layers, i.e., layer n includes layers m < n. In addition to the two presented sequences, other bitstreams (Foreman, Harbour) have been evaluated as well and have shown very similar results.

For the comparison of the gBSD transformations, four different transformation approaches were investigated: (1+2) two XSLT libraries, (3) one approach which significantly improves the performance but is not as generically applicable as an XSLT library, and (4) one approach which improves the performance and still remains generically applicable. As XSLT libraries, libxslt⁶ and xalan-c⁷ were used. The third option is a libxml⁸-based transformation, which performs the transformation of the gBSD by traversing the XML tree of the gBSD and deleting/updating the nodes which need to be adapted. Although this approach provides a very good performance, it is not coding format independent, as the structure of the XML document needs to be considered in the program's structure, i.e., the program control has to know which nodes need to be removed utilizing the transformation

⁵ <http://oprofile.sourceforge.net/>

⁶ <http://xmlsoft.org/XSLT/>

⁷ <http://xml.apache.org/xalan-c/>

⁸ <http://xmlsoft.org/>

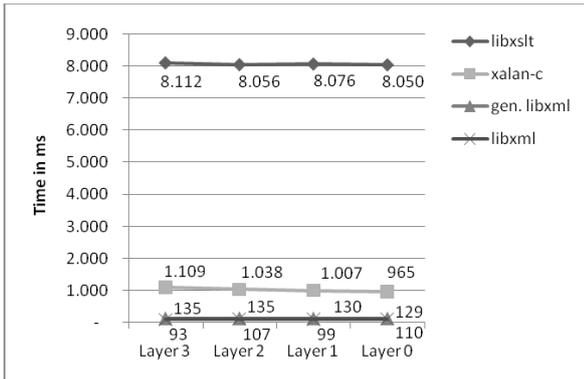


Figure 4: Transformation Results for Mariposa.

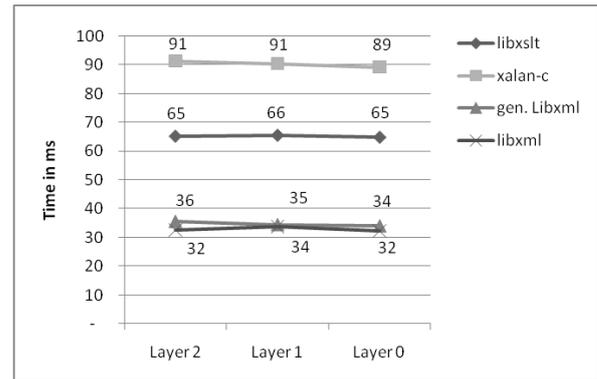


Figure 7: Transformation Results for Ice.

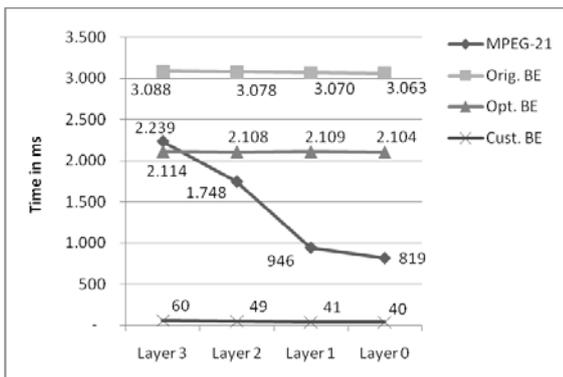


Figure 5: Adaptation Results for Mariposa.

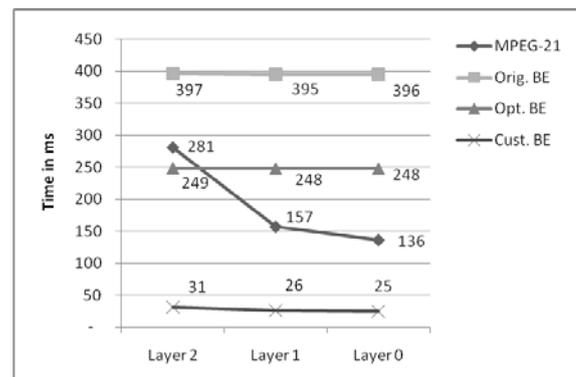


Figure 6: Adaptation Results for Ice.

parameters. Thus, the program needs to be changed and recompiled if another coding format is utilized. The second alternative transformation approach is again a libxml-based transformation. However, for this approach a generic transformation interface is introduced:

- The removal of nodes in the transformation process is done using the *Criteria* interface. A *Criteria* instance identifies the node which needs to be removed by the names of the node, its attribute and the value of the attribute. Furthermore, the *Criteria* specifies the operation for the comparison (e.g., *equals* or *lower/greater than* for numeric values) and possibly an additional descriptor.
- When the libxml-based transformation processor is started, it checks each node of the XML document if the node satisfies the removal criterion. If the check is successful, the node is removed from the XML document.
- The *Criteria* are created as output of the ADTE, maintaining the generic applicability and coding format independence of the MPEG-21 DIA metadata-based adaptation approach.

The results of the comparison of the four transformation approaches for the two bitstreams are displayed in Figure 4 and Figure 7.

The results of the comparison differ significantly for these bitstreams when the XSLT libraries are considered. For the Mariposa sequence, libxslt has by far the worst performance. The reason for this result is that libxslt seems to have problems with large XML files, which makes xalan-c in general the best choice when an XSLT library is used, although libxslt is slightly faster than xalan-c for the Ice sequence.

The basic libxml-based transformation shows the best performance for both sequences. However, the generic libxml-based transformation utilizing the removal-criteria is only approximately one-third slower than the basic implementation for the Mariposa sequence and achieves nearly the same performance for the Ice sequence. As the generic libxml-based transformation approach can be up to 10-times faster than the optimal libxslt library for large bitstream, the generic libxml-based transformation offers an opportunity to significantly improve the overall performance of the MPEG-21 DIA metadata-based adaptation approach.

To perform a fair comparison to the optimized MPEG-21 DIA metadata-based adaptation approach, the Bitstream Extractor of the JSVM reference software has been optimized as well. Firstly, length information has been added to the NAL unit header. Secondly, the optimized Bitstream Extractor utilizes the length information to extract the complete NAL unit at once, without needing to parse the complete NAL unit until the next NAL unit header is found.

However, although these modifications improve the performance of the Bitstream Extractor, the original implementation is not intended to use length information. Thus, the same operations are executed for the NAL units which are removed as well as for the NAL units which are kept (except of the copying to the resulting bitstream). This behaviour is one main reason for the inferior performance of the Bitstream Extractor in comparison to the MPEG-21 DIA metadata-based adaptation approach, which's performance improves significantly if a lower layer is extracted and hence less NAL units are processed and copied to the resulting bitstream.

In order to cope with the limitation indicated above, a customized and optimized version of the Bitstream Extractor has been implemented. This very simple and optimized implementation does not aim to provide the complete functionality of the original Bitstream Extractor, but has been implemented for the sole purpose of adaptation. However, as it has been implemented with the intention to utilize the length information in the NAL unit header, the performance of the customized Bitstream Extractor improves if a lower layer is extracted. The evaluation results of (1) the MPEG-21 DIA metadata-based adaptation approach utilizing the generic libxml-based transformation, (2) the original, (3) the optimized and (4) the customized Bitstream Extractor are illustrated in Figure 5 and Figure 6.

The results show that the optimization of both, the MPEG-21 DIA metadata-based approach and the original Bitstream Extractor have not significantly changed the results of the comparison presented in [7]. As the performance of both approaches has been optimized by approximately one-third, the MPEG-21 DIA metadata-based approach is still significantly faster if lower layers are extracted in comparison to the optimized Bitstream Extractor. This is mainly due to the performance of the gBSDtoBin process, which receives a smaller gBSD if lower layers are extracted and only has to copy a smaller part of the original bitstream.

The performance of the customized Bitstream Extractor is by far the best for all the evaluated sequences. Depending on the layer which needs to be extracted, it is up to 80 times faster than the original Bitstream Extractor for the Mariposa sequence. Additionally, the customized Bitstream Extractor implementation shows that if the length information in the NAL unit header is utilized, the extraction of lower layers is performed clearly faster.

The results show that the generic libxml-based transformation of the gBSDs provides a very good opportunity to improve the performance of the MPEG-21 DIA metadata-based adaptation. Utilizing the generic libxml-based transformation, the MPEG-21 DIA metadata-based approach outperforms even the optimized Bitstream Extractor of the reference software. However, the customized and optimized Bitstream Extractor application shows that there is still a great potential for further improvements of the adaptation process.

5 Conclusion

This paper introduced an interoperable framework including a test-bed for the adaptation and streaming of scalable multimedia content utilizing MPEG-21 DIA metadata for adaptation purposes. The architecture of this test-bed for VoD as well as for multicast was presented. While the VoD architecture allows each client to receive a separate adapted bitstream, the multicast architecture reduces the network traffic by providing a layered multicast to all clients. In case of the VoD scenario, the adapted bitstream is tailored to the user's preferences and the capabilities of the user's terminal by providing the UED/UCD to the server whereas in the case of the layered multicast, the client decides which layers to subscribe to based on these metadata.

Furthermore, a performance evaluation to further improve the performance in terms of execution time of the MPEG-21 DIA metadata-based adaptation engine as well as of the Bitstream Extractor application was performed. The performance bottleneck of the MPEG-21 DIA metadata-based approach, the transformation of the gBSDs, has been significantly improved by utilizing the presented generic libxml-based transformation approach instead of an XSLT library. It has been shown that this approach does not only improve the performance but maintains the generic applicability and coding format independence of the MPEG-21 DIA metadata-based adaptation.

The performance evaluations for the Bitstream Extractor of the JSVM reference software show that the performance can be significantly improved if length information is included in the NAL unit headers. However, to fully exploit the optimization potential of this length information a customized Bitstream Extractor application has been implemented and has shown that there is still a significant potential for improvement for both approaches, the SVC-specific Bitstream Extractor and the MPEG-21 DIA metadata-based adaptation.

Acknowledgements

This work is supported in part by the European Commission in the context of the P2P-Next project (FP7-ICT-216217). Further information is available at <http://www.p2p-next.org/>.

References

- [1] B. Shen, W.-T. Tan, F. Huve, "Dynamic Video Transcoding in Mobile Environments", *IEEE MultiMedia*, vol. 15, no. 1, Jan.-Mar., 2008, pp. 42-51.
- [2] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Trans. on CSVT*, vol. 17, no. 9, September 2007, pp. 1103-1120.
- [3] A. Vetro, "MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access", *IEEE MultiMedia*, vol. 11, no. 1, Jan.-Mar. 2004, pp. 84-87.
- [4] ISO/IEC 14496-12 International Standard – Information technology, Coding of audio-visual objects – Part 12: ISO base media file format.
- [5] T. Schierl, S. Wenger, "Signaling media decoding dependency in Session Description Protocol (SDP)", MMUSIC Working Group Internet Draft, February 2008.
- [6] Joint Scalable Video Model (JSVM) 9, Joint Video Team (JVT) of ISO/IEC and MPEG & ITU-T VCEG, N8751, Marrakech, Morocco, January 2007.
- [7] M. Eberhard, L. Celetto, C. Timmerer, E. Quacchio, and H. Hellwagner, "Performance Analysis of Scalable Video Adaptation: Generic versus Specific Approach", *Proc. WIAMIS 2008*, Klagenfurt, Austria, May 2008.
- [8] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Optimal Adaptation Decision-Taking and Network Quality-of-Service", *IEEE Trans. on Multimedia*, vol. 7, no. 3, Jun. 2005, pp.454-462.
- [9] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," *Proc. ACM SIGCOMM*, Stanford, CA, USA, Aug. 1996.