Scalable Video Coding Impact on Networks

Michael Ransburg[1], Eduardo Martínez Graciá[2], Tiia Sutinen[3], Jordi Ortíz Murillo[2], Michael Sablatschan[1], and Hermann Hellwagner[1]

[1] Multimedia Communication (MMC) Research Group, Institute of Information Technology (ITEC), Klagenfurt University, Klagenfurt, Austria
{mransbur,msablats,hellwagn}@itec.uni-klu.ac.at
[2] Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of Murcia, Murcia, Spain
{edumart,jordi.ortiz}@um.es
[3] VTT Technical Research Centre of Finland, Espoo, Finland
tiia.sutinen@vtt.fi

Abstract. This paper describes the CELTIC project on "Scalable Video Coding Impact on Networks" with the focus of designing a streaming system based on the Scalable Video Coding extension of the H.264/AVC standard. The system is designed to cope with streaming scenarios that can be classified in four use cases: session handover, network congestion, receiver heterogeneity and user driven adaptation. A complete overview of the architecture of the system is given. Two demonstration scenarios are described in detail, which point out the advantages of scalable video coding compared to single layer approaches in multimedia transmission and adaptation scenarios. A concluding section summarizes the work and provides an outlook to future work items.

Keywords: Scalable Video Coding, H.264/SVC, streaming, architecture, video adaptation.

1 Introduction

In this paper we give an overview of the CELTIC project Scalable Video Coding Impact on Networks (SCALNET), which is a 24-month project conducted by 9 industrial and academic partners from Europe. SCALNET focuses on the Scalable Video Coding extension of the AVC standard (H.264/SVC) and its impact on backbone, access and home networks.

The remainder of this paper is organized as follows. In Section 2 we describe the fundamentals of SVC. Section 3 gives an overview of the different use cases which are considered in the project. The architecture of SCALNET, which is based on the requirements resulting from these application scenarios, is presented in Section 4. Subsequently, in Section 4.4 we describe how the SCALNET architecture realizes the envisioned use cases by describing two project demonstrators. Finally, we conclude this paper and provide an outlook to future work items.

J. Rodriguez, R. Tafazolli, C. Verikoukis (Eds.): MOBIMEDIA 2010, LNICST 77, pp. 571–581, 2012.

2 H.264/SVC Fundamentals

This section provides a brief technical introduction to H.264/SVC [1] in which we assume that the reader is familiar with H.264/AVC [2] video coding. In the following we refer to H.264/SVC as SVC and to H.264/AVC as AVC. SVC was standardized in a common effort of the Joint Video Team of the ITU-T VCEG and the ISO/IEC MPEG. The term scalable refers to the possibility to adapt the video bit stream to user preferences, terminal capabilities or network conditions by simply disregarding certain parts, called enhancement layers, from the video bit stream.

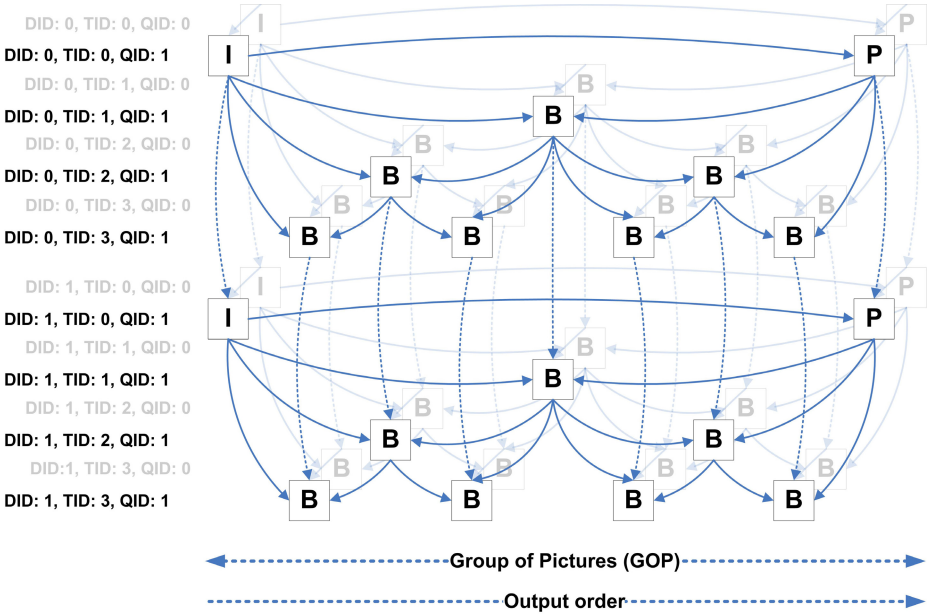

Fig. 1. SVC scalability overview

Figure 1 shows the prediction structure and the different enhancement layers for a Group of Pictures (GoP) which consists of 9 pictures with the structure IBBBBBBBP. Each enhancement layer enhances either the temporal level (frame rate), the spatial level (video dimensions), or the fidelity (signal-to-noise ratio) of each picture of the video bit stream. In addition to intra layer prediction (based on AVC mechanisms), each enhancement layer either predicts from the base layer (if it is the lowest enhancement layer) or from its lower enhancement layer(s). The figure also shows that each picture is divided into two or more units, called Network Abstraction Layer Units (NALUs). The exact number of NALUs per pictures depends, among others, on the number of enhancement layers. In Figure 1 the temporal level of each NALU is indicated by the Temporal ID (TID), the spatial level is indicated by the Dependency ID (DID), and the fidelity is

indicated by the Quality ID (QID). Note that a spatial enhancement layer may be encoded with the same spatial level as a lower layer, in which case only its fidelity would be enhanced. The highest quality enhancement layer (with TID 3, DID 1 and QID 1) can be seen at the bottom of the figure. From this highest quality it is possible to extract lower temporal, spatial and/or fidelity versions by simply filtering NALUs according to the following pseudo code:

```
for each NALU {
    if ( (DID > DID_limit) ||
         (TID > TID_limit) ||
         (DID == DID_limit && QID > QID_limit) ) {
              discard NALU
    }
```

As further described in Section 4, SCALNET implements this filter in the *SVC Filter* component, which filters the SVC stream based on the filter limits set by the *Adaptation Decision Taking Engine* component, before the NALUs are streamed into the network by the *RTSP Streaming Server*.

It must be noted that there are certain restrictions to the simple filtering mechanism introduced above which are mostly due to prediction dependencies. In order to avoid missing reference NALUs, filtering of spatial or temporal enhancement layers can only start at the beginning of a GoP. Filtering of fidelity enhancement layers can, however, be started at every picture. This is enabled by key pictures, which are introduced into the fidelity enhancement layer(s) and which provide a resynchronization point for the decoder in order to limit decoder drift.

In addition to Video Coding Layer (VCL) NALUs, there are also non-VCL NALUs which include meta information, i.e., Parameter Set (PS) NALUs and Supplemental Enhancement Information (SEI) NALUs. SEI NALUs are not required for the decoding of output pictures. They contain supplemental information such as, for example, the bit rate of the current layer which may be used to decide whether to keep or disregard the layer in case of network constraints. PS NALUs, however, contain header information which is needed for decoding, such as the dimensions of the video. These NALUs need to be updated in case that, e.g., spatial enhancement layers are discarded.

The base layer of an SVC stream is backwards compatible to AVC, i.e., it is playable by existing AVC players, which would simply disregard all enhancement layers. However, as the base layer only represents the lowest quality of the H.264/SVC bit-stream, it is often desirable to also provide the higher quality representations to legacy AVC devices. This is achieved by a process named bit-stream rewriting [3]. It allows to efficiently transform an H.264/SVC bit-stream into an H.264/AVC bit-stream without loss by exploiting the similarities of the two codecs, i.e., without requiring complete transcoding of the bit- stream. The JSVM-rewriter, which implements this process, is available as part of the Joint Software Video Model (JSVM) [4]. Within SCALNET we use an run-time improved version of the JSVM-rewriter as detailed in [5] and refer to it as *SVC2AVC Transcoder*.

For additional information on the fundamentals of SVC the reader is referred to [6] and [7].

After this short introduction to SVC, we will introduce the project's use cases in the next section.

3 Use Cases

In order to define the scope of our work, the SCALNET partners described 20 scenarios in which SVC plays an important role. These 20 scenarios were then classified into the following 4 use cases:

- Session Handover
- Congestion of Networks
- Diversity of End Devices
- User Preferences

The *Session Handover* refers to a use case where the current streaming session is handed over. The scenarios include handover between end devices and/or networks. A handover between end devices is performed when a user switches from one end device to another, e.g., from a desktop computer to a mobile. Similarly, a handover between networks is performed, when the session is transferred from one network to another, e.g., from UMTS to WLAN. Both of these use cases can benefit from SVC's characteristic to be easily adapted to a changing usage environment.

The *Congestion of Networks* use case refers to the case where the network capability and/or condition is the limiting factor for the quality of the SVC stream. Network capability refers to static characteristics of the network, e.g., its theoretical maximum available bandwidth, which can serve as a fixed upper limit for the adaptation process. Network condition, on the other hand, refers to current, dynamic characteristics of the network, e.g., the current packet loss rate.

Similarly to the different types and conditions of networks, there are also diverse end devices ranging from a high-end desktop computer to mobile end devices. This *Diversity of End Devices* is another use case, where the end device characteristics are the limiting factor for the quality of the SVC stream. These end device characteristics can be static, e.g., the display resolution of the screen, or dynamic, e.g., the remaining battery power or the current CPU load.

Finally, adaptation based on *User Preferences* forms a fourth use case in the SCALNET project. In this case the user himself is the steering factor for the adaptation process. This use case includes the possibility to pay for switching to an HD version of the current SD content or to switch to a Picture in Picture mode where several video sessions with a reduced spatial resolution are shown together on the screen.

In the next Section, we focus on the SCALNET architecture.

4 Architecture

This section presents a description of the system architecture of the project, whose design is based on the requirements presented in the previous section. In a first approximation, the SCALNET system is presented in Figure 2.

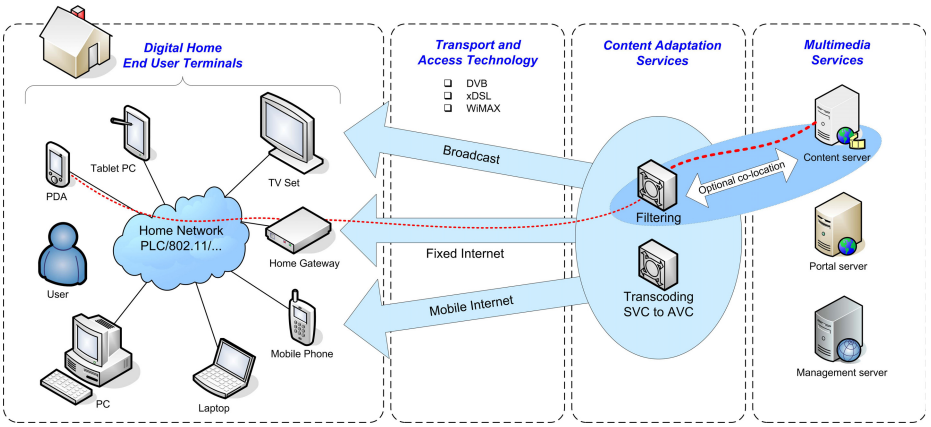

Fig. 2. General architecture

The components identified by this general architecture are described in the following. *Multimedia services* represents services that act as source of multimedia information, or as management mechanisms of user and terminal profiles. Examples for this are IPTV, multimedia broadcasting, Video-on-demand (VOD), mobile video services or User-/Device-/Session Management. *Content adaptation services* implement the procedures of adaptation of the scalable video transmission. Examples for this are filtering SVC and transcoding from SVC to AVC. The *transport and access technologies* include the network technologies used in the backbone and access networks to transfer the scalable video to the user's home. Examples for this are Broadcast (DVB-x), wireless access networks (WLAN, WiMAX), fixed access networks (xDSL, cable, power line). The *home network* represents the set of network technologies that can be used inside the user's home, such as power line network, wireless and wired home networks. Finally, the *end user terminals* represent the heterogeneous set of devices that the user will employ to render the scalable video, e.g., mobile phone, PDA, laptop, personal multimedia player, set-top box or HDTV.

The SCALNET project focuses especially on content adaptation services and how they should be used in different situations in order to achieve better end-user experience. SCALNET also focuses on the impact and adaptations needed in core, access and home networks, either wired or wireless, either fixed or mobile, in order to manage and exploit the SVC technology efficiently. In addition to these topics, SCALNET also studies the control interface between networks and video processing equipments.

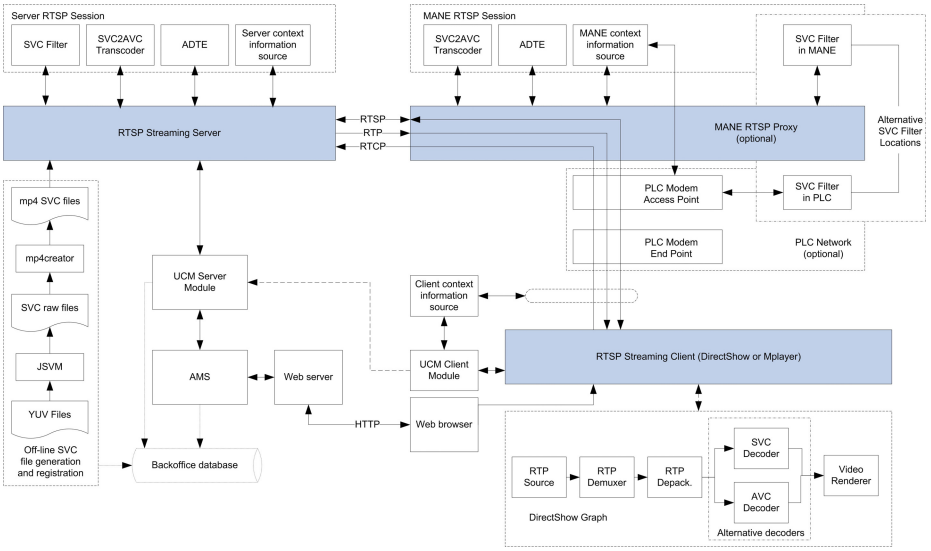

Fig. 3. Functional architecture

The general system architecture is detailed in Figure 3, that shows the actual functional architecture of the project.

It is important to indicate that this figure represents a description of the system that is implemented, that is, it represents concrete choices in the five groups of components of the general architecture described previously. The functional architecture is complete in the sense that it includes the components of all the possible scenarios covered in the project. Some modules are optional, that is, they are implemented but not used in some of the scenarios of the project. For instance, some end terminals use an SVC decoder, whereas others have an AVC decoder.

The following sections contain a short description of the main blocks of the system and the relationships between them. In the Section 4.4 we describe two project demonstration scenarios, which show how the SCALNET architecture realizes the use cases presented in the previous section.

4.1 Management and Portal Servers

The *Administration and Management Server* (AMS) and the *User and Context Management* (UCM) store context information to support the adaptation, and to perform some special functionalities, such as the seamless handover. To employ the system, the user starts a session by logging in a *web portal*. The AMS verifies the user credentials and retrieves the list of video files available. With this information, the portal generates a web page where the user selects a

video. When selecting a video file, the web browser launches the streaming client with a URL that contains streaming session parameters which are used in the *RTSP Streaming Server* to retrieve the static and dynamic context information from the UCM. The context information contains data about the stream quality (network and client performance) and about the stream status (playing, paused, stopped, hold). When the same user logs in on another device, with the same account, the portal can offer to continue a paused stream from the previous device. The same scenario can happen if the network path is interrupted.

4.2 RTSP Streaming Server

The *RTSP streaming server* implements Video on Demand (VoD) and broadcast services. It is a modification of Darwin Streaming Server, an open source streaming server developed by Apple. The modifications permit the server to interact with SVC files [8], that store the scalable video. Content adaptation services are implemented using a set of modules within the server. With the stream session information, retrieved form the UCM, the *Adaptation Decision Taking Engine* (ADTE) can decide to either remove enhancement layers from the SVC stream through the *SVC Filter* or to transcode the SVC stream to AVC through the *SVC2AVC Transcoder*. Quality information from the player is transmitted to the management server by pushing context data to the streaming server through the UCM modules. Additionally, the streaming server can implement a seamless handover between devices, continuing the stream from the last successfully transmitted frame.

4.3 Media Aware Network Element

A *Media Aware Network Element* (MANE) can be located between the *RTSP Streaming Server* and the *RTSP Streaming Client*. The concept of MANE is introduced in [9], as a "network element, such as a middlebox or application layer gateway that is capable of parsing certain aspects of the RTP payload headers or the RTP payload and reacting to the contents". In our project, this MANE is implemented as an RTSP proxy and will preferably reside within the home network, e.g., integrated into a WLAN-AP, or into a Set-Top-Box, as shown in Figure 3. The MANE represents a second possible location of the content adaptation services. In this case, the ADTE receives context information directly from transmission control protocols (RTCP [10]) or from network devices that provide interfaces to retrieve information about the network performance, such as a Power Line Communication (PLC) modem, instead of using the UCM. This approach allows setting the MANE as a plug and play device inside the home network. The SVC Filter, or any module able to adapt an SVC bitstream, can optionally be placed in a network device to implement the adaptation procedure with good performance. An example is a Power Line Connect (PLC) modem.

4.4 RTSP Streaming Client

Finally, the *RTSP Streaming Client* de-packages, decodes and displays the video on the screen. Additionally, the client collects dynamic context information, e.g., the current packet loss rate, and transmits it to the server through the *UCM Client Module*, when needed. The project employs two streaming clients: a modification of *mplayer* able to render SVC video (based on the Open SVC Decoder[1]), and a streaming client developed in the project. The SCALNET streaming client has been implemented using different software components. This methodology enables an easy reuse and modification of applications. In particular, it has been developed using Microsoft DirectShow components, implemented as Microsoft Windows dynamic libraries. The DirectShow client is in charge of organizing the visual interface and the chain of components that enable the reception, decoding and rendering of the video.

This section describes two SCALNET demonstration scenarios in detail.

4.5 Adaptive Session Handover

The Adaptive Session Handover demonstrator, as depicted in Figure 4, demonstrates 1) the automatic adaptation of a session to the current usage environment and 2) the transfer of a session between different end-devices and networks and 3) the support for AVC legacy devices. In practice, this could mean that a user is watching a HD video on demand over his DSL Network and an IP Set top box at home in full HD on his flat screen TV set. When he has to move to the airport, he pauses the video and continues it in the Taxi on his smartphone over the 3G network. As he starts to receive the video it is automatically and constantly adapted to the new usage environment (e.g., available bandwidth, display resolution) in order to guarantee video playback with the best possible quality. If the new usage environment indicates that the end device only supports AVC, the SVC stream is transcoded to AVC on the server in addition to the adaptation which is performed using the SVC Filter.

The demonstrator uses several components, listed below in this clause. The main actors managing a handover scenario are the Administration and Management Server (AMS), the User and Context Management Module (UCMM) and the streaming server. The AMS provides static information about users, contents, devices and network. The Server-Side User and Context Management Module provides dynamic information and an interface between the AMS and the streaming server. The Streaming Server includes the Adaptation Decision Taking Engine, SVC Filter and SVC-to-AVC transcoder. Different user terminals (e.g. PC-based SVC STB, SVC tablet PC, AVC smartphone) are used. Due to the large processing power need of currently available SVC decoders, an SVC tablet PC and a smartphone (with an AVC player) are used as mobile devices. Optionally, the inclusion of the MANE into the demo setup is also possible, e.g., as a powerline communications (PLC) cooperative SVC home gateway.

[1] Open SVC Decoder, http://opensvcdecoder.sf.net/

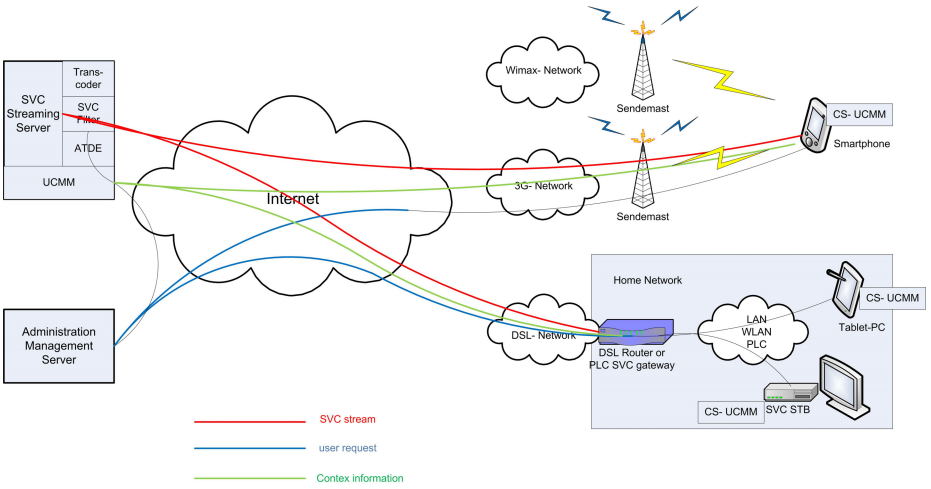

Fig. 4. Handover demonstrator

4.6 Multi-interface Streaming

The Multi-interface Streaming demonstrator, as depicted in in Figure 5, involves the delivery of a scalable video stream using several network interfaces in the terminal device. The main purpose of the demonstrator is to show the benefits gained from using scalable streams in the case where a mobile terminal is equipped with multiple network interfaces (e.g. WLAN, WiMAX, 3G, and DVB-H). The demonstrator can be used to show the flexibility of SVC in cases where 1) no single network connection available to the user is capable of carrying the whole stream but the stream has to be transmitted using multiple network connections (e.g. HSPA and WLAN) simultaneously; and 2) a broadcast service provider (e.g. a DVB-H operator) wants to offer its premium clients access to higher quality video content via a secondary IP unicast connection (e.g. WLAN).

The underlying idea of the demonstrator is that the different layers of the SVC bitstream are sent using multiple RTP sessions (i.e. one or several SVC layers can be sent in one session). Support for this is included in the RTP standard and it allows flexibility in routing the SVC sub-streams via different network paths between the streaming server and client. In the DVB-H case, we assume that the content is sent by the means of IP multicast.

The demonstrator uses different components of the SCALNET architecture, the main ones being the Streaming Client located in the user terminal and Streaming Server. The user terminal is a laptop computer equipped with multiple network interfaces. In the client terminal, the UCM Client Module can be used for collecting client context information regarding the video stream characteristics (e.g. the bit rates of the SVC layers/sub-streams) and that of the available network interfaces (e.g. available bandwidth). This information can then be used in mapping the SVC layers/sub-streams to the different available network

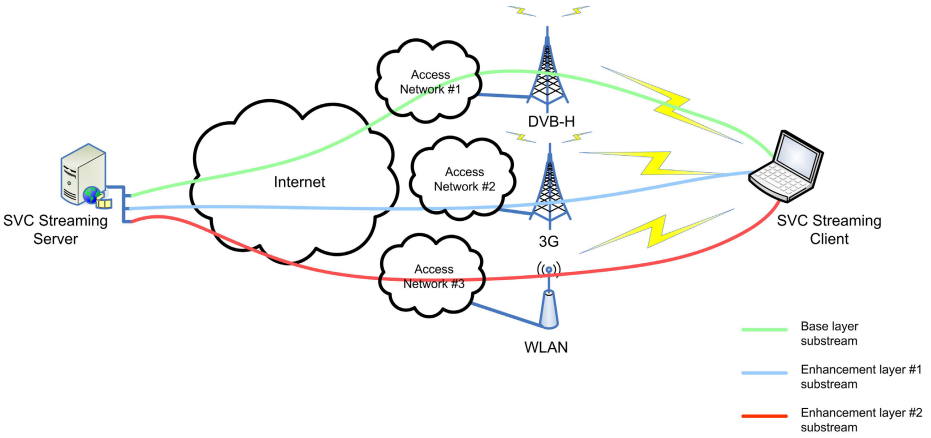

Fig. 5. Multi-interface Streaming demonstrator

interfaces automatically. The inclusion of a MANE into the demo setup is also possible as the MANE shares many of its features with the streaming server. In an envisaged scenario, a MANE located in one of the access networks would filter out excess layers from a SVC sub-stream depending on the transmission conditions in the access network.

5 Conclusions and Future Work

This paper provided an overview of the SCALNET project. It provided an introduction to H.264/SVC and its scalability features. Subsequently, it introduced the SCALNET use cases and its architecture in detail. Finally, we described two of the SCALNET demonstration scenarios, i.e., Adaptive Session Handover and Multi-interface Streaming.

Several future work items can be seen. The SCALNET architecture currently relies on RTSP-based transmission of media. In this context it would be interesting to investigate alternatives, in particular the upcoming HTTP-based streaming mechanisms by IETF [11] and MPEG [2]. Additionally, SCALNET so far focused on the transmission of video only. We need to extend the architecture to also support audio in order to see if this has any additional implications. Moreover, we aim to integrate more efficient SVC codecs into the SCALNET architecture as they become available. Finally, we will evaluate the integration of SCALNET components into the existing systems of SCALNET industrial partners.

Acknowledgments. This work is supported by the Ministerio de Industria, Turismo y Comercio Español (MITYC), the Finnish Funding Agency for Technology and Innovation (Tekes) and the Österreichische Forschungsförderungsgesellschaft mbH (FFG) in the context of the Celtic SCALNET (CP5-022) project.

[2] http://mpeg.chiariglione.org/working_documents.htm#Explorations

References

1. Wiegand, T., Sullivan, G., Schwarz, H., Wien, M. (eds.): ISO/IEC 14496-10:2005/Amd3: Scalable Video Coding. International Standardization Organization (2007)
2. Wiegand, T., Sullivan, G., Bjntegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology 13(7) (July 2003)
3. Segall, A., Zhao, J.: Bit-Stream Rewriting for SVC-to-AVC Conversion. In: Proc. 15th IEEE Int. Conf. on Image Processing (October 2008)
4. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG (eds.): Joint Scalable Video Model. International Standardization Organization (2007)
5. Sablatschan, M., Ransburg, M., Hellwagner, H.: Towards an Improved SVC-to-AVC Rewriter. In: Proc. 2nd International Conference on Advances in Multimedia (June 2010)
6. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology 17(9), 1103–1107 (2007)
7. Wang, Y., Hannuksela, M., Pateaux, S., Eleftheriadis, A., Wenger, S.: System and Transport Interface of SVC. IEEE Transactions on Circuits and Systems for Video Technology 17(9), 1149–1163 (2007)
8. Singer, D., Zubair Visharam, M., Wang, Y., Rathgen, T. (eds.): ISO/IEC 14496-15:2004/Amd2: SVC File Format. International Standardization Organization (2007)
9. Wenger, S., Hannuksela, M.M., Stockhammer, T., Westerlund, M., Singer, D.: RTP Payload Format for H.264 Video. Technical report, Internet Engineering Task Force (February 2005)
10. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications. Technical report, Internet Engineering Task Force, Standard, RFC 3550 (July 2003)
11. Pantos, R.: HTTP Live Streaming. Internet-Draft (work in progress), draft-pantos-http-live-streaming-03 (April 2010) (Expires October 4, 2010)