Open ContEnt Aware Networks

www.ict-ocean.eu

Deliverable D5.5

Distributed delivery methods using Content-aware Network Codes

Public deliverable, Version 2.2, 29 March 2011

Authors

FT	Yannick Le Louédec
AL-BELL	Chris Hawinkel, Danny De Vleeschauwer
IBBT	
IDATE	
N2Nsoft	
UVS	
FRAUNHOFER	Cornelius Hellge, Thomas Schierl, Yago Sanchez de la Fuente
TP	
UERT	

Reviewers Chris Hawinkel, Danny De Vleeschauwer, Yannick Lelouedec

Abstract

This deliverable D5.5 is part of a set of three deliverables D5.5, D5.6 and D5.7 associated with task T5.2. The purpose of this task T5.2 is to investigate the use of Content-aware Network Codes and network coding techniques for highly distributed media delivery to allow efficient delivery to different device types (handheld, UMPC, PC, HD TV set) including media scalability with SVC. While this deliverable D5.5 will investigate promising techniques, deliverable D5.6 and D5.7 will describe the protocol implementation of the technique considered to be the most promising solution for D5.8 in T5.3 and the OCEAN experimental validation in WP6.

© Copyright 2011 OCEAN Consortium

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number 248775

EXECUTIVE SUMMARY

HTTP-Streaming has been gaining popularity in recent years. Contrary to the past tendency of relying on RTP over UDP for multimedia communications thus avoiding the higher end-to-end delay and jitter associated with TCP connections, many content providers have resorted to using HTTP transport (over TCP) for media delivery in cases where the delay constraints allow it. The main reasons for that are first that HTTP is not affected by firewall and network address translation (NAT) traversal issues that exist in traditional streaming scenarios which typically rely on RTP over UDP and second that using HTTP for the file delivery can rely on existing HTTP cache infrastructures on the Internet, consequently substantially relieving the load on the video server and reducing the overall traffic upstream of the cache. Some additional evidence of the increasing interest of the market in HTTP-Streaming is the standardization processes lead by the standardization organizations IETF[1], 3GPP[2], OIPF [3] and MPEG[4].

Dynamic Adaptive Streaming over HTTP (DASH) as defined in [4] refers to a video transport methodology where the clients adapt their requests based on some estimates of their available download rate at regular time instants during the streaming service. For DASH a video is offered in various (typically 4 to 10) versions. Each terminal can choose which version to download depending on its capabilities and the network congestion level. Typically at frequently dispersed time instants during each video there are “switching points” where the client can elect to switch from one quality version to another. One possibility to provide DASH is to encode multiple representations of each of the videos with H.264/AVC at the server and offer them side-by-side. Another is offering all these representations embedded in one file via Scalable Video Coding (SVC). Offering all these representations side-by-side is a high burden on the storage requirements at the origin server as well as on the overall traffic allocation within the CDN. Furthermore, there might also be a decrease in cache performance in comparison to SVC.

The technical benefits as well as the high interest on HTTP streaming within the market as well as in the standardization groups makes HTTP streaming a promising technology for media delivery within the OCEAN system. In this deliverable we describe the OCEAN contribution to the MPEG Dynamic Adaptive Streaming over HTTP (DASH) standardization [41] which enables SVC delivery within DASH.

Furthermore, this deliverable shows coding results for potential configuration for rate adaptation with SVC. Multiple operation points can be achieved with SVC by encoding multiple quality layers. However, each additional quality layer increases the SVC coding penalty. Another possibility is to rely on quality scalability (MGS or CGS), which allows extracting multiple operation points out of one quality layer. Thereby, the SVC penalty can be reduced while multiple operation points can be extracted.

While HTTP streaming, which is based on TCP, will be the focus of this task we further investigated an approach for delivery over unreliable RTP over UDP connection. The third contribution within this deliverable is on content aware network codes focusing on the SVC Layer-Aware FEC (LA-FEC) technology. The basic idea of the proposed Layer-Aware FEC (LA-FEC) approach is to generate the FEC parity data following existing dependencies within the multi-layer media stream in order to improve the robustness of the more important layers. The implementation and application to SVC has already been published by an OCEAN contribution in [42]. Applying a joint decoding, the more important layers are protected by additional parity data, which increases the error correction capabilities of the more important layers without any increase in terms of bitrate. Required signalling for LA-FEC is investigated and further simulation results in a internet transmission scenario show the gain for an unreliable media delivery over UDP.

Copyright © OCEAN Consortium, March 2011

TABLE OF CONTENTS

Copyright © OCEAN Consortium, March 2011

INTRODUCTION

This deliverable D5.5 is part of a set of three deliverables D5.5, D5.6 and D5.7 associated with task T5.2. The purpose of this task T5.2 is to investigate the use of Content-aware Network Codes and network coding techniques for highly distributed media delivery to allow efficient delivery to different device types (handheld, UMPC, PC, HD TV set) including media scalability with SVC. While this deliverable D5.5 will investigate promising techniques, deliverable D5.6 and D5.7 will describe the protocol implementation of the technique considered to be the most promising solution for D5.8 in T5.3 and the OCEAN experimental validation in WP6.

Section 1 describes the state of the art on HTTP streaming for media delivery, including the recent developments in the relevant standardisation bodies consortium members have contributed to. Section 2 analyses the coding performances of SVC. Section 3 introduces a new approach for content aware network codes: SVC layer aware FEC. Section 4 concludes deliverable D5.5 and elaborates on the next steps in Work Package 5.

Copyright © OCEAN Consortium, March 2011

1. HTTP STREAMING FOR MEDIA DELIVERY

HTTP Streaming has been gaining popularity in recent years. Contrary to the past tendency of relying on RTP over UDP for multimedia communications thus avoiding the higher end-to-end delay and jitter imposed by TCP connections, many content providers have resorted to using HTTP transport (over TCP) for media delivery in cases where the delay constraints allow it. In fact, [1] shows that before adaptive streaming over HTTP became popular already a lot of video was delivered over HTTP (often referred to as progressive downloading). Apparently the delay introduced TCP is not a roadblock to many video services. Obviously with the additional feature of adaptivity in DASH the HTTP share in the video traffic is likely to increase more". The main reasons for that are first that HTTP is not affected by firewall and NAT traversal issues that exist in traditional streaming scenarios which typically rely on RTP over UDP and second that using HTTP for the file delivery can substantially relieve the load on the video server by re-using existing HTTP cache infrastructures on the Internet, therefore reducing the overall traffic at the cache feeder link.

Some additional evidence of the increasing interest of the market in HTTP-Streaming is the standardization processes lead by the standardization organizations IETF [1], 3GPP [2], OIPF [3] and MPEG [4].

Dynamic Adaptive Streaming over HTTP (DASH) as defined in [4] refers to a video transport methodology where the clients adapt their requests based on some estimates of their available download rate at every time instant of the streaming service. For DASH a video is offered in various (typically 4 to 10) versions. Each terminal can choose which version to download depending on its capabilities and the network congestion level. This choice is not only made at the beginning of the flow, but at frequently dispersed time instants during the streaming of the video, at which the DASH client can switch from one version to another (for example to alleviate the onset of congestion). Typically this is achieved by segmenting each version in chunks such that the segment boundaries of various versions are aligned in time. All the DASH clients need to do is to consecutively download the most appropriate chunks, based on the information obtained by monitoring recently downloaded chunks of the ongoing movie.

1.1 SVC for adaptive streaming in CDN networks

One possibility to provide DASH is to encode multiple representations of each of the videos with H.264/AVC [6] at the server and offer them side-by-side. Another is offering all these representations embedded in one file via Scalable Video Coding (SVC) [6]. Offering all these representations side-by-side does not only put a high burden on the storage requirements at the origin server, but might also result in a decrease in cache performance in comparison to SVC.

SVC allows embedding multiple substreams within an single bit stream. The base layer of SVC provides the lowest quality level. It is an H.264/AVC compliant bit stream, which ensures backward-compatibility with existing receivers. Each additional decoded enhancement layer increases the video quality in a certain dimension: temporal, spatial, and fidelity scalability. Although the number of such substreams is theoretically unlimited, each additional substream decreases the coding performances compared to a single layer encoding at the same quality. Therefore, there is a trade-off between number of substreams and acceptable coding penalty. Furthermore, due to its layered nature, SVC provides flexibility to DASH, since it allows dividing media content both per SVC layer and per time interval, and thus allows prioritizing very accurately the different elements of the media content according to their importance. Therefore, clients can be designed such that they have a higher responsiveness and better playback quality under adverse network conditions since a request for a time interval is diluted into multiple requests (HTTP GET requests) performed subsequently, one for each of the layers, and when congestion is detected, the client may elect to omit requests for higher

Copyright © OCEAN Consortium, March 2011

layers. Conversely, for the AVC case a unique request is issued for the whole data of a given interval, having to wait longer for the requested data to be downloaded until switching to a lower representation can be performed. The fact that the client needs to issue more HTTP GET requests in SVC case than in the AVC case is not considered to be a major drawback as the (client-to-server) traffic associated with these requests is negligible and HTTP caches can offload the additional processing power that this might introduce on the server.

An added advantage of providing adaptive HTTP streaming using SVC is that content duplication on HTTP servers is avoided and that networks caches such as HTTP proxies and the network itself can be used more efficiently. Figure 1 distinguishes between the feeder link, which is the link from the open internet to the CDN, the CDN Network itself including caches and the last mile delivery to the user. The figure further illustrates the reduced network link usage and how the network cache is used more efficiently when using SVC for adaptive HTTP streaming in comparison to using AVC. This is interesting to Internet Service and Content Delivery Network Providers that typically have to maintain large server and network infrastructures.

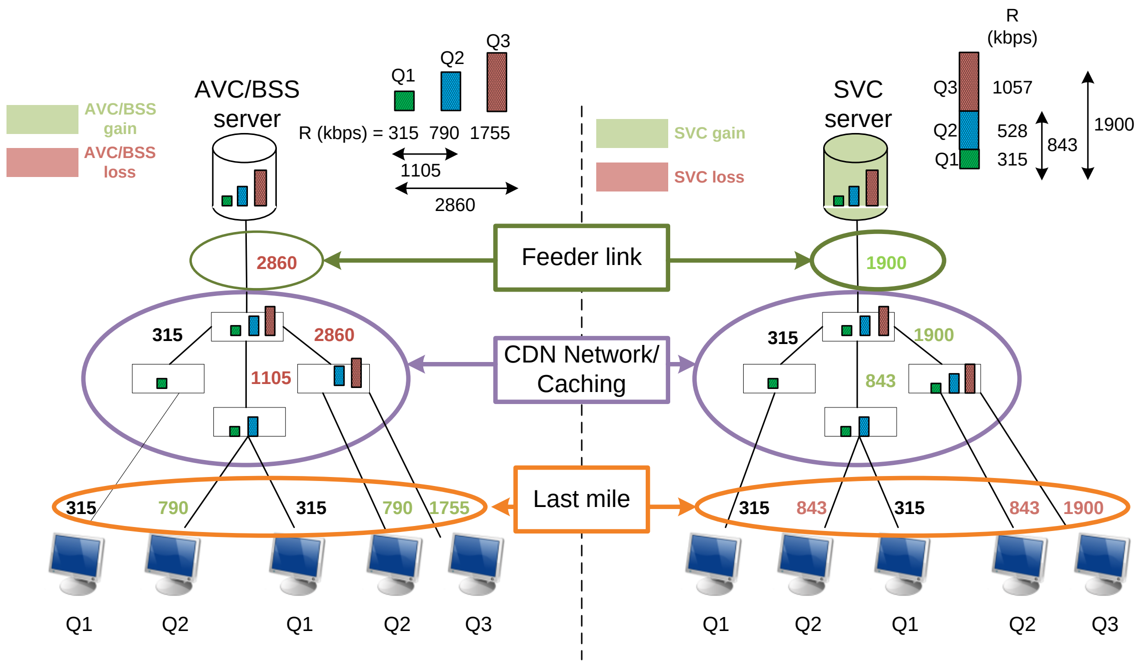


Figure 1: Comparison of CDN with multiple representations using AVC (left) and SVC (right)

In Figure 1 the transmitted rate on the cache feeder link and last mile, as well as the content in network caches are depicted for an example with three different video representations. For the given example an SVC overhead of 10% compared to a single layer encoding at the same level of video quality is considered. Whereas in case of AVC the representations are alternative to each other in case of SVC they are complementary to each other. This means that for AVC three different complete video streams are necessary (Q1,Q2,Q3 in Figure 1) and when one representation is requested the whole video stream has to be transmitted and stored in the network caches. In case of considering SVC, a single encoded video stream is subdivided into layers, each one corresponding to a given quality. Thus, when a representation is requested just the missing layers may be transmitted and stored in the network caches reducing in this way the load on the links and necessary storage in the caches, as exemplarily shown in the figure. Furthermore, the number of clients requesting the same content increases and thus the caching hit ratio increases. Indeed, since all the streaming clients requesting quality Qi with i≥j are expecting to receive a chunk containing layer j, whereas in

Copyright © OCEAN Consortium, March 2011

traditional HTTP streaming with AVC just the streaming clients requesting quality Qj are expecting to receive a chunk for this quality representation.

The influence of SVC on the caching performance and thereby on a reduced feeder link as well as on the last mile will further be investigated in Task T5.3 which combines the outcome of WP4 and WP5. In this deliverable we mainly look at the performance of pure SVC encoding and what is required to enable SVC delivery within MPEG DASH.

1.2 State of the art

1.2.1 ISO base media file format –based Streaming

Adaptive HTTP Streaming (AHS) is based on the idea of adapting the requested media representation to the network condition during streaming. For this purpose it is assumed that the server offers a set of (partial) representations, e.g. as defined in[2], at different media rates, qualities, resolutions or fidelities to allow clients adjusting their requests to the network conditions as well as possibly to the client's capabilities. Herein such presentations are called *alternative* (partial) representations.

A partial representation is a component of the media such as video, audio (language), subtitles or any other part of the media which can be contained in a single ISO base media file format track [21] and thus can be contained in a single 'moof' fragment, also called a segment, representing a certain timeframe of the representation. Following this definition, a partial representation may be also a track containing a subset of an SVC or MVC bitstream [21].

Beside the use of the ISO base media file format, AHS may also support the MPEG-2 Transport Stream container format as, e.g., shown in [23].

Since AHS is based on the ISO base media file format or the MPEG-2 Transport Stream (following [24]), which both are supporting SVC and MVC, the extensions required for supporting multi-layered media are straightforward and limited. AHS mainly requires extensions to the media presentation description and guidelines for using the ISO base media file format or the MPEG-2 Transport Stream in AHS with multi-layer coding standards.

1.2.2 Adaptive HTTP Streaming with single layer codec

One approach for adaptive HTTP streaming is specified in [2]. The main idea of AHS is that the content is stored at the server divided in different segments, and there is a variety of possible representations representing different bitrate encodings of the same content available to be requested by the client, e.g., based on the available downlink transmission rate. Figure 2 and Figure 3 show the content in server and client, respectively.

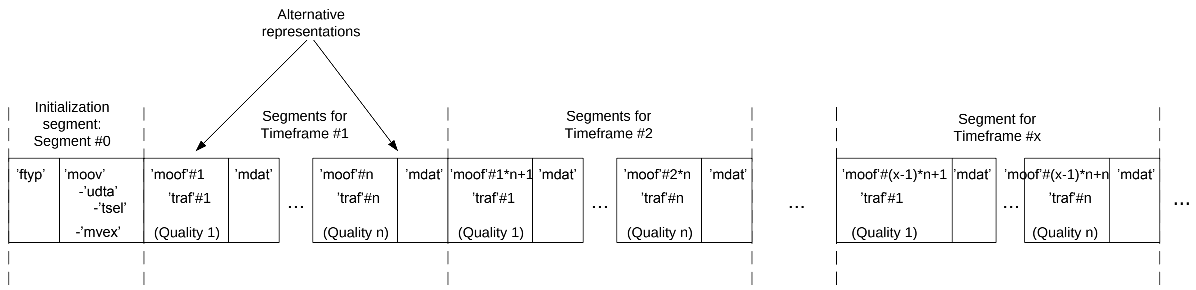


Figure 2: ISO file at the server with single layer video encodings and alternative segments

For simplification, all segments are available as *n* representations associated with different bitrates, thus different sizes of the segments.

Copyright © OCEAN Consortium, March 2011

The client chooses the most adequate representation for each segment/fragment based on, e.g., its available download rate. An example of a downloaded multimedia content to be played is depicted in Figure 3, where different representations have been downloaded. The first segment contains video data of track 1 (Quality 1). Due to a variation on the client's capability, the player downloads the video data from track 3 (Quality 3) for the second segment #2 and so forth.

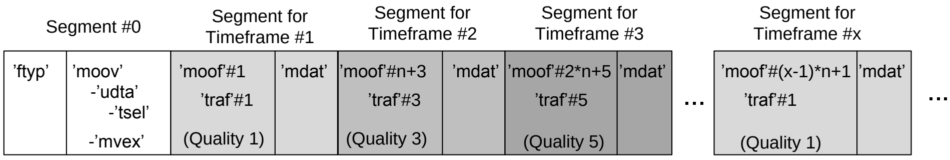


Figure 3: Received ISO file parts at the single layer client

It is important to point out that, due to this adaptation, the received data may consist of fragments containing different tracks, in contrast to commonly used stored content which usually contains a fixed set of tracks for the whole duration of the media file. Therefore, the client has to take care of switching between tracks while loading and decoding newer segments.

1.2.3 Adaptive HTTP Streaming with layered codec

Adaptive streaming using a multi-layer coding standard such as SVC or MVC is similar to AHS as described in section 1.2.2 with the only difference that the video segments are divided in different *complementary* partial representations (video fragments), where possibly multiple complementary partial representations are downloaded for a given timeframe. Which and how many representations are downloaded may depend on the capability of the client or its available downlink rate, i.e., this decision is up to the choice of the client, but the client may need additional information for making such decisions.

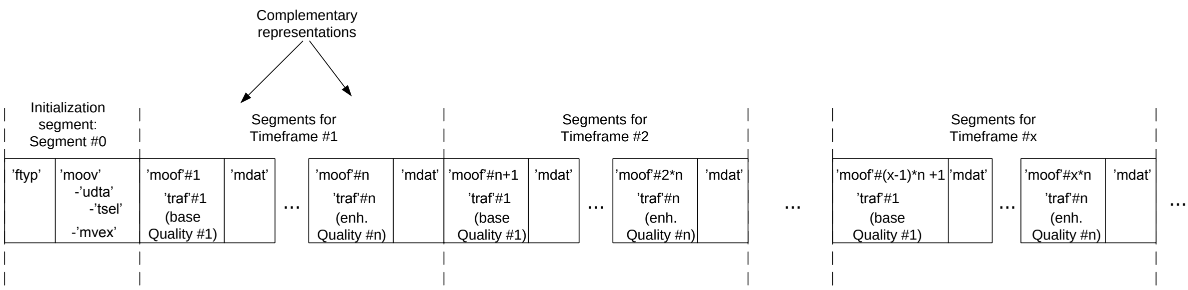


Figure 4: ISO file at the server with layered video encoding and complementary video segments

The main difference to the AHS approach as described [1] is that in case of multi-layer codecs, there may be more than a single video segment for the same timeframe to be jointly decoded at the client, as shown in Figure 5.

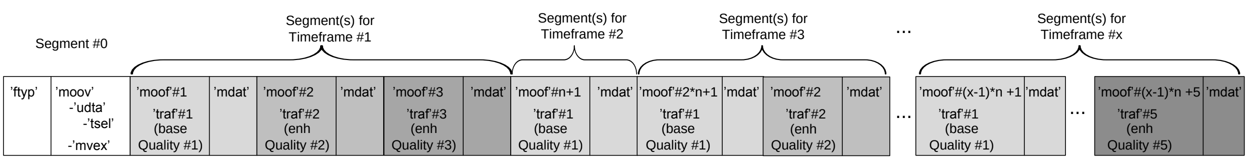


Figure 5: Received ISO file parts at the multi-layer client

Copyright © OCEAN Consortium, March 2011

1.2.4 MPEG 2 – TS-based Streaming

The same AHS concept as for the ISO base media file format can be used with MPEG-2 Transport Stream as container format. Different representations, potentially produced by packet Identifier (PID) filtering, will be considered, and the data will be divided into small chunks (segments), which in this case will be a concatenation of many TS packets of potentially a single PID.

AHS based on MPEG-2 TS is straightforward and no special additions are required, but it is important to remark that synchronization through the Programme Clock Reference (PCR) timestamps would no longer be possible, since all the packets from a given timeframe will be downloaded in a single file and are ready for immediately decoding. This is a general fact rather connected to HTTP Streaming than to the use of multi-layer codecs.

1.3 AHS improvements for multi-layer coding standards - Progress beyond state of the art

1.3.1 Information on ISO base media file format level

As mentioned above, one of the key features of AHS is that the client may switch between tracks in order to allow video adaptation to overcome changes within the network.

The Track Selection Box ('tsel') [21]´in the User Data Box ('udta') within the Track Box ('trak') may be used to provide the player enough information about the offered file in terms of required tracks and contained bitrate/resolution/etc. This information allows for selecting the media content of appropriate quality, which can be deduced by the 'attribute_list' of 'tsel'. It should be noted, that this information may also be obtained by parsing other boxes within the appropriate tracks.

Furthermore, since in a live event the received files could become unacceptably large, not all segments might be saved and kept in memory; therefore, there is a necessity of describing the timing of the fragments, especially, if considering the use of extractors [24]. The new box 'tfdt' [25] within the 'traf' box should be used for this purpose.

1.3.2 Client behavior for track switching

Following [2], in the beginning of a streaming session the initialization segment is requested. This segment typically contains an 'ftyp' box, a 'moov' box and optionally other boxes. In the 'moov' box, the presence of fragments is indicated by the 'mvex' box. The most important aspect of the AHS is that an adaptation may be performed to adjust the video quality to the available downlink rate. In case of single layer codecs (such as AVC), a sole fragment is received per time interval, whereas for layered codecs (such as SVC or MVC) a varying amount of fragments is received. Depending on the track requested in case of single layer transmission or the number of video fragments requested in case of multi-layer transmission, the received media quality is determined.

Due to this main difference, the client functions in different manners. In case of a single layer codec, when receiving a file fragment, the player knows that the track within the fragment for the time fragment can be decoded, but it may have to wait for fragments of other tracks such as audio to synchronously play the media content. Conversely, in case of a layered codec, when receiving a file fragment the client should keep on reading the following fragments to check whether there is another fragment containing another track which is required by the selected track to be jointly decoded. For SVC and MVC, this dependency relation between tracks is indicated in the 'tref' box.

Additional information for selecting between a set of tracks (at certain qualities/resolutions/etc) corresponding to the same SVC or MVC bitstream may be also indicated by the switch group in the 'tsel' box [21].

Copyright © OCEAN Consortium, March 2011

1.3.3 Additional information for media presentation description

1.3.3.1 Dependencies between representations

In order to make the dependencies between representations visible on the media presentation description level, an indication of data similar to the 'tref' information shall be added. Due to the possible presence of multiple dependencies, we propose to add all existing dependencies for a single representation. Therefore, an identifier (ID) is required for indicating each of the representations in order to allow expressing dependencies (DepID) of a representation to other representations in the media presentation.

E.g., the two new attributes 'ID' and 'DepID' may be added to the representation description to the MPD format [24] as follows:

Table 1 - Semantics of additional parameters for Media Presentation Description as specified in [2] (M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory)

Element or Attribute Name	Type (Attribute or Element)	Cardinality	Optionality	Description
MPD	E	1	M	The root element that carries the Media Presentation Description for a Media Presentation.
Period	E	1…N	M	Provides the information of a Period
Representation	E	1..N	M	This element contains a description of a Representation.
ID	A	1	CM Must be present in case of complementary partial representations	Assigns an identifier ID to a presentation in a set of complementary partial representations. Serves at the same time as an indicator for a complementary partial representation.
DepID	A	1	CM Must be present in case of complementary partial representations	Indicates a whitespace separated list of 'ID' attributes indicating all complementary partial representations the representation depends on.

1.3.3.2 The additional XML syntax.

```
<?xml version="1.0" encoding="UTF-8"?>
```


Copyright © OCEAN Consortium, March 2011

```
<xs:schema targetNamespace="urn:MPEG:HTTPStreamingManifest:2010"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="urn:MPEG:HTTPStreamingManifest:2010">
        <xs:annotation>
                <xs:appinfo>MPEG Media Presentation Description</xs:appinfo>
                <xs:documentation xml:lang="en">
                        This Schema defines the MPEG HTTP Streaming Media Presentation Description!
                </xs:documentation>
        </xs:annotation> <!-- MPD: main element -->
        <xs:element name="MPEG-MPD" type="MPDtype"/>

...

<!-- A Representation of the presentation content for a specific Period -->
<xs:complexType name="RepresentationType">
        ...
        <xs:attribute name="ID" type="xs:string"/>
        <xs:attribute name="DepID" type="stringVectorType"/>
        ...
</xs:complexType>

...

<xs:simpleType name="stringVectorType"
  <xs:list itemType="xs:string"/>
</xs:simpleType>

</xs:schema>
```

1.3.3.3 Example

Example for media presentation description as specified in [1]:

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD        xsi:schemaLocation="urn:MPEG:HTTPStreamingManifest:2010        MPEG-Manifest.xsd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="urn:MPEG:HTTPStreamingManifest:2010">
        <Period start="PT0S">
                <Representation mimeType="video/mp4; codecs=svc1" ID="tag0" bandwidth="128000">
                        <SegmentInfo duration="PT10S" baseURL="rep1/">
                                <InitialisationSegmentURL                sourceURL="
http://www.aService.com/aMovie/seg-1-init.mp4"/>
                                <Url sourceURL="seg-1-128k-1.mp4"/>
                                <Url sourceURL="seg-1-128k-2.mp4"/>
                                <Url sourceURL="seg-1-128k-3.mp4"/>
                        </SegmentInfo>
                </Representation>
                <Representation mimeType="video/mp4; codecs=svc1" ID="tag1" DepID="tag0" bandwidth
="256000">
                        <SegmentInfo duration="PT10S" baseURL="rep2/">
                                <InitialisationSegmentURL
sourceURL="http://www.aService.com/aMovie/seg-1-init.mp4"/>
```


Copyright © OCEAN Consortium, March 2011

```
<Url sourceURL="seg-1-256k-1.mp4"/>
<Url sourceURL="seg-1-256k-2.mp4"/>
<Url sourceURL="seg-1-256k-3.mp4"/>
</SegmentInfo>
</Representation>
<Representation mimeType="video/mp4; codecs=svc1" ID="tag2" bandwidth="512000">
<SegmentInfo duration="PT10S" baseURL="rep3/">
<InitialisationSegmentURL
sourceURL="http://www.aService.com/aMovie/seg-2-init.mp4"/>
<Url sourceURL="seg-2-512k-1.mp4"/>
<Url sourceURL="seg-2-512k-2.mp4"/>
<Url sourceURL="seg-2-512k-3.mp4"/>
</SegmentInfo>
</Representation>
<Representation mimeType="video/mp4; codecs=svc1" ID="tag3" DepID="tag2" bandwidth
="768000">
<SegmentInfo duration="PT10S" baseURL="rep4/">
<InitialisationSegmentURL
sourceURL="http://www.aService.com/aMovie/seg-2-init.mp4"/>
<Url sourceURL="seg-2-768k-1.mp4"/>
<Url sourceURL="seg-2-768k-2.mp4"/>
<Url sourceURL="seg-2-768k-3.mp4"/>
</SegmentInfo>
</Representation>
<Representation mimeType="video/mp4; codecs=svc1" ID="tag4" DepID="tag2 tag3"
bandwidth="1024000">
<SegmentInfo duration="PT10S" baseURL="rep5/">
<InitialisationSegmentURL
sourceURL="http://www.aService.com/aMovie/seg-2-init.mp4"/>
<Url sourceURL="seg-2-1024k-1.mp4"/>
<Url sourceURL="seg-2-1024k-2.mp4"/>
<Url sourceURL="seg-2-1024k-3.mp4"/>
</SegmentInfo>
</Representation>
</Period>
</MPD>
```

In the example above, there are two alternative representations (tag0+tag1 and tag2+tag3+tag4). Both of them correspond to SVC bitstreams, each separated into complementary representations.

The receiver could choose between the first and the second SVC alternative streams and, depending on the download rate, complement the representation with the complementary (enhancement) representations.

Copyright © OCEAN Consortium, March 2011

2. SVC CODING PERFORMANCE

In this Section we will analyze the coding performance of SVC.

The selected resolutions for the encodings follows the target resolutions within OCEAN described within D2.2 and listed again in Table 1.

Terminal type	Target display resolution	Framerate
Mobile, TV Set, PC, Laptop	320x240 (QVGA)	25p
	640x480 (VGA)	25p
TV Set, PC, Laptop	1280x720 (720p50)	50p
	1920x1080 (1080p50)	50p

Table 1 - OCEAN targets on screen resolution (From D2.2)

Note that some encodings does not fully match the target resolutions but it can be expected that end devices are able to upscale to the target resolutions without a significant drop in quality.

The results show the overhead of SVC compared to single layer encoding of the highest quality, which influences the required bit rate on the last mile (see Figure 1). Note that the base layer does not show any overhead since with JSVM it is identical to the single layer stream.

Furthermore, we show the gain of SVC compared to encoding single layer encoding of both resolutions which mainly influences the transit link, the CDN network and caching performance (see Figure 1).

Taking this into account it is important to highlight again, that the last mile overhead, as well as the transit link and CDN network and caching performance is not only influenced by the sheer encoding performance but also by other parameters which will be further evaluated within the upcoming deliverables.

Within OCEAN, we can consider two possible encoding settings:

1. **Support multiple target resolutions (bit rates) within one stream**

   - Spatial and temporal scalability

   - Support multiple device capabilities (e.g. 720p50/1080p50 or 480p25/720p50/1080p50)

   - Rate adaptation points depend on number of quality layers

   - Last mile overhead depends on the number of quality layers

2. **Support multiple bit rates for one target resolution within one SVC stream**

   - Quality scalability with MGS/CGS

   - Multiple operation points out of a single quality layer

   - Small last mile overhead

Copyright © OCEAN Consortium, March 2011

The encodings have been performed using the JSVM 9.17 software and an optimized encoder version which employs the R-D optimized multi-layer encoder control as shown in [44][45]. In the following we will explain some basics on the encoder in Section 2.1 and features of SVC like unequal random access point frequency in Section 2.2 and multiple operation points with MGS scalability in Section 2.3 before we show the simulation results in Section 2.4 and Section 2.4.2 for an optimized encoding to show the full potential of SVC and give a conclusion in Section 2.5.3

2.1 JSVM

The JSVM (Joint Scalable Video Model) software [42] is the reference software for the Scalable Video Coding (SVC) project of the Joint Video Team (JVT) of the ISO/IEC Moving Pictures Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). JSVM can be used to encode standard compliant H.264/AVC streams as well as scalable video streams The main purpose of the JSVM is to reflect the standard, which only specifies the decoder. Therefore, there is room for encoder optimizations beyond JSVM as shown in [44][45] and in Section 2.5.

2.2 Unequal Random Access Point Frequency

SVC offers the possibility to provide Random Access Points (RAPs) for the different layers (i.e., different values of dependency_id) of a bitstream at different time instances (i.e., in different access units). An SVC RAP for a particular layer (i.e., a particular value of dependency_id) enables a decoder to start decoding the particular layer, but in general it is not possible to start decoding any other layer. An SVC IRD can start decoding (and displaying pictures) at each present RAP. In the worst case, it can start decoding (and displaying) only the base layer, and after a small period of time it can continue with decoding (and displaying) the enhancement layer. At the same time, a larger maximum interval between enhancement layer RAPs enables providing a coding efficiency that is very close to that of single layer coding, since intra pictures (which require a larger number of bits than inter-predicted pictures) can be coded less often.

Decreasing the frequency of enhancement layer RAPs can result in a significantly increased coding efficiency while providing the same channel switching delay (when the base layer is decoded and displayed as long as no enhancement layer RAP was received).

In Figure 6, the decoding process is illustrated for an example of accessing an SVC bitstream at a base layer RAP. The decoding process starts with decoding the base layer representation for the base layer RAP and all access units that follow the base layer RAP and precede the enhancement layer RAP in decoding order. For the enhancement layer RAP and all access units that follow the enhancement layer RAP in decoding order, the enhancement layer representations are decoded. For the base layer RAP and all access units that follow the base layer RAP in output order and precede the enhancement layer RAP in decoding order, the base layer representations are output. For the enhancement layer RAP and all access units that follow the enhancement layer RAP in output order, the enhancement layer representations are output. No pictures are output for the access units that follow the enhancement layer RAP in decoding order but precede it in output order.

In the following we will refer to it as unequal RAP frequency. Whereas equal RAP frequency refers to having the same RAP frequency in both layers.

Copyright © OCEAN Consortium, March 2011

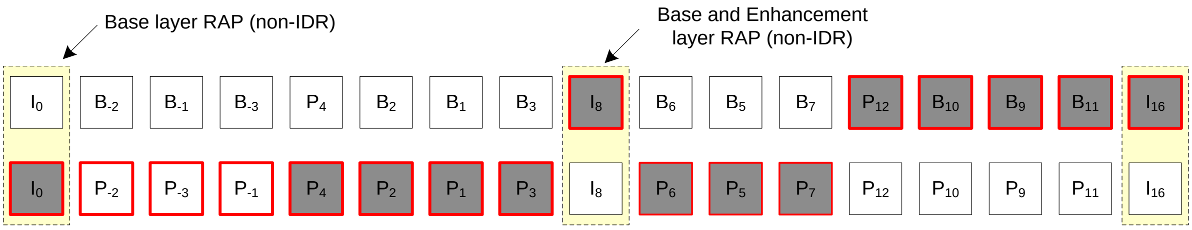


Figure 6: Illustration of the decoding process with output picture skipping when accessing a two-layer SVC bitstream at a base layer RAP using Scalable Baseline Profile. The access units are displayed in decoding order (from left to right). The subscript numbers indicate the output order. The representations that are decoded are marked with red frames; the representations that are output are marked grey.

2.3 Multiple operation points with MGS/CGS scalability

The MGS scalability allows embedding multiple operation points within of one or two quality layers while keeping the framerate constant.

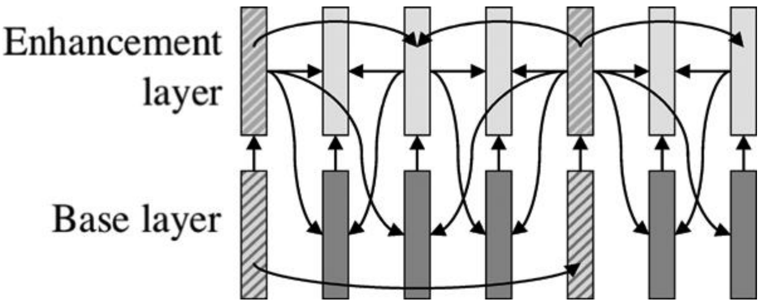


Figure 7: MGS prediction structure

The following figures show how multiple operation points can be achieved for an exemplary stream with two quality layers by dropping NALunits of a specific temporal layer and quality layer. Figure 8 shows an exemplary setting with two quality layers (Q1 and Q2). The figures show GOP 8 with an hierarchical prediction structure resulting in four temporal layers (T0,T1,T2,T3). It is important to note here, that base layer units are not dropped and therefore the framerate is kept constant within all operation points. The full video stream has a total bitrate of 7000 kbps. Dropping the second quality layer of the highest temporal layer reduces the bit rate to 6380 kbps while reducing the quality of every second picture at the same time. The other figures show how the bit rate can be further reduced by dropping additional packets and resulting in exemplary 9 operation points out of two quality layers. The subjective quality depends on the selection of operation points.

Copyright © OCEAN Consortium, March 2011

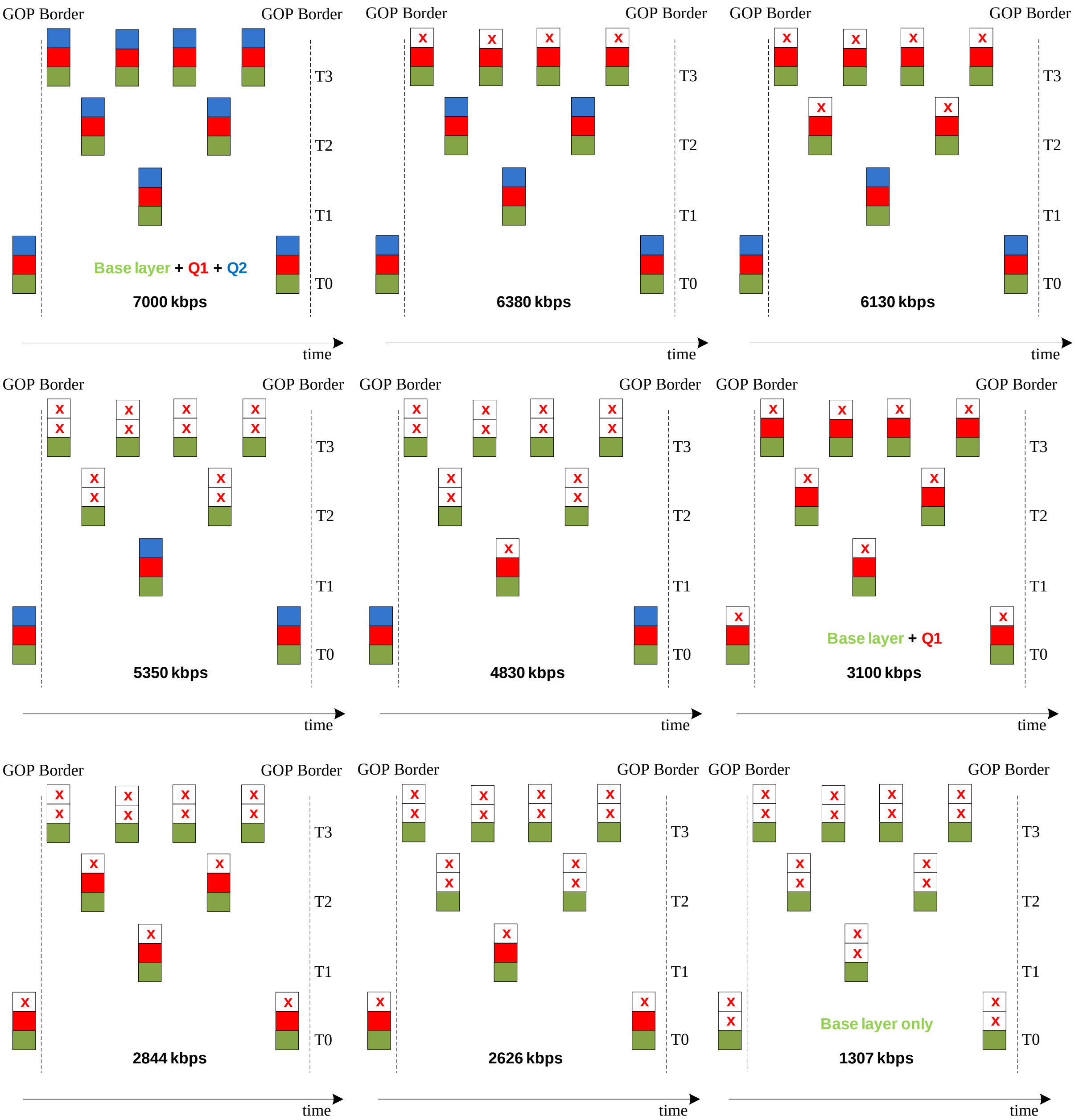


Figure 8: Multiple operation points with MGS scalability

Note 1. Further analysis on MGS operation point selection will be performed within the on-going FP7 OPTIBAND project. Exchanges between OCEAN and OPTIBAND project are envisioned in this domain.

Note 2. Such a multiple operation point extraction can be also applied with CGS coding, which has not been tested within this deliverable.

Copyright © OCEAN Consortium, March 2011

2.4 Encoding results with JSVM

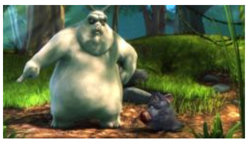
2.4.1 Encoding parameters

This section shows coding results performed within OCEAN. the sequences and common coding parameters used for the coding results based on the JSVM 9.17. All encodings are based on Scalable High (SVC) and High profile (H.264/AVC).

For better interpretation of the presented results, it should be noted, that as a reasonable quality point for the video sequences, an PSNR value of 34 dB is assumed.

The test sequences are listed in Table 1 and the common coding parameters are shown in Table 3

Table 2: Sequences

Sequence	Coding difficulty	Original	Generation of input sequences for each UC
 Aloha Wave	medium-high critical	1920x1080p, 50Hz	Down-sampling using the JSVM software
 Police Boat	highly critical	1920x1080p, 50Hz	Down-sampling using the JSVM software
 Crowd Run	medium critical	1920x1080p, 50Hz	Down-sampling using the JSVM software
 Old Town Pane	low critical	1920x1080p, 50Hz	Down-sampling using the JSVM software
 Ice Dance	low-medium critical	1920x1080p, 50Hz	Down-sampling using the JSVM software
 Big Buck Bunny	Low	1920x1080p, 50Hz	Down-sampling using the JSVM software

Copyright © OCEAN Consortium, March 2011

Table 3: Common coding parameters

General	
Coding structure	Hierarchical B pictures (GOP size of 8 pictures / 4 hierarchy levels)
Entropy coding	CABAC
8x8 Transform	Enabled
Deblocking	Enabled

2.4.2 Encoding results 1.Option: Support multiple target resolutions (bit rates) within one stream

2.4.2.1 2 layers - 1280x720@50Hz - 1920x1080@50Hz

In this section we show encodings which allow the support of 720p50 and 1080p50 within one SVC stream. In addition this encoding allows for one rate operation point (apart from temporal scalability).

Table 4: Encoding settings for 1280x720@50Hz - 1920x1080@50Hz

	SVC RAP 0.49s/0.49s	**SVC RAP 0.49s/1.92s**	**SL RAP 0.49s**
Base layer (and corresponding single layer)			
Profile	High		
Format	1280x720, 50 Hz		
RAP (I frame) intervall	24 frames (0.48s)		
Enhancement layer (and corresponding single layer			
Profile	Scalable High		High
Format	1920x1080, 50 Hz		
RAP (I frame) intervall	24 frames (0.48s)	96 frames (1.92s)	24 frames (0.48s)
Delta QP	0		-

Copyright © OCEAN Consortium, March 2011

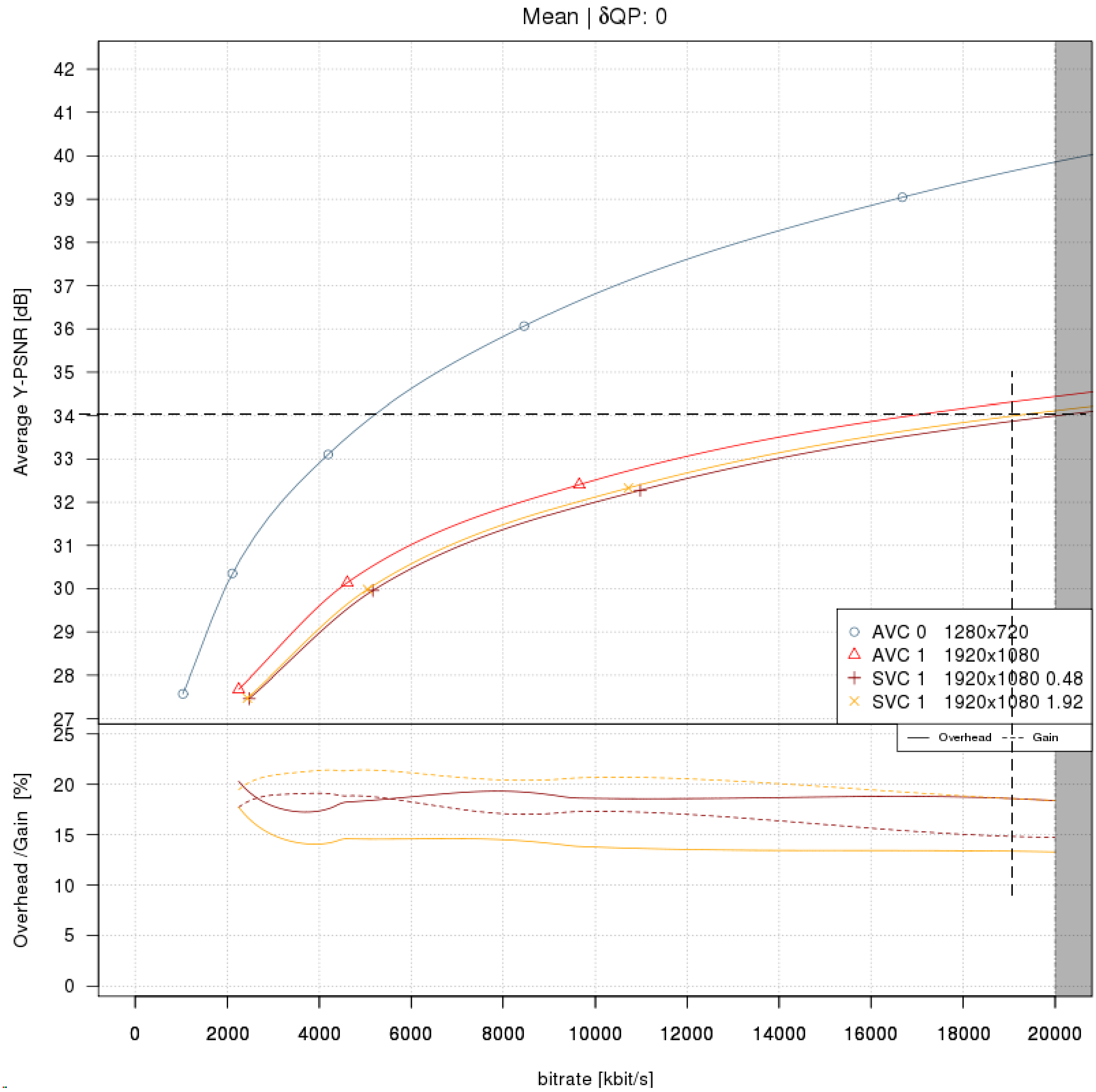


Figure 9: Mean R-D plot for 1280x720@50Hz - 1920x1080@50Hz spatial scalability with a target video quality of approx. 34 dB.

Figure 9 shows that with two spatial layers and a factor of 1.5 between the two resolutions and unequal random access point frequency the SVC penalty compared to SL1 is below 15% and the gain compare to simulcast (SL0+SL1) is over 20% at the target video quality of approx 34 dB.

2.4.2.2 2 layers - 960x540@50Hz - 1920x1080@50Hz

In this section we show encodings which allow the support of 960x540@50Hz and 1920x1080@50Hz which shows a significant increased encoding performance compared to 720p50 and 1080p50 due to the factor 2 between the resolutions. Such an encoding could be used to have one rate adaptation point for below 1080p and in parallel supporting two resolutions.

Copyright © OCEAN Consortium, March 2011

Table 5: Encoding settings for 960x540@50Hz - 1920x1080@50Hz

	SVC RAP 0.48s/0.48s	SVC RAP 0.48s/1.92s	SL RAP 0.48s
Base layer (and corresponding single layer)			
Profile	High		
Format	960x540, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)		
Enhancement layer (and corresponding single layer			
Profile	Scalable High		High
Format	1920x1080, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)	96 frames (1.92s)	24 frames (0.48s)
Delta QP	0		-

Copyright © OCEAN Consortium, March 2011

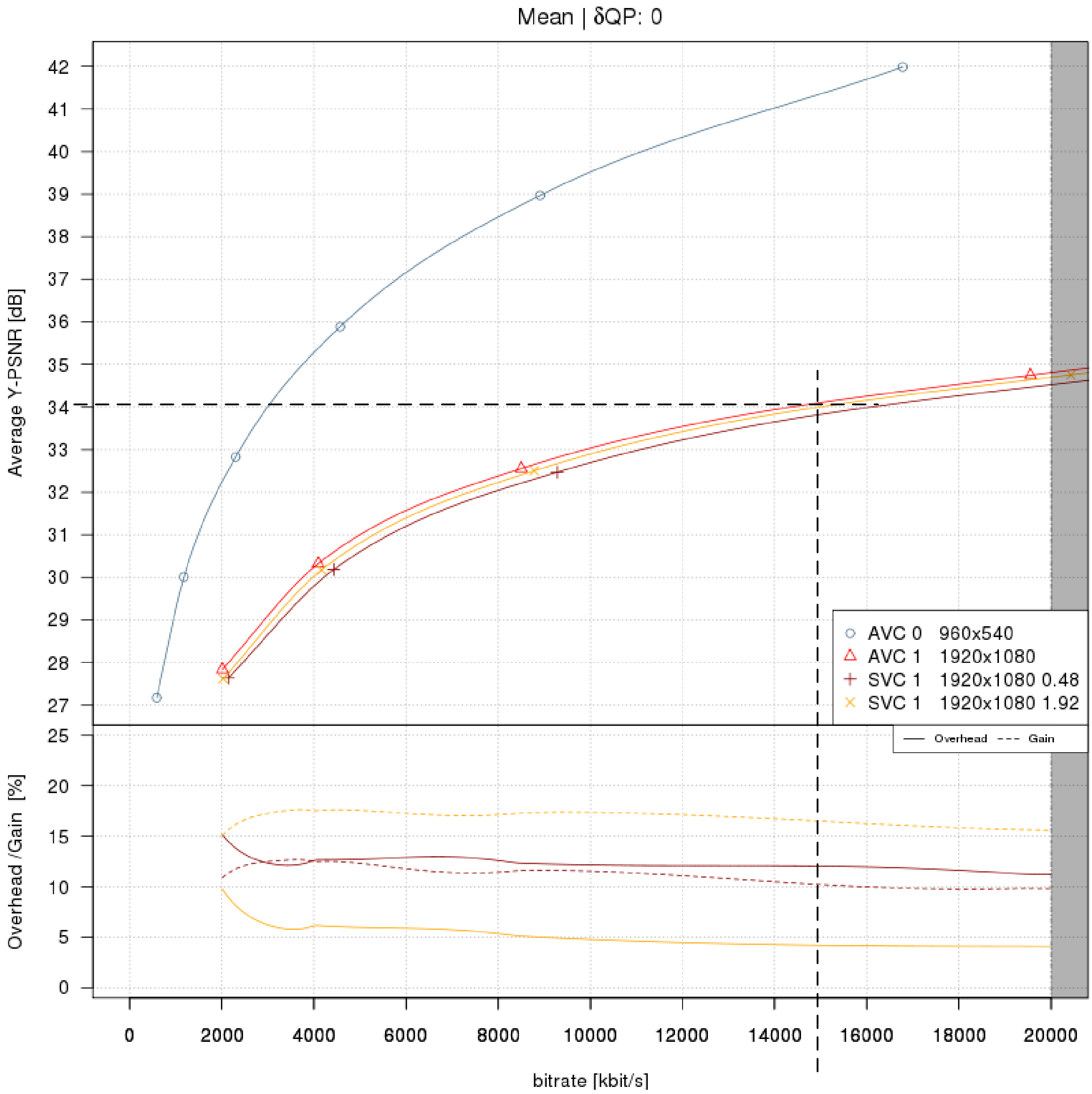


Figure 10: Mean R-D plot for 960x540@50Hz - 1920x1080@50Hz spatial scalability at a target quality in terms of PSNR of approx. 34dB

Figure 10 shows that with two spatial layers and an factor of two between the resolutions and unequal random access point frequency the SVC penalty compared to AVC1 is below 5% and the gain compared to simulcast (AVC0+AVC1) is over 15% at the target video quality of approx. 34 dB.

2.4.2.3 2 and 3 layers - 848x480@50Hz - 1272x720@50Hz - 1908x1080@50Hz

In this section we show encodings which allow the support of three resolutions 848x480@50Hz, 1272x720@50Hz, and 1908x1080@50Hz Such an encoding could be used to have three rate

Copyright © OCEAN Consortium, March 2011

adaptation point below 1080p (apart from temporal scalability) and in parallel supporting three resolutions.

Table 6: Encoding settings for 848x480@50Hz - 1272x720@50Hz - 1908x1080@50Hz

	SVC RAP 0.49s/0.49s	**SVC RAP 0.49s/1.92s**	**SL RAP 0.49s**
Base layer (SL0) (and corresponding single layer)			
Profile	High		
Format	848x480, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)		
Enhancement layer 1 (SVC1) (and corresponding single layer)			
Profile	Scalable High		High
Format	1272x720, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)	96 frames (1.92s)	24 frames (0.48s)
Delta QP	0		-
Enhancement layer 2 (SVC 2) (and corresponding single layer)			
Profile	Scalable High		High
Format	1908x1080, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)	96 frames (1.92s)	24 frames (0.48s)
Delta QP	0		-

Copyright © OCEAN Consortium, March 2011

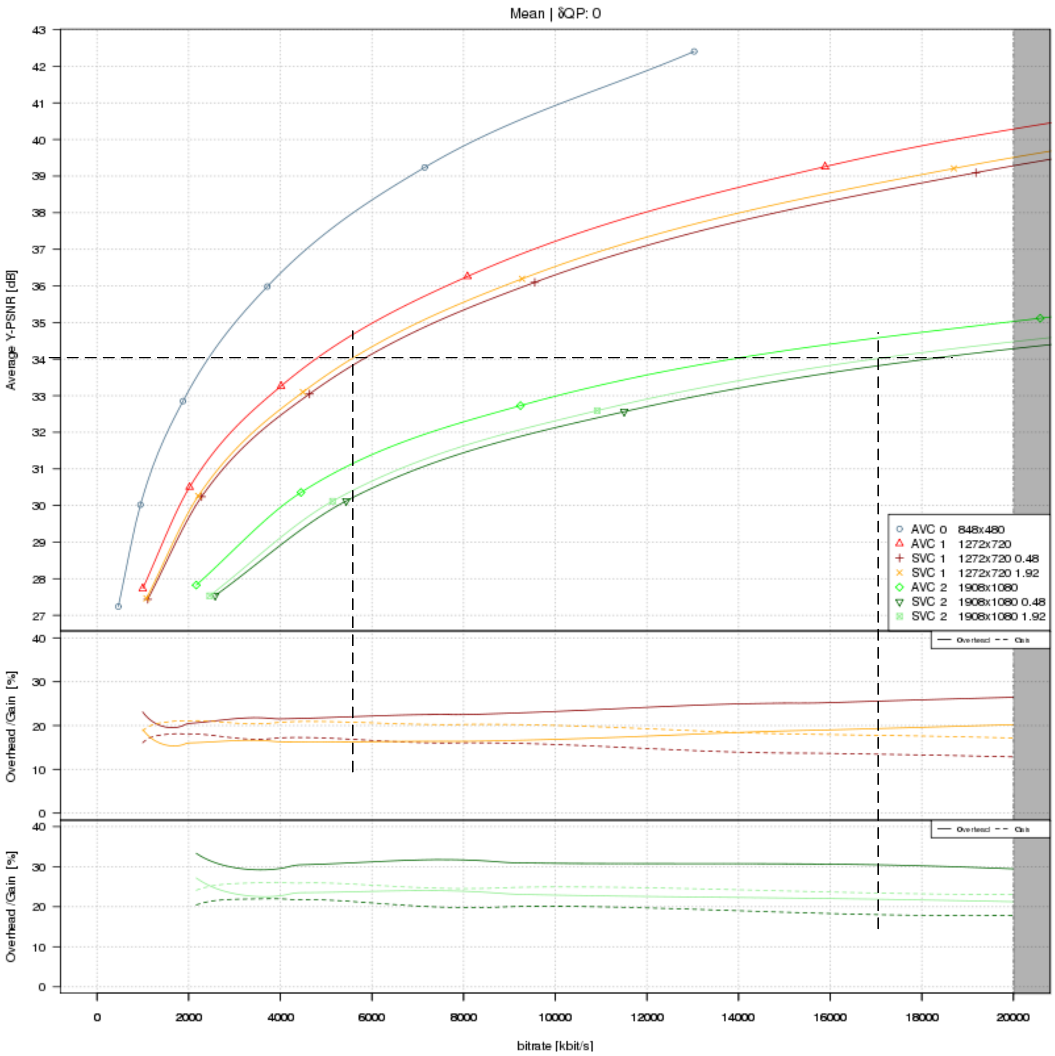


Figure 11: Mean R-D plot for 848x480@50Hz - 1272x720@50Hz - 1908x1080@50Hz spatial scalability at target quality of 34dB.

Figure 11 shows the overhead and gain for two and three spatial layers and an factor of 1.5 between the different resolutions and unequal random access point frequency. For two resolutions (upper overhead/gain plot) the results show an overhead approx. 15% and a gain of more than 20% for a video quality of 34dB. For three layers (lower overhead/gain plot), the SVC overhead compared to AVC2 is around 20% and the gain compared to simulcast (AVC0+AVC1+AVC2) is increased to 30% at the target video quality of approx. 34 dB.

2.4.3 Encoding results 2.Option: Support multiple bit rates for one target resolution within one SVC stream

2.4.3.1 2 layers - MGS scalability 1280x720@50Hz

In this section we show encodings for quality scalability using MGS, which allows the support of multiple operation points (see Section 2.3) out of one quality enhancement layer. The target resolution is 720p50.

Copyright © OCEAN Consortium, March 2011

Table 7: Encoding settings for 2 layers using MGS scalability 1280x720@50Hz

	SVC RAP 0.48s/0.48s	SL RAP 0.48s
Base layer (and corresponding single layer)		
Profile	High	
Format	1280x720, 50 Hz	
RAP (I frame) interval	24 frames (0.48s)	
Enhancement layer (and corresponding single layer		
Profile	Scalable High	High
Format	1280x720, 50 Hz	
RAP (I frame) interval	24 frames (0.48s)	24 frames (0.48s)
Delta QP	8	-

Copyright © OCEAN Consortium, March 2011

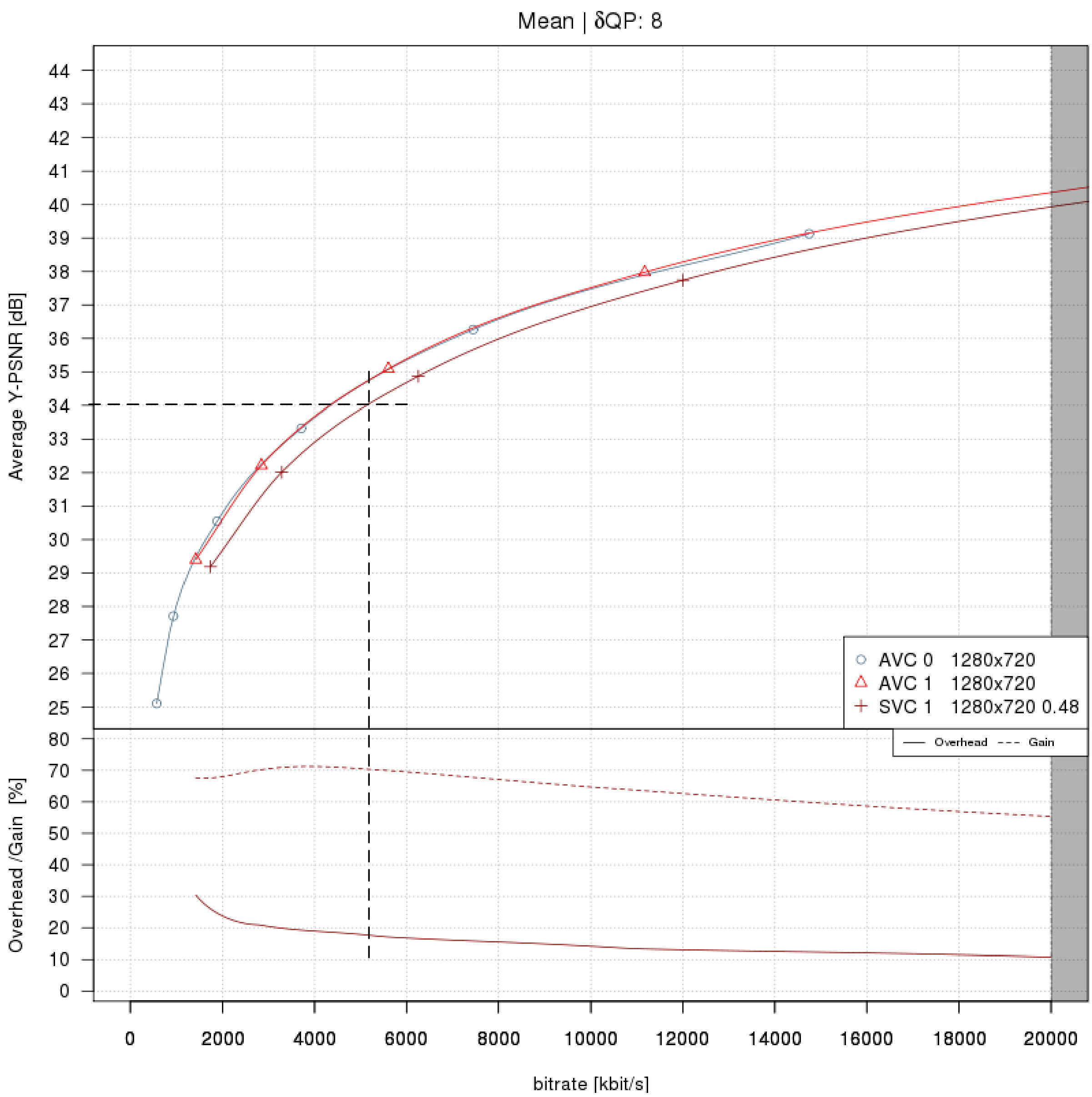


Figure 12: Mean R-D plot for 1 quality layer using MGS scalability 1280x720@50Hz at target video quality of 34dB.

Figure 12 shows the overhead and gain for two quality layers using MGS coding. At the target video quality of 34dB, the SVC penalty compared to AVC1 is below 20%. The gain compared to simulcast (AVC0+AVC1) is above 70%. Note that the overhead is decreasing and the gain is increasing with an increasing bitrate/quality.

2.4.3.2 2 layers - CGS scalability 1280x720@50Hz

In this section we show encodings for quality scalability using CGS, which allows the support of multiple operation points (see Section 2.3) out of one quality enhancement layer. The target resolution is 720p50.

Copyright © OCEAN Consortium, March 2011

Table 8: Encoding settings for 2 layers using MGS scalability 1280x720@50Hz

	SVC RAP 0.48s/0.49s	**SVC RAP 0.48s/1.92s**	**SL RAP 0.48s**
Base layer (and corresponding single layer)			
Profile	High		
Format	1280x720, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)		
Enhancement layer (and corresponding single layer			
Profile	Scalable High		High
Format	1280x720, 50 Hz		
RAP (I frame) interval	24 frames (0.48s)		24 frames (0.48s)
Delta QP	8		-

Copyright © OCEAN Consortium, March 2011

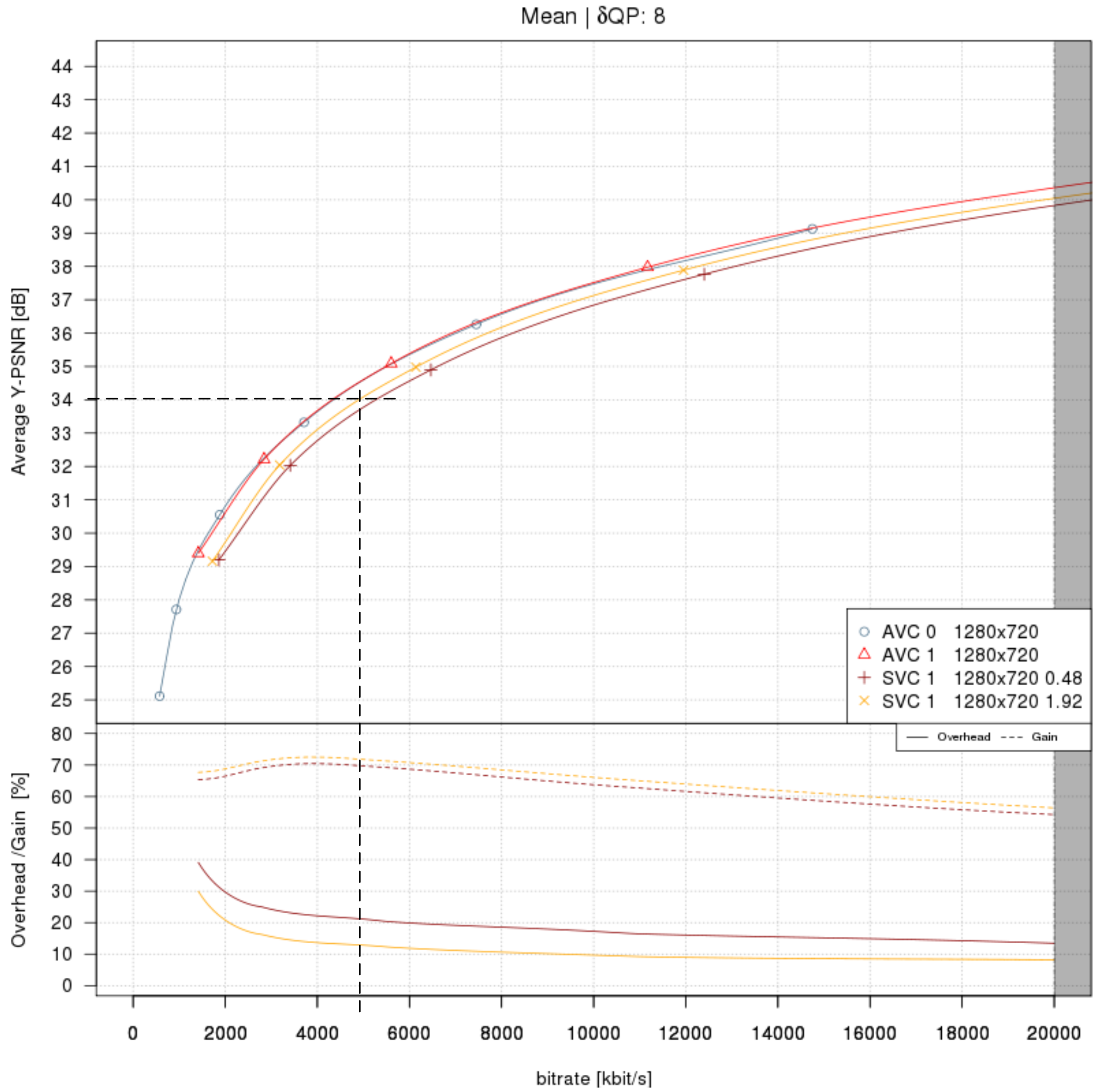


Figure 13: Mean R-D plot for 1 quality layer using CGS scalability 1280x720@50Hz

Figure 13 shows the overhead and gain for two quality layers using CGS coding using equal and unequal RAP frequency. With unequal RAP and at the target video quality of 34dB the SVC penalty compared to AVC1 is below 15%. The gain compared to simulcast (AVC0+AVC1) is above 70%. Note that the overhead is decreasing and the gain is increasing with an increasing bitrate/quality.

2.5 Encoding results with optimized JSVM

The results within this section have been presented already to the DVB tm-avc group in [46]. They give an impression of the potential of SVC encoder beyond JSVM, with an R-D optimized multi-layer encoder. They have been performed with different sequences than the encodings in Section 2.4.

Copyright © OCEAN Consortium, March 2011

2.5.1 Encoding parameters

Table 9: Test sequenced

Sequence	Format	Original	Generation of input sequences
Aloha Wave	1280x704, 50 Hz 640x352, 50 Hz (base layer)	720p, 50Hz	– **enhancement layer:** cropping of a 1280x704 area (in luma samples) from the center of original sequence – **base layer:** downsampling of the enhancement layer signal using the JSVM software
Big Ships			
City			
Old Town Pan			
Sailormen			

Table 10: Coding parameters

	SVC RAP 0.5s / 0.5s	SVC RAP 0.5s / 2.5s	SL RAP 0.5s
general			
Coding structure	hierarchical B pictures (GOP size of 8 pictures / 4 hierarchy levels)		
Entropy coding	CABAC		
8x8 Transform	enabled		
Deblocking	enabled		
Weighted prediction	disabled		
base layer (and corresponding single layer)			
Profile	High		
Format	640x352, 50 Hz		
RAP (I frame) interval	24 frames (0.48 s)		
QP	36, 32, 28		38, 35, 32, 29, 26
enhancement layer (and corresponding single layer)			
Profile	Scalable High		High
Format	1280x704, 50 Hz		
RAP (I frame) interval	24 frames (0.48 s)	120 frames (2.4 s)	24 frames (0.48 s)

2.5.2 Encoding results spatial scalability

Table 11: Summary of results

	Average rate savings against SVC RAP 0.5s / 0.5s	Average rate overhead against single layer coding (SL RAP 0.5s)
Aloha Wave	8.96 %	0.61 %

Copyright © OCEAN Consortium, March 2011

Big Ships	7.51 %	-0.62 %
City	12.15 %	-3.49 %
Old Town Pan	23.43 %	-7.75 %
Sailormen	4.45 %	1.77 %
Average	**11.3 %**	**-1.90 %**

The results show, that with an optimized SVC encoder in combination with unequal random access point efficiency improvement, the SVC penalty can be significantly reduced. For some sequences it is even possible to outperform the single layer coding due to the unequal random access point gain.

2.5.3 Conclusion

Some important features for SVC like unequal random access point frequency and multiple operation points with MGS or CGS scalability have been shown in this section. The presented encoding results based on JSVM show the penalty and pure coding gain introduced by SVC for spatial and quality scalability. The penalty depends on the encoding setup and lies between 5% and 30% overhead, whereas the gain is between 15% and 70% compared to simulcast transmission. Furthermore, encoding results with an optimized SVC encoder show, that there is room for a significant performance optimization beyond the JSVM.

The combination of quality scalability and the extraction of multiple operation points as shown in Section 2.3 seems to be a very promising solution for rate adaptation within OCEAN. If we take the MGS/CGS coding overhead with one quality layer of about 15%, we can already extract five operation points with this penalty. In addition the gain compared to 5 operation points with single layer is significant. The gain of such an SVC stream within an CDN system with caching will further be evaluated within the next deliverable D5.6. and within task T5.3 in deliverable D5.8.

Copyright © OCEAN Consortium, March 2011

3. CONTENT AWARE NETWORK CODES -SVC LAYER-AWARE FEC

3.1 State of the art

In modern video codecs, various dependency structures can be used. One important dependency structure is introduced by motion compensation, where a picture reference (e.g. from the past) is used to predict another picture. Another set of dependency structures is introduced by layered video coding, such as scalable SVC or MVC, where a base layer is referenced by one or more enhancement layers as shown in Figure 14.

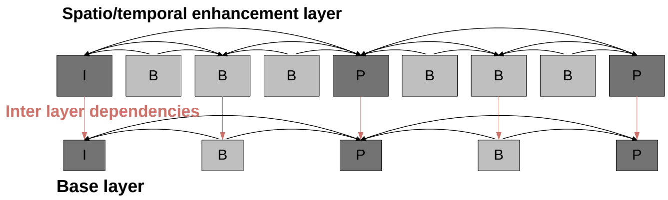


Figure 14: Dependencies within a temporal/spatial SVC stream due to hierarchical prediction structure and inter-layer prediction.

Such an enhancement layer can be further referenced by other enhancement layers and so on. A loss of a picture affects all referencing pictures in some way. Using multi-layer media streams, each quality layer has a different level of importance in the decoding process. If a lower layer is lost, higher enhancement layers cannot be decoded due to missing references as depicted in Figure 15. In the given example with *N* quality layers, Layer 1 is lost due to transmission errors, so that all predicting layers cannot be decoded. Hence, it is desirable to strengthen the error robustness of more important pictures or layers.

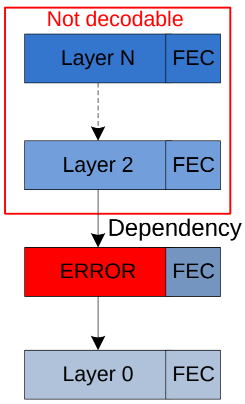


Figure 15: Transmission errors affect all dependent quality layers and parity data of standard FEC schemes (ST-FEC) become useless.

In SVC, the base layer is more important than the enhancement layer due to the enhancement layer information becomes useless if the base layer information is lost. Hence, a differentiation in robustness is in general beneficial for the transmission of SVC, where the base layer gets a stronger protection than the enhancement layers. Therefore, typically unequal error protection (UEP) is used, to give the more important layers a stronger robustness.

The basic idea of the proposed Layer-Aware FEC (LA-FEC) approach is to generate the FEC parity data following existing dependencies within the multi-layer media stream in order to improve the robustness of the more important layers. Applying a joint decoding, the more important layers are protected by additional parity data, which increase the error correction capabilities of the more

Copyright © OCEAN Consortium, March 2011

important layers without any increase in terms of bitrate. The scheme in Figure 16 illustrates the cross layer FEC generation. While the base layer (layer 0) FEC generation is not changed, the FEC data of layer 1 is generated across source symbols of layer 0, FEC data of layer 2 is generated across layer 1 and layer 0 up to the FEC data of layer N, which is generated across the source symbols of all dependent layers. As a generic FEC approach, LA-FEC can be implemented at any layer (physical, link, or application layer), and to any FEC code (such as LDPC, Raptor, or Reed-Solomon codes), by simply extending the encoding process of the enhancement layers over all dependent layers.

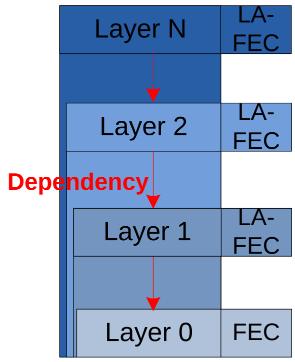


Figure 16: Generation of FEC data by LA-FEC across layers following dependency within the media stream.

To illustrate the principle of the LA-FEC approach we apply a simple FEC algorithm which generates parity bits by XOR combinations of source symbols (one bit per symbol). Figure 16 compares the encoding process, and Figure 17 the decoding process of standard FEC (ST-FEC) on the left side and LA-FEC on the right side. LA-FEC modifications are marked in green. In the given example, which is based on an erasure channel (erroneous packets are treated as lost packets), there are two quality layers, where Layer 1 depends on Layer 0 due to inter-layer prediction within the media stream (e.g. SVC). Each layer consists of three source symbols and two parity symbols.

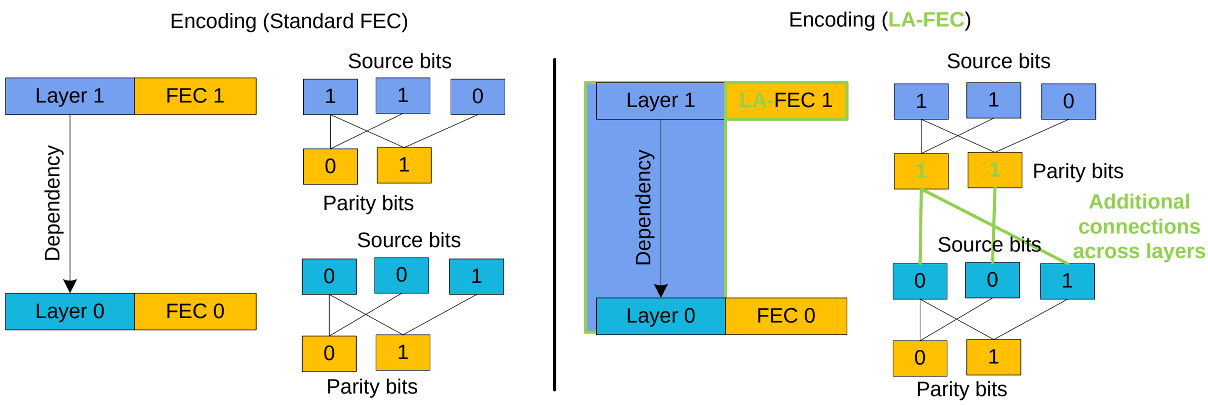


Figure 17: Encoding for standard FEC (left) and LA-FEC (right). LA-FEC extends generation of parity bits across layer 0 symbols.

Copyright © OCEAN Consortium, March 2011

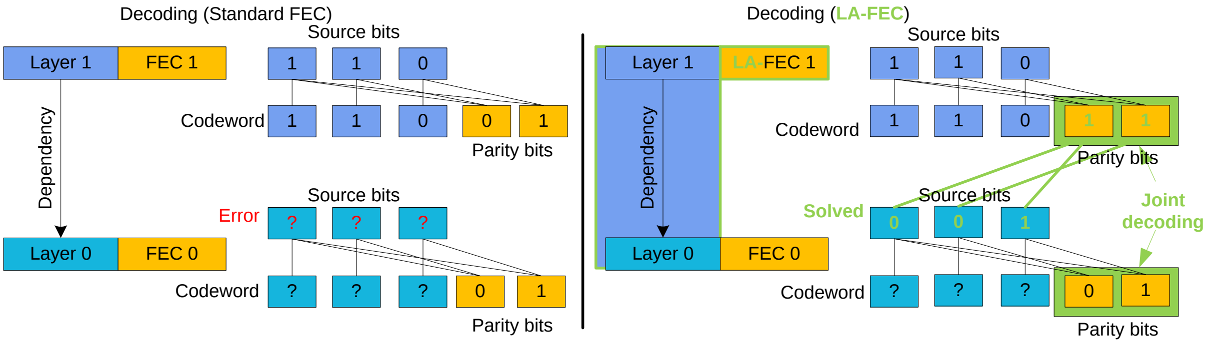


Figure 18: Decoding of standard FEC and LA-FEC. Using LA-FEC the parity bits of both layers can be used for a combined decoding.

Regarding the encoding process in Figure 16, the parity bits are computed by a simple XORing process of the source bits. Using ST-FEC, the XORing process is applied within the current layer, whereas using LA-FEC, the XORing process is extended across layers following existing dependencies. Hence, the parity bits of Layer 1 are generated over the source bits of both layers, Layer 0 and Layer 1, and can further be used for error correction of both layers together with the parity bits of Layer 0. It should be noted here, applying LA-FEC to realistic FEC algorithms also the source symbols may be connected to dependent layers. After encoding, the source and parity bits of each layer are combined to a codeword and transmitted over an error prone channel.

In the outlined decoding example in Figure 17, there are three transmission errors within the codeword of Layer 0 marked by "?" and no errors in Layer 1. Using ST-FEC, there are not enough parity bits within Layer 0. Therefore it cannot be corrected. Although Layer 1 is successfully received, it cannot be used due to the missing dependencies in Layer 0. Using the LA-FEC, the parity bits of Layer 1 can be used together with the parity bits of Layer 0 for correcting Layer 0. In the given example, only with LA-FEC both layers can be corrected. With LA-FEC, the enhancement layer cannot be independently corrected of the base layer. Therefore, the improvement in base layer protection comes at the expense of a reduced protection of the enhancement layer. Nevertheless, due to the existing dependencies within the media stream, in such cases where the base layer is lost, the enhancement layer data cannot be used anyway. Therefore, LA-FEC never performs worse than the ST-FEC in terms of video quality.

3.1.1 Layer-Aware Raptor code

Raptor codes are one of the first known classes of fountain code with linear time encoding and decoding [26]. The design of Raptor consist of a fixed rate 'precode', typically any erasure code. The precoded symbols are forwarded to the LT-encoder, which generated the 'fountain' of encoded symbols. Applying the LA-FEC approach to the Raptor specification [28], the precoding process and the LT-encoding process must be extended to all dependent media layers [29][30]. The required extensions use the algorithms for precode generation and LT Encoding as specified in [28], leaving the specification and the defined constraints of the algorithms untouched. Figure 19 illustrates the required extensions for the first and the second encoding step of a potential second media layer (Layer 1) which depends on Layer 0. Note that for the first layer (Layer 0) the encoding process remains unchanged.

To keep the systematic behaviour of the Raptor code for Layer 1, the precoding matrix must be extended to Layer 0 as shown in Figure 19 (left) (changes marked with green). The extension matrix $G_{LayeredLT}(1)$ is exactly the same as the extension matrix G_{LT0*} used for the LT-encoding process. With

Copyright © OCEAN Consortium, March 2011

the extension, the precoding symbols of Layer 1 (PSs_1) are modified in such a way, that the following LT-Encoding process generates a systematic code.

Finally, the LT-Encoding process is extended to the base layer as shown in Figure 19 (right). The extended LT-Encoding matrix introduces additional connections of the encoded symbols of Layer 1 to Layer 0, following the existing dependencies within the media stream. Therefore, the *ESs* of Layer 1 can now be used together with the *ESs* of Layer 0 for a combined decoding.

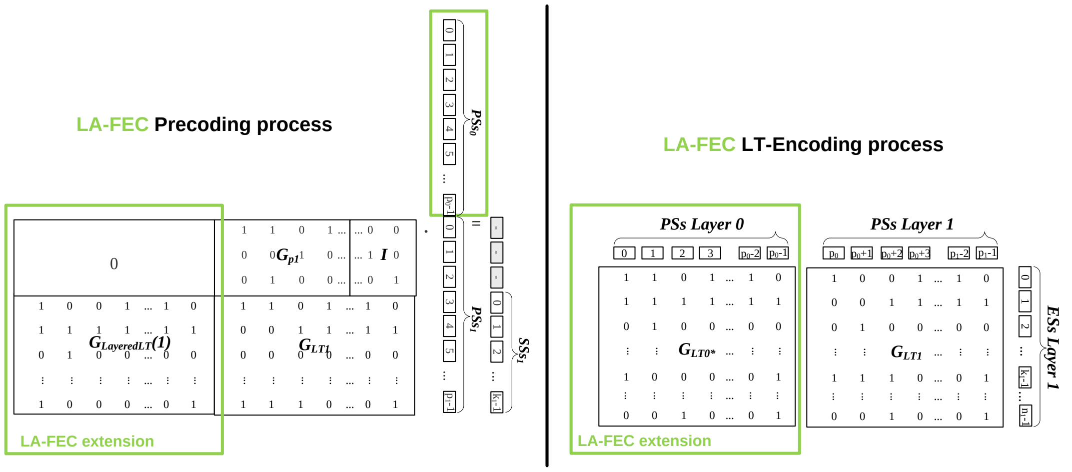


Figure 19: First step (precoding) of the Raptor encoding process with extended matrix for the second layer (left). The LA-FEC extension (marked in green) keeps the systematic behaviour of the layer-aware Raptor code. Second step (LT-Encoding) of the Raptor encoding process with extended LA-FEC matrix for the second layer (First layer see Figure 3). The LA-FEC extension additionally connects the encoded symbols (ESs) of Layer 1 to Layer 0, following the existing dependencies within the media stream.

3.2 Progress beyond state of the art

3.2.1 Signalling and Transport of Layer-Aware FEC

The usage of the LA-FEC in systems requires specialized signalling and adjusted transport techniques. The integration of the presented LDPC code on physical layer requires cross-layer interfaces, which signal media dependencies between specific LDPC blocks. Such cross layer interfaces would be anyway required for providing UEP, to signal dependencies between SVC layers.

The integration of the LA-FEC Raptor extension on application layer relies on real-time transmission and therefore on the relevant IETF standards for real-time delivery of media and FEC data. In real-time applications, typically RTP [31] is used over UDP [32] due to its connectionless and non-reliable nature it allows for minimal delay in transport. RTP provides basic features such as media synchronization, transmission order recovery, multiplexing, source identification and reception feedback information. For SVC, the RTP Payload Format for SVC [33] is required for media payload packetization. In particular, this payload format for SVC defines the transmission of the SVC data in multiple RTP sessions, which allows the transmission system and the LA-FEC coding process to simply differentiate between SVC layers based on the transport address, such as an IP address, the UDP port or the source identifier in the RTP packet header (SSRC) [31]. Signalling of session related information is defined in the Session Description Protocol [35][35] and the related payload information defined in [34]. In order to signal the dependency of RTP sessions containing layers of the same codec, the SDP extensions in [34] are required.

Copyright © OCEAN Consortium, March 2011

For transporting the FEC coded data, the IETF created the generic FECFRAME framework defining basic means for FEC based content delivery protocols, which can be also used with RTP. This framework defines beside other features how multiple media and repair flows are treated and provides an identification mechanism for source symbols as a part of the payload packetization information. To use this framework with the Raptor code, the technique of [36] is used.

In order to make this framework applicable to the LA-FEC the Raptor FEC scheme [37] and the Raptor RTP Payload format [38] can be used for packetizing the repair flow. The signalling for the Raptor FEC scheme is defined in [36], where the indication of depending repair flows is defined in [39], as required for the LA-FEC base layer protection and the LA-FEC enhancement layer protection.

3.2.2 Simulation results for delivery on the last mile

The simulated scenario is illustrated in Figure 20. A connection between a SENDER and RECEIVER is simulated. The connection is affected by losses simulated by a Gilbert Elliot Model (GE) and the available bit rate is varied from 0-1075 kbps. The SENDER provides a single scalable stream with two resolutions, qCIF and CIF. There is no means for any media coding or transcoding available at the SENDER. Moreover, the SENDER is aware of the available throughput and loss rates based on feedback received from the RECEIVER. The only way for the SENDER to adapt to the given channel conditions is to apply FEC to each layer. The further sections explain how the video coding stream, the packet losses, and the FEC redundancy are simulated.

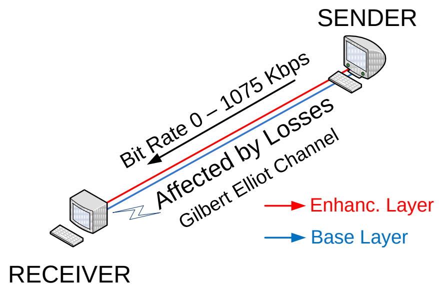


Figure 20: Simulation scenario. The sender transmits a SVC stream over channel affected by packet losses

3.2.2.1 Channel simulation

To simulate packet loss due to congestion we assumed the loss rates probabilities described in [40], which, once introduced into the Gilbert Elliot Model yields an IP packet error rate of 22 % with a mean burst length error average of 1.8 IP packets. In the performed simulations the generated IP packets are equal size to the MTU size, which means 1400 bytes. Occasionally some IP packets result with a smaller size due to fragmentation issues. The Gilbert Elliot diagram is based on a two state Markov-model as shown in Figure 21 State 0 represents the state of successful arrival of the packet, while state 1 represents the state of packet lost. The transition probability p_{10} from state 1 to state 0, the transition probability p_{01} from state 0 to state 1 as well as the IP packet error rate and the average burst length are summarized in Table II. For detailed calculations of the mentioned data, the interested reader is invited to resort to [40]. The simulations have been carried out increasing the bit rate available of the channel and choosing the optimal code rates for each point. Starting at bit rate 0 kbps, we increase the throughput of the link until the enhancement layer reaches an IP packet loss rate of 1%, which corresponds with a bit rate value of 1075 kbps.

Copyright © OCEAN Consortium, March 2011

Table 12: Parameters of Gilbert Elliot Channel Model

State Transition		***Channel Parameters***	
p_{10}	p_{01}	***IP packet error rate***	***Average burst length***
0.5479	0.0986	22%	1.8 IP packets

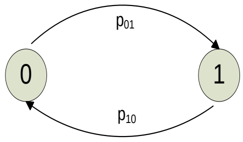


Figure 21: State diagram of the Gilbert Elliot model used for packet loss simulation

3.2.2.2 Simulation results

The performed simulations show the behavior of the base and enhancement layer depending on the bit rate available. Specifically, we analyze and plot for each layer the performance of the code rates applied (Figure 22), the IP packet loss rate experienced in the transmission (Figure 23) and the PSNR obtained in reception after the final decoding and correction steps (Figure 24). Due to the higher importance of the base layer within the SVC stream as explained before, the base layer is transmitted in first term. Moreover, an IP packet loss rate of 1% in the base layer is the target to reach before any attempt of transmitting the enhancement layer. When sufficient bit rate, the enhancement layer will be transmitted until the same packet loss rate is achieved.

Regarding Figure 22 and Figure 23:

- **Area A:** Between 0 and 164 Kbps, there is not enough bitrate available to transmit the base ayer, no transmission is possible.
- **Area B:** When a bit rate of 164 Kbps is available, the transmission of the base layer can be started. Firstly, the base layer is transmitted unprotected due to no remaining bit rate for redundancy (Point 1). From that point on, the more bit rate is available the more protection can be given to the base layer. Thus, the base layer code rate is reduced until a value of 0.52 where the target of 1% IP packet loss for the base layer is reached at bit rate 344 kbps (Point 2).
- **Area C:** We keep sending the base layer with 1% IP packet loss rate due to no bit rate available to incorporate the enhancement layer in the transmission.
- **Area D:** As soon as a bit rate of 724 kbps is ready for use, the Enhancement layer can be transmitted together with the base layer. As before, first no protection is applied to the enhancement layer (Point 3). Afterwards, gradually the code rate of the enhancement layer is reduced until 1% IP packet loss is reached. In case of using Standard FEC scheme protection, a bit rate of 1073 kbps is needed to fulfill the constraint of 1% IP packet loss in the enhancement layer (Point 5). On the other hand, when LA-FEC is used, the same constraint of packet loss is fulfilled at a bit rate of 969 kbps (Point 4), which means we manage to obtain a gain of more than 100 kbps.

Copyright © OCEAN Consortium, March 2011

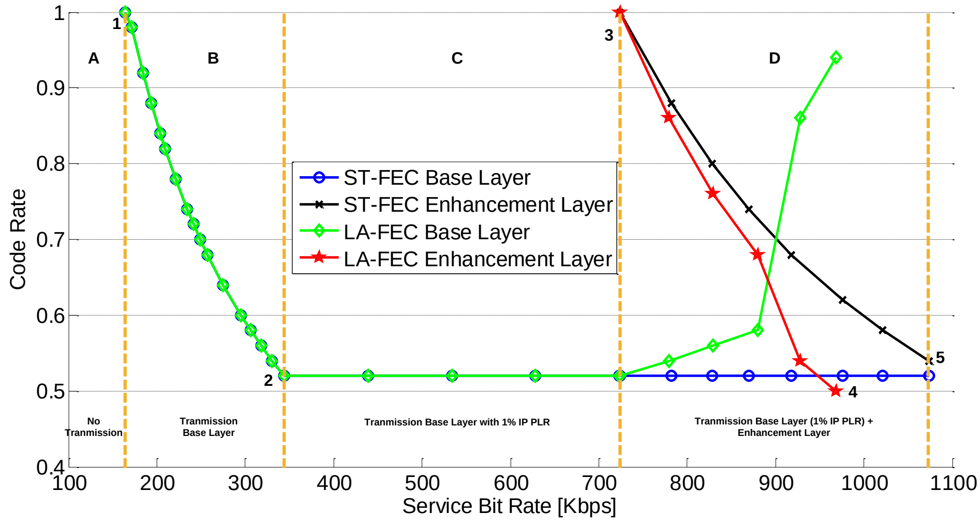


Figure 22: Code rates vs. Bit Rate available for Standard FEC (ST-FEC) and Layer-Aware FEC (LA-FEC)

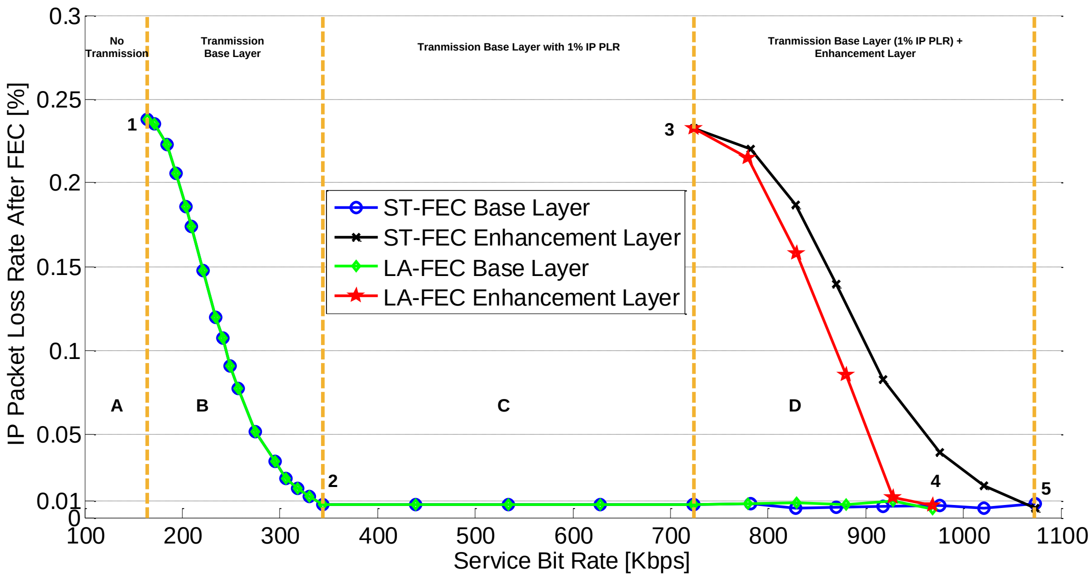


Figure 23: IP packet loss rate vs. Bit Rate available for Standard FEC (ST-FEC) and Layer-Aware FEC (LA-FEC)

Copyright © OCEAN Consortium, March 2011

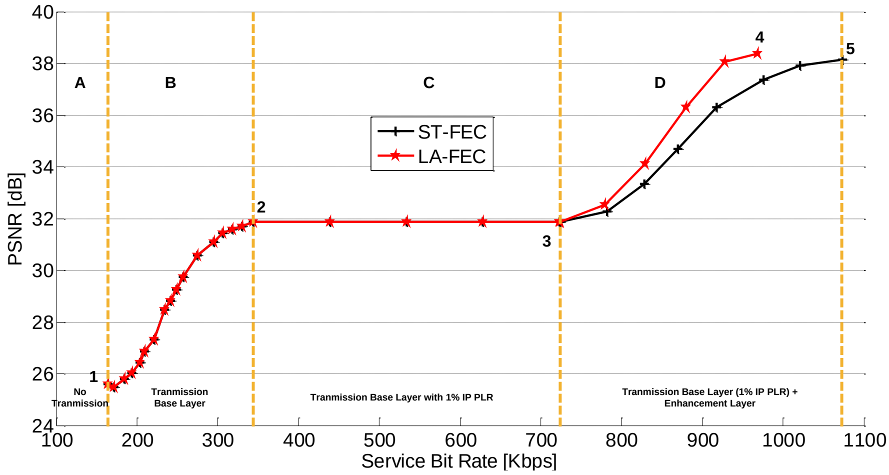


Figure 24: PSNR vs. Bit Rate available for Standard FEC (ST-FEC) and Layer-Aware FEC (LA-FEC)

Using ST-FEC, since there is no dependency within the layers, the code rate for the base layer must be kept constant for all service bit rates to keep the IP packet error rate at 1%. Using LA-FEC, the base layer protection can be reduced due to the increasing enhancement layer protection also protects the base layer. This translates into a PSNR gain on the final reception and into an earlier arrival at the IP packet loss rate limit of 1%.

Together with the previously described Figures, we also considered and analyzed the PSNR achieved at the receiver after the forward error correction step. Figure 24 shows the behavior related to the bit rate available on the channel:

- **Area A:** Not enough bit rate is available, so the base layer cannot be transmitted.
- **Area B:** When transmitting the base layer, the PSNR increases due to more bit rate is available, so more protection is given, achieving a maximum value of 31.87 at 344 kbps (Point 2).
- **Area C:** The transmission and the PSNR of the base layer keep constant during this area since there is not enough bit rate available to start transmitting the enhancement layer.
- **Area D:** When enhancement layer is transmitted along with the base layer, the difference between the two FEC protection schemes can be seen. While the ST-FEC simulation is reaching the highest PSNR at 1073 kbps (Point 5), the LA-FEC scheme is doing so at 969 kbps (Point 4). Once more, LA-FEC outperforms the ST- FEC technique in terms of PSNR. The gain is especially noticeable from 920 kbps on, where a difference between the two schemes of more than 1.5 dB is obtained at some points.

3.3 Conclusion

Layer-Aware FEC brings a significant gain for unreliable video transmission over UDP over error prone channels as shown in the presented results. While the base layer protection remains unchanged, the additional protection to the base layer increases the robustness of this more important layer.

Copyright © OCEAN Consortium, March 2011

4. CONCLUSION

We have developed and contributed to the required extension for MPEG DASH standard for supporting SVC within DASH. The required extension has been successfully contributed to the MPEG DASH standardization by OCEAN and is now part of the MPEG DASH standard. Thereby, the OCEAN media delivery with HTTP and SVC can be implemented in a standard conform way.

Within this contribution we further showed that SVC allows having multiple operation points within one stream. Either by encoding multiple SVC quality layers or by using MGS/CGS scalability and extracting multiple operation points out of one quality layer. The presented coding results show examples of the SVC coding gain and penalty for both setups, which influences the throughput on the transit link, within the CDN, and on the last mile. However, the throughput is not only influenced by the sheer coding overhead but also by the caching infrastructure and request statistics. This will be further investigated within Task T5.3.

Layer-Aware FEC (LA-FEC) shows gain for media delivery of scalable content over unreliable connections using RTP over UDP. However, due to the huge interest of the market and technical benefits of an HTTP streaming solution, which is based on TCP, HTTP streaming over TCP seems to be the most interesting delivery technology for the OCEAN system, on which will be the focus within the upcoming deliverables.

Copyright © OCEAN Consortium, March 2011

REFERENCES

[1] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An Analysis of Live Streaming Workloads on the Internet", In Proc. of the 4th ACM SIGCOMM conference on Internet measurement, 2004.
[2] R. Pantos, and W. May, "HTTP Live Streaming" Draft V04, IETF, June 5, 2010, http://tools.ietf.org/html/draft-pantos-http-live-streaming-04.
[3] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 9); 3GPP TS 26.234 V9.3.0 (2010-06), Section 12: Adaptive HTTP Streaming.
[4] Open IPTV Forum – Release 2 Specification, HTTP Adaptive Streaming, Draft V0.06 - June 7, 2010.
[5] ISO/IEC JTC 1/SC 29/WG 11 (MPEG), "Dynamic adaptive streaming over HTTP", w11578, CD 23001-6, Guangzhou, China, Oct 2010.
[6] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst.Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.
[7] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no.9, pp 1103–1120, 2007.
[8] T. Schierl, Y. Sanchez, R. Globisch, C. Hellge, and T. Wiegand, "Priority-based Media Delivery using SVC with RTP and HTTP streaming", Multimedia Tools and Application, 2010
[9] F. Hartanto, J. Kangasharju, M. Reisslein, and K. Ross, "Caching video objects: layers vs. versions?", Multimedia Tools and Applications, vol. 1, n. 2, 221-245, 2006.
[10] Y. Sánchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, Y. Le Louedec. "Improved caching for HTTP-based Video on Demand using Scalable Video Coding", Proceedings of the 8th Annual IEEE Consumer Communications and Networking Conference - Special Session IPTV and Multimedia CDN - (CCNC'2011 - SS IPTV)", Las Vegas (NV), January 9-12, 2011.
[11]H. Bahn, K. Koh, S. H. Noh, and S. Lyul Min, "Efficient Replacement of Nonuniform Objects in Web Caches", IEEE Computer magazine, vol. 35 no. 6, p.65-73, June 2002.
[12]S. Podlipnig, L. Böszörményi, "A survey of Web cache replacement strategies", ACM Computing Surveys, vol. 35 (2003), pp. 331-373, 2003.
[13]D. Hong, D. De Vleeschauwer, F. Baccelli, "A Chunk-based Caching Algorithm for Streaming Video", Proceedings of the 4th Workshop on Network Control and Optimization, Ghent (Belgium), November 29 – December 1, 2010.
[14]H. Schwarz and T. Wiegand, "Further results for an rd-optimized multi-loop SVC encoder", JVT-W071, JVT Meeting San Jose, USA, 2007, ftp://avguest@ftp3.itu.int/jvt-site/2007_04_SanJose/JVT-W071.zip.
[15]T. Stockhammer, M. Walter, G. Liebl, "Optimized H. 264-Based Bitstream Switching for Wireless Video Streaming. Multimedia and Expo," ICME 2005. Amsterdam, The Netherlands pp. 1396-1399, 2005.
[16]T. Schierl, T. Wiegand, M. Kampmann, "3GPP compliant adaptive wireless video streaming using H.264/AVC," IEEE International Conference Image Processing, ICIP 2005. vol. 3, pp. 696-699, 2005.
[17]N. Färber, S. Döhla, J. Issing, "Adaptive progressive download based on the MPEG-4 file format," Journal of Zhejiang University Science A, JZUS-A, Vol. 7, no. 1. pp. 164-111, 2006.
[18]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 TCSVT. vol. 17, no. 9, pp. 1227 - 1237, 2007.

Copyright © OCEAN Consortium, March 2011

[19]D. Renzi, P. Amon, S. Battista, "Video Content Adaptation Based on SVC and Associated RTP Packet Loss detection and Signaling," International Workshop on Image Analysis for Multimedia Interactive Services, 9th International Workshop, WIAMIS 2008. pp. 97-100,2008.
[20]T. Schierl, S. Johansen, C. Hellge, T. Stockhammer, T. Wiegand, "Distributed Rate-Distortion Optimization for Rateless Coded Scalable Video in Mobile Ad Hoc Networks," IEEE International Conference Image Processing, 2007. ICIP 2007. vol. 6, pp. 497-500, 2007.
[21] Information technology — Coding of audio-visual objects — Part 12: ISO base media file format, ISO/IEC 14496-12:2008(E)
[22]ITU-T Recommendation H.264 – ISO/IEC 14496-10 Advanced video coding (2009 Ed.).
[23]Open IPTV Forum – Release 2 Specification, HTTP Adaptive Streaming, DRAFT V0.06 - June 7, 2010
[24]ISO/IEC JTC1/SC29/WG11, Requirements on HTTP Streaming of MPEG Media, N11340, N1338, April 2010 + Updated Annexes of N1338, M17698, July 2010.
[25]Information technology — Coding of audio-visual objects — Part 15: Advanced Video Coding (AVC) file format, ISO/IEC 14496-15:2010(E)
[26]Information technology — Coding of audio-visual objects —Part 12: ISO base media file format ISO/IEC 14496-12:2008/FDAM2, Support for sub-track selection & switching, post-decoder requirements, and color information
[27]A. Shokrollahi, "Raptor Codes," IEEE Transactions on Information Theory, Vol. 52, Issue 6, pp. 2551-2567, 2006.
[28]IETF RFC 5053, “Raptor Forward Error Correction Scheme for Object Delivery”.ETSI TS 102 472 v1.2.1, “Digital Video Broadcast (DVB); IP Datacast over DVB-H: Content Delivery Protocols”.
[29]C. Hellge, T. Schierl, and T. Wiegand, "Mobile TV using scalable video voding and layer-aware forward error correction," IEEE International Conference on Multimedia and Expo (ICME'08), Hanover, Germany, June 2008.
[30]C. Hellge, T. Schierl, T. Wiegand, "Multidimensional Layered Forward Error Correction Using Rateless Codes," IEEE International conference on communication (ICC), Bejing, China, 2008.
[31]H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-time Applications", IETF STD 0064, RFC 3550, July 2003, http://tools.ietf.org/html/rfc3550
[32]J. Postel, "User Datagram Protocol", IETF STD 6, RFC 768, September 1981, http://tools.ietf.org/html/rfc768.
[33]S. Wenger, Y.-K. Wang, T. Schierl, and A. Eleftheriadis, "RTP payload format for SVC video", work in progress, Internet Engineering Task Force (IETF), Audio Video Transport Group (avt), draft-ietf-avt-rtp-svc-21, April 2010, http://tools.ietf.org/html/draft-ietf-avt-rtp-svc.
[34]T. Schierl and S. Wenger, "Signaling media decoding dependency in Session Description Protocol (SDP)", IETF RFC5583, Internet Engineering Task Force (IETF), Multiparty Multimedia Session Control (mmusic) , April 2009, http://tools.ietf.org/html/rfc5583.
[35]M. Handly, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol", IETF RFC 4566, July 2006, http://tools.ietf.org/html/rfc4566.
[36]M. Watson, "Forward Error Correction (FEC) Framework", IETF FECFRAME draft-ietf-fecframe-framework-09, July 2010, http://tools.ietf.org/html/draft-ietf-fecframe-framework.
[37]M. Watson, "Raptor FEC Schemes for FECFRAME", IETF FECFRAME, draft-ietf-fecframe-raptor-02, March 2010, http://tools.ietf.org/html/ draft-ietf-fecframe-raptor.
[38]M. Watson, "RTP Payload Format for Raptor FEC", IETF FECFRAME, draft-ietf-fecframe-rtp-raptor-03, March 2010, http://tools.ietf.org/html/ draft-ietf-fecframe-rtp-raptor.
[39]A. Begen, "SDP Elements for FEC Framework", IETF draft-ietf-fecframe-sdp-elements-07, July 2010, http://tools.ietf.org/html/draft-ietf-fecframe-sdp-elements-04.

Copyright © OCEAN Consortium, March 2011

[40]V. Markovski, F. Xue, and L. Trajkovic, “Simulation and analysis of packet loss in user datagram protocol transfers,” The Journal of Supercomputing, Vol. 20, No. 2, September 2001.

[41]T. Schierl, Y. Sanchez, K. Grüneberg, "On Adaptive HTTP Streaming using SVC and MVC," Response to CfP on HTTP Streaming of MPEG Media, MPEG#93, Geneva, 26-30 July 2010.

[42]C. Hellge, D. Gomez-Barquero, T. Schierl, T. Wiegand, "Intra-Burst Layer-Aware FEC for scalable video coding delivery in DVB-H," 2010 International Conference on Multimedia and Expo (ICME 2010), Singapore, Singapore, 19-23 July 2010.

[43]JSVM reference software, version 9.17, available via CVS from "garcon.ient.rwth-aachen.de:/cvs/jvt".

[44]H. Schwarz and T. Wiegand, "Further results for an rd-optimized multi-loop SVC encoder," JVT Meeting, April 2007.

[45]H. Schwarz and T. Wiegand, "R-D optimized multi-layer encoder control for SVC," IEEE ICIP 2007, San Antonio, TX, USA, September 2007.

[46]DVB tm-avc0417, H. Schwarz and T. Wiegand, "Experimental validation of the advantages of the SVC RAP definition in the current drafts for TS 101 154 and TS 102 005," 2008.

Copyright © OCEAN Consortium, March 2011

ACRONYMS

AHS	Adaptive HTTP Streaming
AVC	Advanced Video Coding
CABAC	Context-adaptive binary arithmetic coding
CGS	Coarse Grain scalability
CDN	Content Delivery Network
DASH	Dynamic Adaptive Streaming over HTTP
DVB	Digital Video Broadcasting
FEC	Forward Error Correction
GE	Gilbert-Elliot Model
GOP	Group Of Pictures
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
ISO	International Organization for Standardization
LA-FEC	Layer-Aware FEC
LDPC	Low-density Parity-check Code
LT-Code	Luby transform code
MGS	Medium Grain Scalability
MPEG	Moving Picture Experts Group
MTU	Maximum Transmission Unit
MVC	Multiview Video Coding
NAT	Network Address Translation
OCEAN	Open ContEnt Aware Networks
PCR	Programme Clock Reference
PID	Packet IDentifier
PSNR	Peak Signal-to-Noise Ratio
QP	Quantisation Parameter
RAP	Random Access Point
R-D	Rate-Distortion
RTP	Real-time Transport Protocol
ST-FEC	Standard FEC
SVC	Scalable Video Coding
TCP	Transmission Control Protocol

Copyright © OCEAN Consortium, March 2011

UDP	User Datagram Protocol
UEP	Unequal error protection
UMPC	Ultra-Mobile PC

Copyright © OCEAN Consortium, March 2011