DIstribution Of Multi-view Entertainment using content aware DElivery Systems

DIOMEDES

Grant Agreement Number: 247996

D5.2

Report on proof of concept prototypes (overall concept)

Document description	
Name of document	Report on proof of concept prototypes (overall concept)
Abstract	This report outlines the overall descriptions of final proof of concept prototypes main server and user terminal developed in WP5, along with the specific performance measurements for the included modules.
Document identifier	D5.2
Document class	Deliverable
Version	1.0
Author(s)	H.Gökmen, E.Doğan, S.Marangoz, O.Solakoğlu (ARC), S. Dogan, E. Ekmekcioglu, H. Kodikara Arachchi (UNIS), P. Kovács, K. Lackner (HOL), J. Hasselbach, T. Korn (IDMT), T.Adari (OPTEC), G.Gürler (KOC), N.Just (IRT)
QAT team	P. Kovács (HOL) T. Adari (OPTEC)
Date of creation	14 December 2011
Date of last modification	31 January 2012
Status	Final
Destination	European Commission
WP number	WP5

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Purpose of the document

The purpose of this document is to provide the overall descriptions of final proof of concept prototypes main server and user terminal developed in WP5, along with the specific performance measurements for all included modules.

1.2 Scope of the work

The scope of this document is to provide the descriptions of final proof of concept modules with their performance results.

1.3 Objectives

The objective of this document is to have a clear description and specific performance measurements of user terminal and server prototypes developed in WP5.

1.4 Structure of the document

After the introduction part, the proof-of-concept prototypes and their performance measurement results are described in Chapter 2 and Chapter 3 respectively. Finally, summary and conclusions are provided in Chapter 4. Please refer to "D5.3 Server, peer, and user terminal proof of concept prototypes" for the technical description of the software implementation of the proof of concept prototypes. These are explained in detail in that deliverable.

2 DESCRIPTION OF PROTOTYPES

The prototypes developed in WP5 include the User Terminal and the Server parts. The User Terminal side covers Control Module, Interaction Devices, AV Sync & Demux and Buffer, Audio and Video Decoders and Renderers, Adaptation Decision Engine, P2P Software, DVB-T reception and Authentication & Decryption. The Server side covers P2P Main Seed Server, TS Chunker, Access Control and Content Registration Server and 3D Content Server & DVB-T2 Transmitter. These are shortly explained below:

2.1 The User Terminal

The overall functional breakdown of the user terminal can be seen in Figure 1. Included modules' descriptions are explained between sections 2.1.1 and 2.1.9.

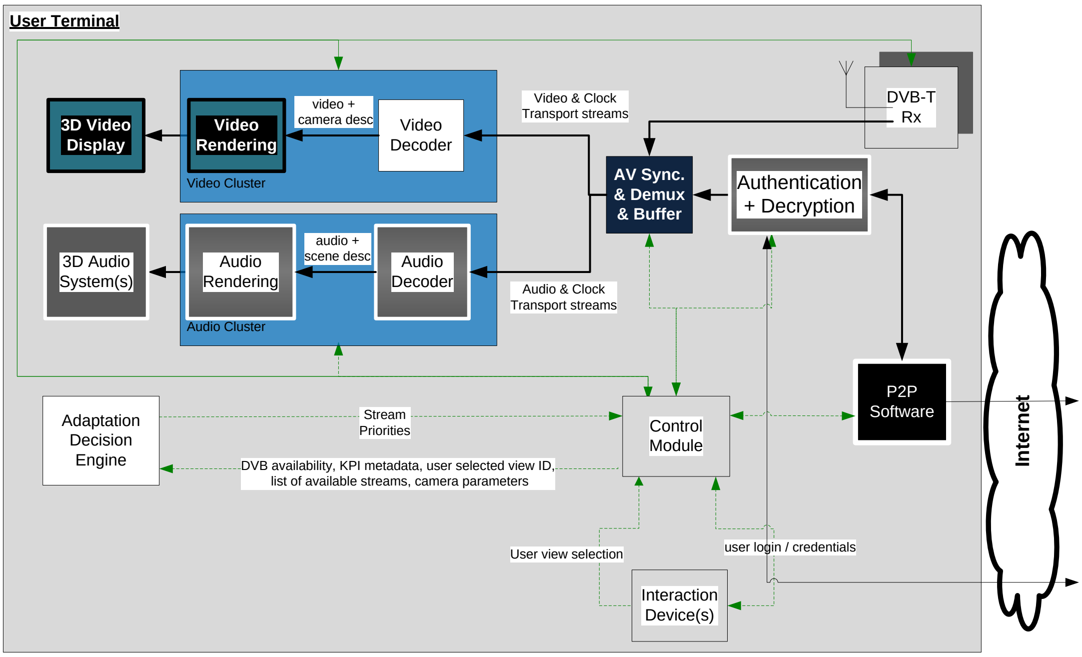

Figure 1 – User Terminal Overview

The user terminal and the server HW platforms are desktop systems with following features:

- Intel Core i7 Quad Processor
- min. 8 GB DDR3-1600,
- Solid State Disk,
- 2 Gigabit Ethernet Adapters
- running under 64bits Windows 7 OS (Audio and Video renderers will use dedicated clusters)

2.1.1 Control Module

Control module is the centralized module that is aware of all other system modules. It forms the central control mechanism of the user terminal functions. Thus it knows about all other modules and is able to control them by sending commands and receiving status information about the system.

2.1.2 Interaction Device(s) Module

Interaction devices at the User Terminal are used to manage user preferences (i.e., free-viewpoint selection) and to select content and to change module options. Based on user's preferences, available permissions/keys and program choices, corresponding content (e.g. 4-view video and multi-channel audio) will be delivered, processed and displayed. For free-viewpoint viewing feature, the interaction device facilitates a user interface. By using this interface, user is able to select a desired viewpoint (based on a reference camera coordinate system) in the units of the audio & video renderers can interpret.

2.1.3 Adaptation Decision Engine Module

Adaptation Decision Engine Module (ADEM) responds to context updates. When new Key Performance Indicator (KPI) metadata or specific user requested viewpoint information is received, it is considered as a context update. KPI metadata is embedded within the video streams, whereas user requests are obtained via the Interaction Devices of the User Terminal. The function of the ADEM is to produce a list of stream priorities by identifying the best combination of available viewpoints for synthesising a virtual viewpoint based on the received contextual information. To carry out this, it should be informed about the relative position of each camera and the coordinate system used. In this way, the best available viewpoints are prioritised over the others, so that the P2P Software Module can retrieve viewpoints that are essential for synthesising the required viewpoint when the available network resources are limited. In cases of bandwidth restrictions, the stream priority list generated by the ADEM comes particularly handy for identifying which views as well as stream layers should or should not be received for an acceptable level of user experience. Thus, prioritisation of scalable video layers other than viewpoints, such as quality enhancement layers, is also a part of the ADEM operation. Audio streams delivered through P2P connection (i.e., audio objects) are given the highest priority over the video related streams at all times, by default. This is based on the assumption that their bit-rate is considerably lower compared to video streams (hence, no useful bit-rate adaptation range is gained by discarding them before video streams) and they have significant influence on the perceived audio-visual 3D scene.

2.1.4 P2P Software Module

The P2P software module is responsible for receiving the content from other peers and the main seed server in an adaptive way. Moreover, it is also responsible for handling data requests from other peers and can forward video chunks. When it starts, the P2P module is in slave mode, waiting command from the Control Module.

The user initiates 3D content request through the “Control Module” which notifies the “P2P module” at the requesting peer. Then, the “P2P module” at the requesting peer contacts the 3D content server and requests a metadata file. The metadata file includes information about the 3D content and the key information that is requested by the security client is downloaded prior to initialization of the adaptive multi-view streaming.

Once the initialization period is over the Adaptation Decision Engine Module (ADEM, see 2.1.3) sends the prioritization list of streams which indicates their importance level. Based on this information the P2P module initializes the downloading process and forwards data to the AV Sync & Demux & Buffer Module (AV-Sync) via Authentication and Decryption Module. There are two different cases for P2P video streaming. In the first case, the play out is in synchronization with DVB. Therefore, the AV-Sync module frequently updates the P2P module about the PCR clock of the DVB channel. Based on this information the P2P module jumps to the correct chunk and maintains synchronized video streaming. In the second case, the AV-Sync module only expects data from the P2P module. In this case the play out should start when there is data in the buffer.

2.1.5 DVB-T Receiving Module

The DVB Receiving Module uses a DVB-T demodulator card to receive the modulated DVB-T signal and sends the transport stream via UDP/IP to the AV-Sync module. The transport stream is not modified and also its timing is preserved. This enables a DVB conformant transmission to the AV-Sync module.

Please refer to D5.5 for a detailed description of the DVB components and the DVB-T/T2.

2.1.6 Authentication + Decryption Module (Security Client)

The Security Client decrypts P2P chunks received from the P2P Client and validates the integrity and authenticity of the chunks. In case of validation failure, e.g. due to transmission error or manipulation, it notifies the P2P Client to recover data and/or to prevent further sharing of corrupted data. Finally it sends the decrypted and validated MPEG2-TS packets to the AV-Sync Module.

The main functionalities of this module are:

- to authenticate data coming from the P2P Software Module and to provide feedback in case of failed authentication
- to decrypt data coming from the P2P Client
- to split the decrypted P2P chunks into MPEG2-TS packets, suitable for further processing
- to manage related key information (for both content decryption and authentication)

For more detailed information including APIs, workflows and the implemented broadcast encryption scheme, please refer to D5.4 “Security Components Prototypes”.

2.1.7 AV Sync. & Demux & Buffer Module

The AV Sync. & Demux & Buffer module is responsible for:

- Demultiplexing the transport streams arriving from DVB-T and P2P paths to separate buffers of video and audio.
- Buffering the audio streams and the video-layer streams (layers form the video-viewpoints).
- Holding several seconds of data (transport packets) before streaming it to the relevant cluster.
- Synchronising the audio and the video, the streams of the video views, the streams arriving from DVB-T and P2P.
- Sending the synchronised streams to the Video cluster & Audio cluster some frames

before the indicated presentation time in order to allow the decoders enough time to decode and wait till it is time to send the frame for rendering.

2.1.8 Audio Cluster Module

The audio cluster module receives the audio transport streams from the AVSync&Demux&Buffer module and (viewpoint) adaptation control data from the control module. The transport stream is decoded and used for synchronisation to the transmitted timestamps. The audio rendering modules drive the loudspeaker system used for reproduction of the programme's audio scenes.

2.1.9 Video Cluster Module

The Video Cluster consists of four main components: Decoder Control, ES Demux, Video Decoder(s), and Video Renderer(s). These modules together handle the decoding of the incoming video streams, and visualize them on the user's display. Apart from the video streams, this module receives control signals from outside the video cluster, to support features like audio-video sync, adaptation decisions, viewpoint changes, to name a few.

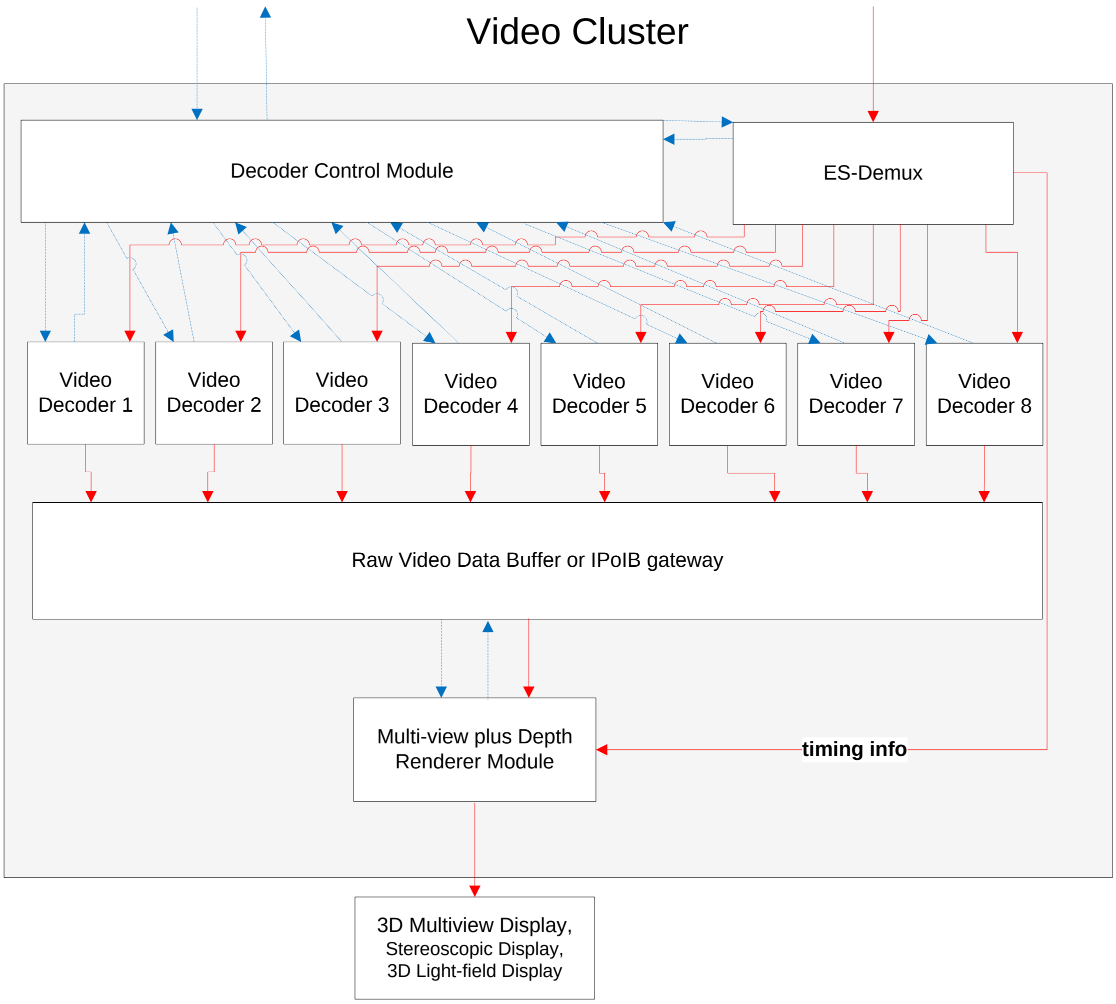

Figure 2 – Video Cluster

2.1.9.1 Decoder Control

Decoder control module is the central communication module within the video cluster. It is in charge of initiating the transport stream demultiplexing, decoding and rendering processes. Its operation is based on the set of commands it receives from the Control Module, which includes the metadata that is related to the multimedia content, including all PID-stream-layer associations. Based on the received set of metadata, decoder control module initiates the stream demultiplexer with corresponding input parameters, the Decoder(s) with input and output port associations and the Renderer with the required rendering mode, etc. It directs the user preferred viewing point message coming from the Control Module to the Renderer and also directs the outgoing messages to the Control Module (e.g. KPI feedback messages generated in the Decoders).

2.1.9.2 ES Demux

ES Demux is responsible of extracting the elementary streams (i.e., the encoded video in MPEG-2 Annex B format) from the received multiplex of MPEG-2 Transport Stream. The Transport Stream multiplex containing video layer streams (including depth map) is received from the AV-Sync module over one UDP connection and parsed in real time. The aggregated Network Abstraction Layer (NAL) units of each camera viewpoint (or depth map) is then forwarded to the input of the respective decoder block over an output UDP port. While doing so, ES Demux combines the base layer and enhancement layer NAL units with the same presentation time stamp and send them to the decoder in a single packet. This is required by the decoder, since the decoding process happens access unit by access unit. ES Demux also parses and sends the Program Clock Rate (PCR) information received within the multiplexed Transport Stream to the Renderer module to regulate synchronised playback.

2.1.9.3 Video Decoder

This module receives the elementary bit-stream for multi-view colour and multi-view depth videos from the ES demultiplexing buffer and generates raw image frames. The video decoder is designed to decode one viewpoint (either colour video or depth map video) in a single process. The output frames are forwarded to the Video Renderer using either shared memory (when both module is on the same PC) or TCP connection. In addition to the task of decoding, the video decoder module is responsible for extracting the KPI information that is embedded into the elementary stream and forwarding the content to the decoder control module.

The video decoder software unit is built on top of the OpenSVC decoder software library, which is an optimised version of the SVC reference software developed by JVT for real-time decoding (available under LGPL license in: http://sourceforge.net/projects/opensvcdecoder).

2.1.9.4 Video Renderer

The video renderer is responsible for transforming the uncompressed images + depth map pairs into 3D views, converting these to the input format of the attached 3D display, and present them in sync with the audio cluster. It supports a wide range of 3D display formats. During normal operation, it synthesizes virtual views according to the requirements of the display and the viewing position adjusted by the user in real-time. In the simplest case, it accepts stereoscopic 3D video and presents it in the desired stereoscopic format. The video renderer is tightly connected to the video decoder. The huge amount of uncompressed video data can be transmitted via shared memory, Ethernet or Infiniband connections from the decoder(s) to the renderer(s).

2.2 The Server

The server covers Access Control & Content Registration Server, P2P Main Seed Server, TS Chunker, 3D Content Server and DVB-T Transmission modules.

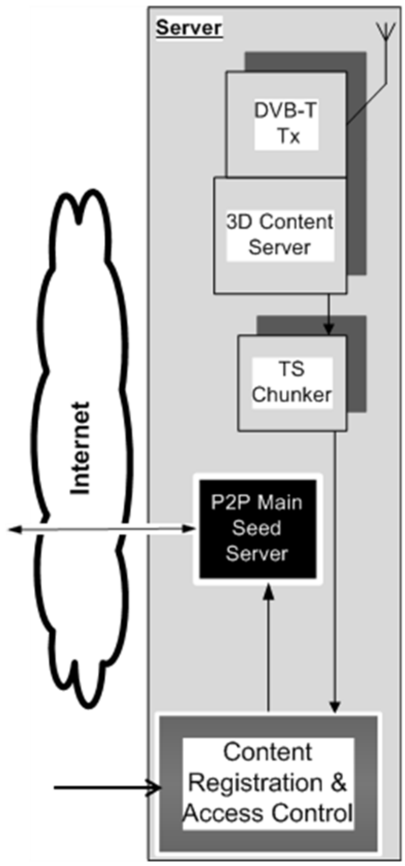

Figure 3 - The Server Side

2.2.1 P2P Main Seed Server

The main seed server is responsible for 5 key tasks which are listed below:

1. The first and most important role of the main seed server is to initiate the distribution of the 3D content to a limited number of peers that reside inside the swarm.
2. Main Seed Server is to behave as a tracker server for each session and help a new coming peer to find other peers in the swarm.
3. Responsible to store the metadata chunk that is to be forwarded to peers in order to initialize the receiving system.
4. Stores the key information that is used for content authentication at the receiver side.
5. Performs sanity checks on the content that is being uploaded. Tests are performed in order to accept content. The list of checks that are performed can be found in D2.3

Final Reference Architecture.

2.2.2 Access Control and Content Registration Server (Security Server)

The main functionalities of the Security Server are:

- Access control - in order to avoid unauthorized access to content
- Content registration and authentication - in order to prevent sharing of malware and unauthorized content
- Content discovery – in order to allow users to discover and select content to be retrieved via P2P

The Security Server provides a central entry point for new content. It encrypts and digitally signs the transport stream chunks provided by the 3D Content Server, before forwarding them to the P2P Main Seed Server. Moreover, this component is responsible for the management of keys and receivers - including key generation, storage and efficient key distribution using the available broadcast channels. The Security Server generates so called P2P chunks consisting of a P2P chunk header and the actual payload. The payload can contain encrypted and signed A/V data, signed content metadata, or signed key information generated by the Security Server. For more detailed information including APIs, workflows and the implemented broadcast encryption scheme, please refer to D5.4 “Security Components Prototypes”.

2.2.3 3D Content Server & DVB-T Transmission

The following components are used for the creation of the transport streams transmitted to the DVB and P2P path:

A software multiplexer is used to create the streams for the two delivery channels from single program transport streams carrying the audio and video components for the broadcasted service.

The TsChunker module is used to split the transport stream for P2P distribution in chunks according to defined rules.

Since the software multiplexer also supports modulator output it is also used for the DVB-Transmission. The modulator is configured through this module and also the transport stream bitrate is adapted to match the required bitrate according to the modulation settings.

Please refer to D5.5 for a detailed description of the DVB components and DVB-T/T2.

3 PERFORMANCE MEASUREMENT RESULTS OF PROTOTYPES

The performance measurement results for the user terminal and the server components are summarized below. It is shown that the performance of included prototypes should be sufficient for the DIOMEDES demonstrator, following the integration work in WP6.

3.1 The User Terminal

The user terminal is measured in terms of the rendered audio-visual quality, which is being assessed using the QoE metric developed in Task 3.2.

3.1.1 Adaptation Decision Engine Module and P2P Software Module

A number of streaming tests are performed on a controlled LAN environment to obtain the initial performance results for the P2P Software module that works in conjunction with the Adaptation Decision Engine Module. A network emulator library on Linux OS is utilised for this purpose. The results are extensively depicted and an analysis is provided in WP4 deliverable D4.5 (Report on results of integrated MD-SMVD and P2P system). Readers can refer to this deliverable for more details. A summary of the results in several test scenarios is provided here.

The adaptation performance under dynamic network conditions is evaluated both in terms of the quality of service (QoS) and the quality of user's experience (QoE). From the perspective of the QoS, the following observations are done:

- The prioritisation order sent out from the Adaptation Decision Engine Module clearly influences the relative percentages of received chunks from each stream (e.g. the most important stream arrives with over 99% completeness even under the network condition, where the total download capacity is merely equal to one of the stream's bit-rate)
- The critical duration is the few seconds after the streaming-session start, where the system tends to treat all streams equally. Unless 2-3 seconds of a buffer duration exists, even chunk losses from the most important stream are observed within the first five seconds (up to 25% of total chunks within the first 10 seconds)
- If the network conditions are severe, some streams are downloaded at significantly low chunk reception percentage (e.g. the least important stream). Depending on the user's QoE, the P2P streaming client should be able to decide whether to completely discarding a stream or to continue partially receiving it.

From the perspective of the Quality of Experience (QoE), the following observations are done:

- In parallel with the QoS observations, the perceived 3D video quality of the stream tagged as the most important by Adaptation Decision Engine Module is consistently higher over time.
- Even when the total available bandwidth is consistently equal to merely one half of the required source data rate, the most important stream's quality is around 0.8 3DQ (i.e., good broadcast quality) for 65% – 90% of the streaming time, while the quality of the other streams is sacrificed.

These preliminary results suggest that the envisaged adaptation features will be enough to fulfil the demonstration requirements.

The scalability performance of the P2P system itself is tested by increasing the total number of media consuming peers while observing the total burden on the main seed server(s). The followings are observed:

- The deployed system explicitly augments the streaming over the peers, rather than via the seed servers. The total percentage of chunks delivered via other peers (by each peer) increases by 30% (from ~40% to ~70%), if the number of peers in a system is increased from 2 to 10.
- The deployed system ensures that the load on the main seed server(s) is not linearly increasing with increasing the number of peers. In the tests, when the total number of peers in a system changes from 2 to 10, the load on the server increases only 2.5 times.

- In the experiments, the average link capacity per peer is changed in between 3.5 Mbps and 6.5 Mbps. Increasing link capacity of each peer has decreased the average bit-rate requirements on main seed servers by 8%-15%.

The performance of the applied adaptive multiple-descriptions streaming is observed in an environment with three peers with dynamic uplink capacities, where the base layer chunks are streamed (a) fully redundantly, (b) selective redundantly and (c) with no redundancy. It is seen that over a number of trials, the average correct download rate of base layer chunks in selective redundant scheme is 1.65 times of the average correct download rate in fully redundant scheme, and ~1.2 times of the average correct download rate in no redundancy scheme. Hence, selective redundancy (or multiple description streaming) has proved to be the most suitable technique for the demos.

3.1.2 Authentication + Decryption Module (Security Client)

The main requirement regarding the Security Client (and the security components in general) is to provide an appropriate level of security while achieving a good performance regarding processing time. In order to measure the performance the following setup has been used:

- Intel Core 2 Duo CPU (T7500@2,2 GHz), 2 GB Ram, Windows 7 (64 bit)
- Content: ~200 MB size, ~6500 chunks having a size between 20 kb and 70 kb per chunk, real time data rate of the source content: ~9,6Mb/s

Using this (low-end) setup, the achieved data rates for both chunk decryption and authentication are significantly higher than the required real time data rate of the source content. For more detailed information, please refer to D5.4 “Security Components Prototypes”. It can be assumed, that the usage of the actual prototype setup (Intel Core i7, 8 GB RAM) will achieve considerable higher performance (data rates).

3.1.3 AV Sync. & Demux & Buffer Module

Test environment:

- Maximum of 4 video streams (AVC format) of video views displaying time-codes.
- Source length: 15 minutes.
- One stream from DVB-T and 3 from P2P.
- VLC is used as the decoder.

Test result:

- Difference was measured using snapshot of the VLC screens. Sampling every minute.
- The maximal difference between the views was of 2 frames. Usually it was 0-1 frames.

Subjective tests did not indicate A/V sync problems.

3.1.4 Audio Cluster Module

The DIOMEDES audio cluster implementation was used in a series of perceptual experiments. The performance was successfully tested during the iterations of module implementation and in conjunction with the perception tests.

Audio transmission performance related to Quality of Experience

Deliverable D3.4 gives a detailed description of the perceptual test results and the bitrate / angular deviation ranges that have to be chosen for transmission to achieve excellent perceptual impressions.

Rendering performance tests

The following section is mainly based on the Deliverable D3.5/D3.6 section on audio rendering functionality evaluation.

The functionalities of the audio cluster were tested on different PC platforms in various loudspeaker constellations to validate the module's performance within the DIOMEDES specification. Since the computational need of the audio cluster depends on the number of the attached loudspeakers, there are no fixed conditions to conduct general performance measurements. Instead of measuring one general test case, the core audio rendering modules were tested at Fraunhofer IDMT's various audio reproduction systems, rendering an audio scene of 32 simultaneous sources each (live signal input):

- WFS reproduction on a setup of 88 loudspeakers and 4 subwoofers
- Low resolution 2D reproduction on a setup of 23 loudspeakers and 4 subwoofers
- 3D low resolution reproduction on a setup of 52 loudspeakers and 4 subwoofers
- 3D low resolution reproduction on a setup of 60 loudspeakers and 4 subwoofers

In all situations, the audio processing was running on Linux x86 quad core PCs (2.8GHz). The rendering behaviour was stable for all tests (no audio drop-outs were observed, CPU usage varies depending on the test configuration and is individually distributed on 4 CPU cores in the IDMT tests).

Performance of audio rendering with network decoding

The audio rendering-only configurations were extended by the DIOMEDES network audio scene decoder module. This scenario represents the use of the audio decoding and rendering modules within the DIOMEDES terminal demonstrator. The decoder module simultaneously performs audio decompression of 32 audio channels of the example audio scenes. The audio scenes consisted of DIOMEDES demo scenes, streams from DIOMEDES AV QoE experiments and additional audio scenes that were transferred into DIOMEDES stream format. Depending on the content of the scenes, subsets of the 32 coded channels can contain muted audio objects on currently unused channels (e.g. the DIOMEDES demonstrator audio scenes only occupy subsets of the 32 available simultaneous audio objects). Unused audio channels don't affect the rendering performance. The combinations of decoding and rendering were tested on the following setups:

- Low resolution 2D reproduction on a setup of 8 loudspeakers (demonstrator during DIOMEDES integration meetings)
- Low resolution 2D reproduction on a setup of 23 loudspeakers and 4 subwoofers
- 3D low resolution reproduction on a setup of 52 loudspeakers and 4 subwoofers
- 3D low resolution reproduction on a setup of 60 loudspeakers and 4 subwoofers

The audio rendering configurations with the spatial audio decoder were stable for all cases (no audio drop-outs were observed, CPU usage varies depending on the test configuration and is individually distributed on 4 CPU cores in the IDMT tests).

Performance of 2 rendering and 2 decoding instances on one PC

For the audio visual quality perception test described in D3.4, the audio rendering and decoding structures were doubled on one PC. An object based audio scene streaming configuration was created that allows the rendering of a scene on a high resolution loudspeaker setup when streamed to a first stream decoder, and the rendering on a subset of the loudspeaker setup when streamed to the second network decoder module running on the

PC. In addition to that, 2 different sets of 32 equalisation filters were applied simultaneously to the object signals before rendering (compensation of the different loudspeaker densities for both rendering options, using processing modules for fast FIR convolution).

Two rendering instances were configured to deliver the audio signals for different subsets of the connected 88 loudspeaker setup from 32 simultaneous audio objects:

- WFS reproduction on a setup of 88 loudspeakers (complete setup) and 4 subwoofers
- Low resolution 2D reproduction on a setup of 22 loudspeakers (subset of the 88 loudspeakers) and 4 subwoofers

The signal outputs of these rendering instances were added to be output to the same loudspeaker system.

Two spatial audio scene streaming decoders were receiving and decoding the incoming audio scene streams. The 32 output signals of the decoders were fed to the audio renderers via 2 arrays of 32 FIR convolvers each for frequency response equalization.

Only one decoder module received an audio scene stream at a time.

The system was fed with multiplexed stereo HD video and object based audio scene streams within a series of the 40min AV-perception tests. The audio rendering was reliable throughout the tests series (no audio drop-outs were observed).

Performance of audio timing / AV synchronisation

The synchronisation of audio rendering with an external video playback was controlled preparing different MPEG-2 TS example streams. Each stream contained object based audio scenes, mono or stereo video streams, and in some cases channel based audio streams to validate channel based format decoding. The following list shows some of the content examples that were used for the test streams. Since the DIOMEDES audio recordings and productions are not very complex, Fraunhofer IDMT used additional AV content for testing:

- DIOMEDES “Music”: HD Stereo Video, object+channel based audio, about 20sec
- DIOMEDES “Lecture”: HD Stereo Video, object based audio, about 20sec
- DIOMEDES “Fencing”: HD Video, object+channel based audio, about 15sec
- DIOMEDES AV Sync Test video: HD video, object+channel based audio, >5min
- DIOMEDES “T-Rex”: HD video, object+channel based audio, >5min
- Trailer “The Settlers” PC game trailer: PAL Video, object based audio, >3min
- Trailer “Creating Waves”: PAL Video (CGI), object based audio, >1.5min
- DVB-S example TV program streams (ASTRA satellite): PAL or HD Video, channel based audio
- DVD example streams: PAL video, channel based audio, streamed from DVD Video using VLC player software

All streams were sent from a dedicated streaming PC to both audio rendering PC and a video playback PC (using VLC player as video decoder and player). Streaming was done using DIOMEDES tools from IRT and OPTIBASE, a dedicated DIOMEDES TS-streaming tool from Fraunhofer IDMT, VLC Player software and embedded streaming software of a DVB-S receiver.

All test streams showed very good synchronization properties. Audio rendering was stable even over long test periods (e.g. AV perception tests using long series of DIOMEDES lecture and music sequences).

The tests of AV synchronisation between DIOMEDES audio cluster and DIOMEDES video cluster still have to be completed.

Decoding performance when changing stream conditions

The spatial audio / MPEG-2 TS decoder module is designed to handle changing stream conditions. The decoder module was tested to validate the following behaviour:

- The highest priority (please refer to D3.5) stream is chosen for decoding.
- If TS reception is completely interrupted, decoding is stopped. No audio output.
- It the currently decoded audio stream is interrupted (e.g. if DIOMEDES P2P transmission is interrupted due to packet loss) but alternative streams are still received, an alternative stream of lower priority is chosen. If no valid TS packets are received, decoding is stopped (no audio output).

The automatic stream switching performance was successfully tested during demonstrator implementation and integration. The described cases are handled properly.

3.1.5 Video Cluster Module

The performance of the Video Cluster Module is, in most cases, determined by the bandwidth of the transmission channel between the video decoder(s) and the video renderer(s), carrying uncompressed image and depth data.

The video decoder and video renderer have been tested with 2 views and 2 depth maps. When parallel TCP sessions (views) were tested over single Infiniband connection between the decoder and the renderer, views can be transmitted above 25fps. For HD Ready resolution, 4 views can be transmitted, while for Full HD resolution 2 views can be transmitted reliably.

3.1.5.1 Video Decoder

We use OpenSVCDecoder for performing the decoding task in real-time. Even with an optimized decoder, a high performance PC is required to deliver the task of decoding in real-time. **Error! Reference source not found.** provides the performance of video decoding in units of decoded frames per second using Intel i7 processor at 3.4 Ghz with 4 real-cores.

Error! Reference source not found. provides the results for decoding colour videos that can comprise up to two layers. Since the depth maps are encoded at a comparably lower bitrate, e.g. the sizes of the decodable NAL units are much smaller, and since the skip macroblock modes are deployed more, decoding speed for depth maps is significantly high. It should be also noted that depth maps comprise only a single layer.

	1 View		2 View		4 View	
	Base	Base+Enh.	Base	Base+Enh.	Base	Base+Enh.
Music	71	39	59	30	59	24

Table 1 - Decoder Performance of Video Cluster for Colour Views (fps)

3.1.5.2 Video Renderer

The performance of the video renderer in isolation can be easily measured if the renderer is used with shared memory input, which results in input data always available immediately. As the renderer offloads most steps of the view generation task to the GPU, the performance when measured in isolation is affected by the type of the GPU available in the system. During these measurements, the incoming data from shared memory is uploaded to the GPU before rendering each frame (even if the data did not change) in order to measure real-time performance. This is important because uploading content from main memory to GPU memory is still a time sensitive step, even when done asynchronously.

The reported performance values have been measured on a 4-core, 2,3 GHz desktop computer, equipped with 4 GB RAM and an NVIDIA Geforce GTX 280 GPU. When converting and presenting 1248*688 pixel resolution stereoscopic content on a 1280*1024 pixel sized stereoscopic display, the refresh rate is 110 FPS. Synthesizing 2 views (1280*1024 each) from 2 images and 2 depth maps runs with 464 FPS, while synthesizing 8 views (1280*1024 each) from 6 images and 6 depth maps runs with 13 FPS.

From these results we can see that generation of free-viewpoint stereoscopic output by far surpasses the 25 FPS requirement, while multi-view generation is also performs close to real-time. A GPU upgrade can easily make multi-view generation a real-time process too.

Renderer performance / number of incoming image-depth pairs / number of generated views	2 I+D	4 I+D	6 I+D
1 views	69	46	39
2 views	46	35	30
3 views	35	29	25
4 views	29	24	22
5 views	23	21	18
6 views	20	18	17
7 views	18	16	15
8 views	16	15	13

Table 2 - Video Renderer Performance

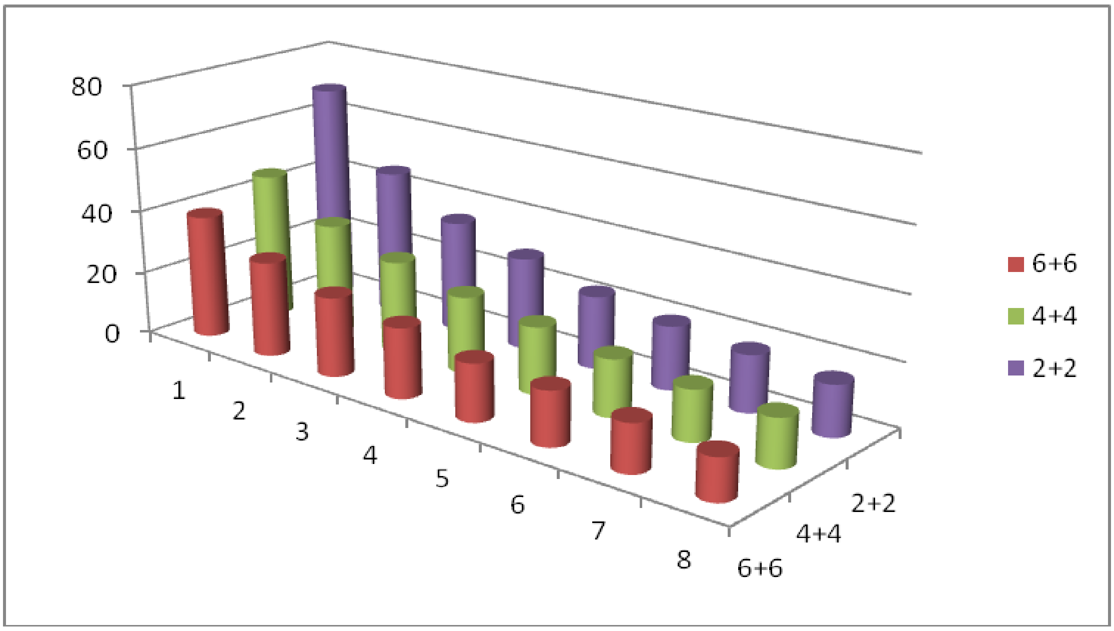

Figure 4 - Performance of video renderer for generating 1-9 views, when using 6, 4, and 2 image+depth pairs as input, respectively

3.2 The Server

The performance measurement results of the following server components P2P Main Seed Server, Access Control and Content Registration Server and 3D Content Server and DVB-T Transmission are summarized below:

3.2.1 P2P Main Seed Server

The main seed server can perform all of its tasks that are enlisted in Section 2.2.1 in real-time using any modern PC (e.g., Intel Core-i3).

3.2.2 Access Control and Content Registration Server (Security Server)

The main requirement regarding the Security Server (and the security components in general) is to provide an appropriate level of security while achieving a good performance regarding processing time. In order to measure the performance the following setup has been used:

- Intel Core 2 Duo CPU (T7500@2,2 GHz), 2 GB Ram, Windows 7 (64 bit)
- Content: ~200 MB size, ~6500 chunks having a size between 20 kb and 70 kb per chunk, real time data rate of the source content: ~9,6 Mb/s. Using this (low-end) setup for the chunk generation (including encryption and signing), processing at content source real time data rate could be achieved. For more detailed information, please refer to D5.4 “Security Components Prototypes”. It can be assumed, that the usage of the actual prototype setup (Intel Core i7, 8 GB RAM) will achieve considerably higher performance (data rates).

3.2.3 3D Content Server & DVB-T Transmission

All modules relevant for the DVB transmission are running on a desktop-PC (Intel Core i7, 8GB RAM, Windows 7 64bit). Transport streams with up to 31 Mbps (Maximum bitrate for DVB-T) were tested successfully.

Please refer to D5.5 for a detailed description of the DVB components and DVB-T/T2.

4 SUMMARY AND CONCLUSIONS

This document has provided the overall descriptions of the proof of concept prototypes in the user terminal and in the server, together with their specific performance measurements.

We have seen that the developed proof of concept prototypes in the user terminal and the server side can perform all of their tasks on the PC's used in the demonstration setup.

Thus, we can conclude that the performance of the developed proof of concept prototypes which are summarized in Chapter 3 should be sufficient to fulfil the requirements of the DIOMEDES demonstrator.

APPENDIX A: GLOSSARY OF ABBREVIATIONS

A	
ADEM	Adaptation Decision Engine Module
AVC	Advanced Video Coding
API	Application Programmer Interface
C	
CPU	Central Processing Unit
D	
DDR	Double Data Rate
DVB	Digital Video Broadcasting
DVB-C/C2	Digital Video Broadcasting – cable
DVB-S/S2	Digital Video Broadcasting – satellite
DVB-T/T2	Digital Video Broadcasting – terrestrial
DVD	Digital Versatile/Video Disc
E	
ETSI	European Telecommunications Standards Institute
F	
FEC	Forward Error Correction
FEF	Future Extension Frames
FFT	Fast Fourier Transformation
Fps	Frames per second
G	
GPU	Graphics Processing Unit
GUID	Globally Unique Identifier
H	
HP Stream	High Priority Stream
I	
i	Interlaced scan
J	
JSON	JavaScript Object Notation
K	
KPI	Key Performance Indicator
L	
LDPC	Low Density Parity Check
LGPL	Lesser General Public Licence
LP Stream	Low Priority Stream
M	
MD-SMVD	Multiple Description - Scalable Multi-view Video plus Depth
MPEG	Motion Picture Experts Group

MPEG2-TS	MPEG2-Transport Stream
N	
NAL	Network Abstraction Layer
O	
OS	Operating System
P	
p	Progressive scan
PAL	Phase Alternating Line
PC	Personal Computer
P2P	Peer to Peer
PCR	Program Clock Reference
Q	
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
QoS	Quality of Service
R	
RAM	Random Access Memory
S	
SI	Service Information
SVC	Scalable Video Coding
U	
UDP/IP	User Datagram Protocol/Internet Protocol
V	
VLC	Video LAN Client
VQM	Video Quality Metric
W	
WFS	Wave Field Synthesis