Project Number: 248495
Project acronym: OptiBand
Project title: Optimization of Bandwidth for IPTV Video Streaming

Deliverable reference number: D3.4.1
Deliverable title: IPTV data dropping algorithm prototype Version 1 and associated documentation

Due date of deliverable: M19
Actual submission date: 31-10-2011 (M22)

Start date of project: 1 January 2010   Duration: 30 months

Organisation name of lead contractor for this deliverable: Corrigent Systems Ltd (CSL)

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Table of figures

Table of Tables

Glossary

Abbreviation / acronym	Description
ADSL	Asymmetric Digital Subscriber Line
AVC	Advanced Video Coding
CBR	Constant Bit Rate
CC	Continuity Check
CLI	Command Line Interface
DSL	Digital Subscriber Line
DTS	Decode Time Stamp
IDR	I-frame known as an instantaneous decoder refresh (IDR) frame
LUT	Lookup table with MOS values for some content types
Metro Network	A metropolitan area network (MAN) is a network that usually spans a city. A MAN usually interconnects a number of local area networks (LANs) using a high-capacity backbone technology, and provides uplink services to wide area networks (or WAN) and the Internet.
MGS	Medium Grained Scalability
NAL	Network Abstraction Layer
OP	Operation Point is defined as a unique combination of temporal and quality levels
PAT	Program Association Table
PCR	Program Clock Reference
PDA	Packet Dropping Algorithm
PDD	Packet Dropping Device
PID	Program ID
PMT	Program Map Table
PSI	Program Specific Information, metadata about a program (channel) and part of a MPEG transport stream, the PSI data contains five tables: PAT (Program Association Table) CAT (Conditional Access Table) PMT (Program Map Table) NIT (Network Information Table) TDT (Time and Date Table)
PTS	Presentation Time Stamp
RTP	Real Time Protocol
STB	Set Top Box
SVC	Scalable Video Coding
VBR	Variable Bit Rate
YUV	Colour space presentation

Executive Summary

OptiBand provides a solution for Telecom operators to offer IPTV services to any user over any type of access technologies. The major advantage of using OptiBand solution is the ability to provide HD video streams to users with limited access bandwidth, such as DSL users.

The solution is based on packet dropping algorithm (PDA) that is located at the edge of the metro network and the access network. The PDA monitors users' requests to conclude which channel each user is watching. It knows the user access link rate by configuration and it intelligently discards packets in order to adapt the video stream rate to the user access rate.

At this stage of the project, two algorithms of different technologies are evaluated. These algorithms perform intelligent packet dropping in order to achieve the desired video rate at the maximum possible quality. WP3 deals with the Network side solution.

Method 1: AVC Multi bitrate streams

In this method, multiple streams per channel are created at the head-end by a H.264/AVC encoder and are encapsulated in a MPEG-TS container, where each stream has a different bit-rate. The streams belonging to the same channel are synchronized to allow the packet dropping device (PDD) to seamlessly switch from one stream to another. The PDD outputs one stream per channel, dynamically switching between the streams based on the conditions of the ADSL link and the combination of competing channels over same shared ADSL link.

Method 2: SVC with MGS scalability

In this method, a single stream per channel is created at the head-end by an SVC encoder using MGS SNR scalability [1][4]. The stream is encapsulated in MPEG-2 TS packets and, subsequently, in a set of RTP packets. The method allows dropping data that carries SNR (i.e. quality) enhancements to a picture. This method refers to *quality_id* as quality level, which indicates the MGS layer a NAL unit belongs to, and to *temporal_id* as temporal level, which indicates importance of a NAL unit in the temporal domain, being a NAL unit with temporal level i dependent on NAL units of with temporal level $0...i$-1. Based on the quality levels and temporal levels, this method defines operation points as a set of NAL units with a quality level and a temporal level lower than some given values predefined for the operation point. Each RTP packet carries a metadata in SSRC field of the header that enable the packet dropping algorithm to associate the packet data with (an) operation point(s). The bit-rate reduction is achieved by dropping packets that contain data outside of the chosen operation point.

This document describes the results of Task 3.4 "Build and test prototypes" of the OptiBand project. This deliverable includes a preliminary version of the Packet Dropping device prototypes at the network side.

1. Introduction

D3.4.1 – Data Dropping Algorithm Prototype (ver1) describes a preliminary prototype implementing the required data drop algorithm at the transport network. The prototype will include both hardware and software, and will be used to test each of the proposed packet dropping algorithms as a network solution for the OptiBand project.

D3.4.1 specifies two approaches for packet dropping: [1] AVC multi bitrate streams and [2] SVC with MGS scalability. For each approach the document describes the prototype architecture, the interfaces and interaction with other IPTV elements, such as the head-end, middleware and STB. It specifies the prototype functionality on data-plane (video streams handling), control-plane (protocols and signalling handling), and management plane (configuration and monitoring).

The final version of the Data Dropping Algorithm Prototype will be performed and reported in separate documents (D.3.4.2 due M23).

1.1 Reference documents

The following documents have been used as reference documents:

[1] D1.1 Functional specification document

[2] D3.2 IPTV data dropping algorithm description

[3] D3.3.1 IPTV data dropping algorithm simulator Version 1 and report

[4] D3.3.2 IPTV data dropping algorithm simulator Version 2 and report

[5] D4.1 Head end video processing description

[6] D4.3 Metadata objects description

[7] D4.4.1 Head end simulation Version 1 and report

[8] D4.4.2 Head end simulation Version 2 and report

[9] D4.5.1 Head end prototype Version 1 and associated documentation

[10] D5.2 System Architecture & Functional Definitions – Interim version

2. Packet Dropping Algorithm Prototypes

This section provides a detailed description of the 1st version Packet Dropping Algorithm (PDA) prototype. In this prototype, the algorithm is implemented and integrated in a Packet Dropping Device (PDD) that is located at the metro aggregation area of the transport network.

The prototype is designed to operate in a real time environment, it able to receive video streams that were originated by the head-end device and perform the dropping decisions based on real-time indications from the IPTV eco-system and from the transport network management system (NMS).

The 1st version of the PDA prototype is challenged to achieve 16% bandwidth reduction of HD video channels with minimal QoE degradation.

This section specifies two algorithms that OptiBand evaluates:

- AVC multi bitrate video streams
- SVC with MGS scalability

2.1 AVC Multi Bitrate Streams Prototype [CSL]

2.1.1 General Description

The multi bitrate Packet Dropping Algorithm (PDA) prototype is implemented at a transport network edge device. The algorithm uses several H.264/AVC video streams generated by the head-end and distributed over the IPTV network up to the IPTV network edge device that performs the dropping algorithm, as illustrated in the following figure:
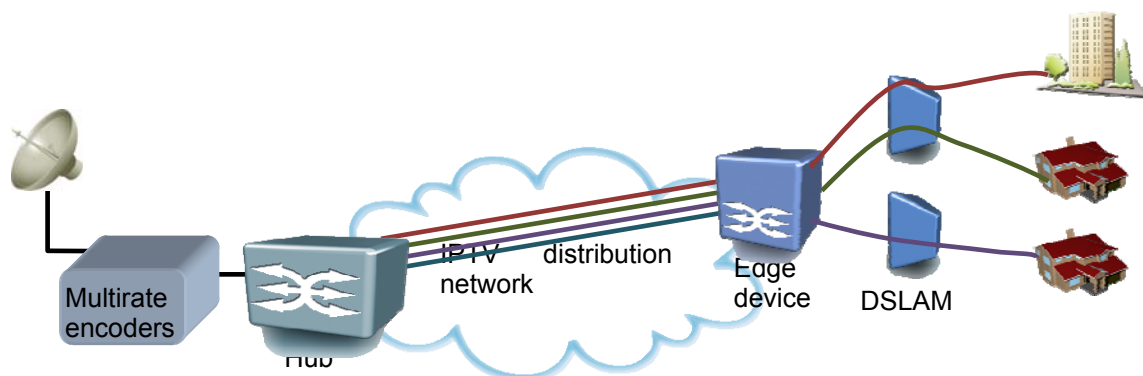
Figure 1 - IPTV network with Multi AVC streams

Each of the distributed streams has different bitrates quality and hence different bitrate, the edge device that informed with the user's DSL link rate in advance chooses the best video stream that fits into the specific user DSL link by forwarding the video stream with the highest bitrate that still complies with the DSL link rate.

Although previous simulations, which have been made and reported at D3.3.1 and D3.3.2, used VBR (Variable Bitrate) it was agreed between OptiBand's partners to use only CBR (Constant Bitrate). Main reason for that is that VBR is mainly used for Broadcast (Satellite & Terrestrial) as it is coupled with statistical multiplexing to improve the bandwidth usage, while for broadband

(IPTV), there is not such an advantage to use VBR. In order to comply with the best practice and to comply with all existing equipment it was suggested to use CBR mode only.

A user may have several active STBs that share the same DSL access link bandwidth. In order to handle multiple channels over limited DSL link the PDA has to perform intelligent decisions to select the best video streams per each active STB with the goal to provide the maximum quality and ensure their sum rate doesn't violate the DSL allowed rate.

The PDA has the flexibility to switch on a real-time basis between the available multi-streams per each channel to adapt its decision to the dynamic changes within each channel content, meaning, when the algorithm identifies that one of the channels starts a new scene that requires more bandwidth to maintain certain quality level, it may dynamically allocate more bandwidth to this demanding channel on the expense of the other channels that compete on the user DSL link budget.

It should be noted that the PDA performs decision per DSL link, another user with different conditions, such as DSL rate or different combinations of channels will experience a different PDA decisions. The PDA personalizes the video channels to each OptiBand based IPTV user.

The PDA device integrates two functionalities that are addressed by the prototype implementation:

- Network edge device – The prototype is a transport network element that aggregates multiple DSLAM access devices. As a transport network element, the prototype has interfaces with the metro/aggregation network and with the access networks, it handles all network traffic in addition to the IPTV network (such as data and VOIP traffic), and it is managed by the Network Management System (NMS).

- IPTV PDA – the prototype is part of the IPTV service, it provides the OptiBand benefits to existing IPTV service model. As IPTV element, the prototype interacts with the IPTV ecosystem including the Middleware, the Head-end and the STBs.

The following subsections describe the prototype architecture and its interactions with both Network elements and IPTV elements.

2.1.2 Prototype Architecture

The following figure describes the packet dropping algorithm modules at the prototype device:
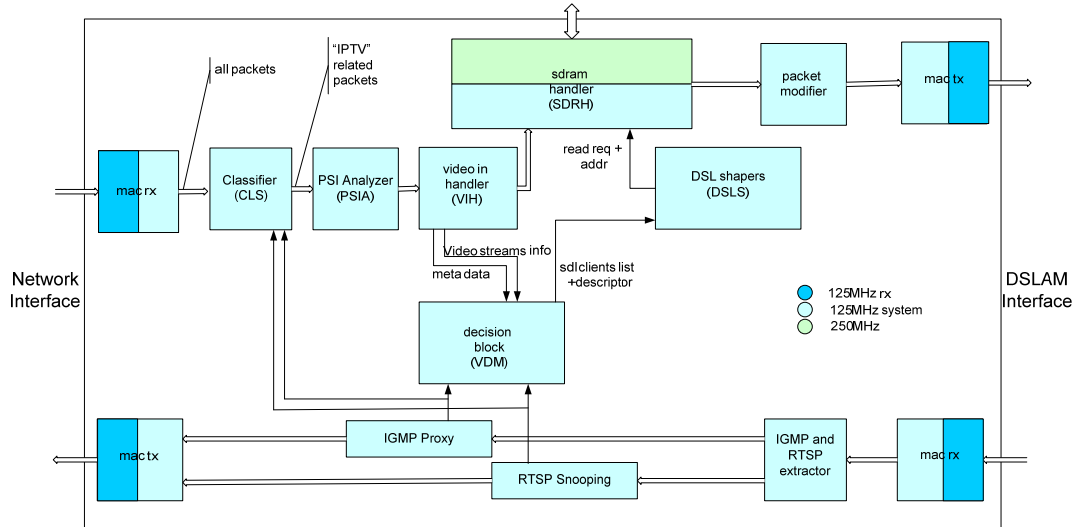
Figure 2 - PDA Top level block diagram

This block diagram illustrates the data flow on both downstream and upstream directions, between the head-end to the STB. As previously described, the prototype is located at the metro edge, it receives the multi bitrate streams from the networks side, performs dropping decisions, and forwards the adapted video streams towards the DSLAM access devices.

mac_rx and mac_tx modules

These modules are implementing the data link layer that provides the layer2 connectivity and addressing which is needed to connect the prototype device as a network element to the IPTV network.

The mac_rx modules receive the multi bitrate video stream packets, which were originally generated by the head-end, from the metro network.

The mac_tx modules transmit the adapted video stream packets to the destined DSLAM.

IGMP Proxy module

This module detects IGMP requests that are generated by the user's STBs when the user zaps between broadcast channels.

When a user zaps from Channel-A to Channel-B, the STB generates IGMP-leave request for IP multicast address of Channel-A, and generates IGMP-join request for IP multicast address of Channel-B.

The "proxy" functionality filters IGMP packets in order to reduce load on the multicast router. Join and leave requests heading upstream to the router are filtered so that only the minimal quantity of information is sent. The PDD is trying to ensure the router only has a single entry for the group, regardless of how many active listeners there are. If there are two active listeners in a group and the first one leaves, then the PDD determines that the router does not need this information since it does not affect the status of the group from the router's point of view. However the next time there is a routine query from the router the PDD will forward the reply from the remaining host, to prevent the router from believing there are no active listeners.

The IGMP proxy flow chart is illustrated in Figure 3:
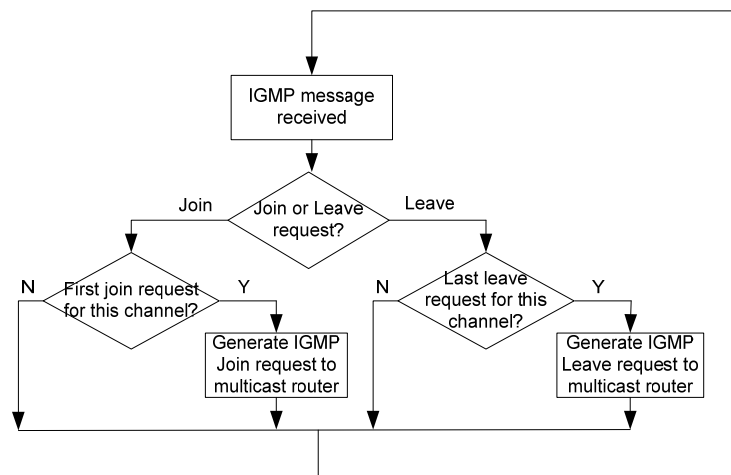
Figure 3 - IGMP Proxy functionality flow chart

RTSP Snooping module

This module detects RTSP sessions between the STB and the VoD server. When a user starts or terminates a session with the VoD server, the STB uses RTSP protocol that can be identified by the PDD element.

The "snooping" functionality only listens to the RTSP in order to identify that the STB is watching VoD channel and the algorithm needs to take this channel into account when performing bandwidth allocation to STBs that share the same ADSL link.

Since VoD application is not part of the integration or trial phase the RTSP will not be implemented for the prototype phase.

classifier (CLS) module

This module is the entry module that receives IP packets from the mac_rx module and identifies whether the packet is an "IPTV" packet or a packet of another service.

Moreover, for IPTV packets, the CLS module identifies whether this packet belongs to broadcast TV (BCTV) of video on demand (VoD) channel.

The identification of "IPTV" packets and the channel type (BCTV or VoD) is done by monitoring the IP source and destinations addresses.

BCTV channels have a dedicated, pre-assigned, multicast IP address. The list of multicast IP addresses of BCTV channels may be maintained at the PDD by one of the following alternatives:

- The MiddleWare configures the PDD which each IPTV BCTV channels in the same way as it configures the STBs with the list of channels and their multicast IP addresses.

- The PDD that also functions as IGMP proxy element at the network, monitors the IGMP requests generates by the user's STB and builds the multicast channels list accordingly, when at least one user is watching a specific BCTV channel, the PDD adds this channel multicast IP address to the CLS channels list. When the last user that watched a channel zaps to another channel, the PDD removes this BCTV channel entry from the CLS list.

The BCTV and VoD channels classification is done as specified in Table 1:

Table 1 - BCTV and VoD channels identification

Channel type	Source IP	Destination IP	Description
BCTV	Any value	Multicast address	Each BCTV channel has specific pre-assigned multicast IP address. For the 1st phase of the prototype, this information is provided to the PDD by the middleware
VoD	VoD server	STB address	VoD content is sourced by VoD server with well known IP address. VoD streams are unicast and are directly streamed to the requesting STB. For the 1st phase of the prototype, the VoD server IP address and the STB IP address are provided to the PDD by the middleware

Packets that were identified by the CLS as "IPTV" packets, are sent to "PSI_Analyzer" (PSIA) module for further handling, all other packets (the non IPTV packets) are transparently bypassed to the users.

PSI_Analyzer (PSIA) module

The PSIA received only IPTV packets that were identified by the preceding CLS module. The PSIA identifies MPEG-TS packets that carry PSI tables and analyzes them to extract information, as specified in Table 2:

Table 2 - PSI tables processed by PSIA module

MPEG-TS type	Description
PAT	PAT tables have constant PID=0x0000 PSIA identifies the PAT table based on PID (all 0s) PSIA analyzes the PAT table to extract the PID of the PMT
PMT	PSIA identifies the PMT based on the PID value indicated at the PAT PSIA analyzes the PMT table to extract the PID of all multi bitrate video streams

Note that the PSIA is using the MPEG-TS headers for packets identifications, where MPEG-TS headers are not encrypted (only payload is encrypted).

video_in_handler (VIH) module

The VIH module receives IPTV MPEG-TS packets, such as video streams of all stream rates, audio, subtitles, meta data etc of BCTV and VoD channels. It identifies video MPEG-TS packets based on their PID values that were extracted from the PMT table by the PSIA module, and extracts information from the video streams as specified in Table 3.

Table 3 - VIH extracted information

Extracted info	Identified based on	Description
Video stream index	PID value	Each of the multi bitrate video stream has different PID value
GOP start	Indication at the MPEG-TS header	The 1st MPEG-TS packet of an IDR frame is marked to indicate a beginning of a new GOP

The VIH module forwards the IPTV MPEG-TS packets to the SDRAM handler module (SDRH) to be buffered at the external memory. For each MPEG-TS packet, the VIH module builds a descriptor that specifies the location of the buffered MPEG-TS at the SDRAM memory. The descriptor is sent to the VDM for further handling.

The VIH module can also be configured to identify Null MPEG-TS packets (null packets have constant PID=0x1FFF), and to drop them to reduce the bandwidth consumption over the ADSL link.

video_decision module (VDM)

The VDM has three main roles:

Implements the packet dropping algorithm and hence, decides for each STB which of the multi bitrate video stream to select, and drops all the MPEG-TS packets of the non-selected multi bitrate streams that were not selected by the algorithm.

- Maintains the database of the channels that are distributed at the IPTV network. The information maintained at the database is specified in Table 4
- Maintains the database of the clients (ADSL/STB). The information maintained at the database is specified in Table 5

Table 4 - VDM database per Channel

Information	Provided by	Description
Channel Index	CLS module	Index that the PDD assigns to each channel

Channel Priority	Middleware	Value between 1…8 for indicating channel priority among the channels competing on a single ADSL line.
MPEG-TS video	PSIA module	Indicates MPEG-TS packet type (video, or other)
Video stream index	VIH module	Index that the PDD assigns to each video stream within the channel
GOP start pointer	VIH module	Points to the MPEG-TS (in the memory) that carries the beginning of the IDR frame that starts a GOP
GOP size	VIH module	A value that specifies the GOP size, which can be used to the stream rate of the starting GOP
Video stream rate	Middleware	Indicates the bitrate of each video stream within the multi bitrate channel
Video channel quality	Middleware	Indicates the quality of the channel using LUT

Table 5 - VDM database per ADSL subscriber

Information	Provided by	Description
ADSL Rate	NMS	The ADSL rate allocated for IPTV service
Number of STBs	IGMP Proxy module RTSP snooping module	Number of active (turned on) STBs that share the same ADSL link
STB[i] channel index	IGMP Proxy module RTSP snooping module	The channel index that each STB currently displays
STB[i] MAC address	Middleware	The MAC address of the destined STB
STB[i].genre[j].priority	Middleware	Genre priority per STB at an household

[*] [i] is the STB index (0.. Number of STBs)

Whenever a video block descriptor is received from the VIH module, the entire "per ADSL" database is scanned to find which STB should receive the block. The descriptor is sent to every DSL shaper that a STB behind it watches the movie (if 2 STB behind a DSL watch the movie, 2 copies of the descriptor are sent to the DSL shaper module.

The VDM that implements the dropping algorithm performs decision calculation at the beginning of each GOP, this allows the algorithm to switch between the different stream rates on IDR boundaries to avoid error propagation when switching from low quality stream to higher quality stream. Switching between streams on frame boundary for example, may result in a selection of IDR frame from low quality stream that will be used as poor reference for following frames within the same GOP that may be selected from higher quality stream.

The algorithm uses the "per Channel" database to extract information per channel, specifically, it uses the GOP size information (provided by the VIH module) and the video streams quality information (as indicated in Table 4), for making dropping decisions based on 2 criteria:

- The streams quality information differentiate between the multi bitrate streams of same channel, the algorithm will have a sense of a quality level of each stream rate, which has strict correlation with the user QoE, the algorithm will take this indication into consideration when some of the bitrate streams have poor quality and are not suitable to satisfy user QoE

- The algorithm equalizes as much as possible the qualities of streams that intended to different STBs that shares the same ADSL line. In some cases, one channel may require more bandwidth to achieve fair QoE while the other channel at the 2nd STB may use a lower rate stream to achieve same quality level as the channel of the 1st STB. Equalizing the ADSL rate between the active STBs doesn't always lead to equalization of qualities.

This quality indication can't be measured or estimated by the PDA mainly due to the fact that the streams are encrypted when transported over the network, and such information is not accessible to the algorithm.

The algorithm shall take into consideration the remainder of the ADSL link rate after allocating it to streams with discrete bitrates. The remainder shall be allocated among channels based on the channel priority.

In case where a channel is being played more than once (i.e. for several STBs), each play is treated as a different channel from the algorithms point of view.

The algorithm calculates the Complexity value of each channel per GOP:

Complexity(channel) =

$(\sum (Rate(stream[i]) / channel\_Quality) / NumStreams(channel)) * (STB.genre.priority)$

Where:

Complexity (channel) is the normalized channel complexity

[i] is the stream index within the multi bitrate channel

Rate() is the measured rate of stream[i] at the current GOP

channel_Quality is the quality level of the channel (based on LUT)

NumStreams(channel) is the number of multi bitrate streams that the channel offers over the IPTV network

STB.genre.priority is the priority of a genre per the playing STB (a number between 1-1.5)

High Complexity indicates that the channel demand more bandwidth to maintain quality.

In general, the allowed rate for each channel is the channel's complexity relatively to all other channels complexity (that competes on same ADSL link) multiplied by the ADSL line rate, by applying:

$STB\_allowedRate = ADSL\_rate * Complexity(channel) / \sum Complexity(all\ channels)$

Where:

Complexity channel) is the calculated Complexity value of the channel that STB[i] displays

$\sum$Complexity (*all channels*) is the sum of Complexities of channels that are transported over the ADSL link

More bandwidth demanding channels, which have higher complexity, gets more bitrate.

Next, for each channel the PDA selects the stream with the highest rate that still below the channel allowed rate. Note that since for each channel there are only few discrete rates the implementation should take this into account and making sure leftover bandwidth is minimized based on channel priority:

For i=0 $\rightarrow$ num_of_channels

$STB\_allowedRate = ADSL\_rate * Complexity(i) / \sum Complexity(i \text{ to num\_of\_channels})$

Allocate $Max(Rate(stream[j])) \leq STB\_allowedRate$

For i=highest priority channel → lowest priority channel

Increase allowed rate as much as possible within ADSL rate remainder

* For CBR with constant quality assessment per stream, the algorithms can be executed for each configuration change or each user behaviour change (zapping, leaving or joining a channel) rather than at each GOP.

* The remainder handler won't be part of the 1st version of the prototype.

* For the 1st prototype version quality is considered as same value among all streams and channels.

DSL Shapers (DSLS) module

The DSLS module implements a rate shaper for each ADSL line. The rate shaper can be configured in steps of 10Kbit/sec.

The DSLS receives from the VDM module a valid signal per ADSL line, whenever a new MPEG-TS packet received and is waiting at the SDRAM memory for transmission.

DSLS module is described in Figure 4:
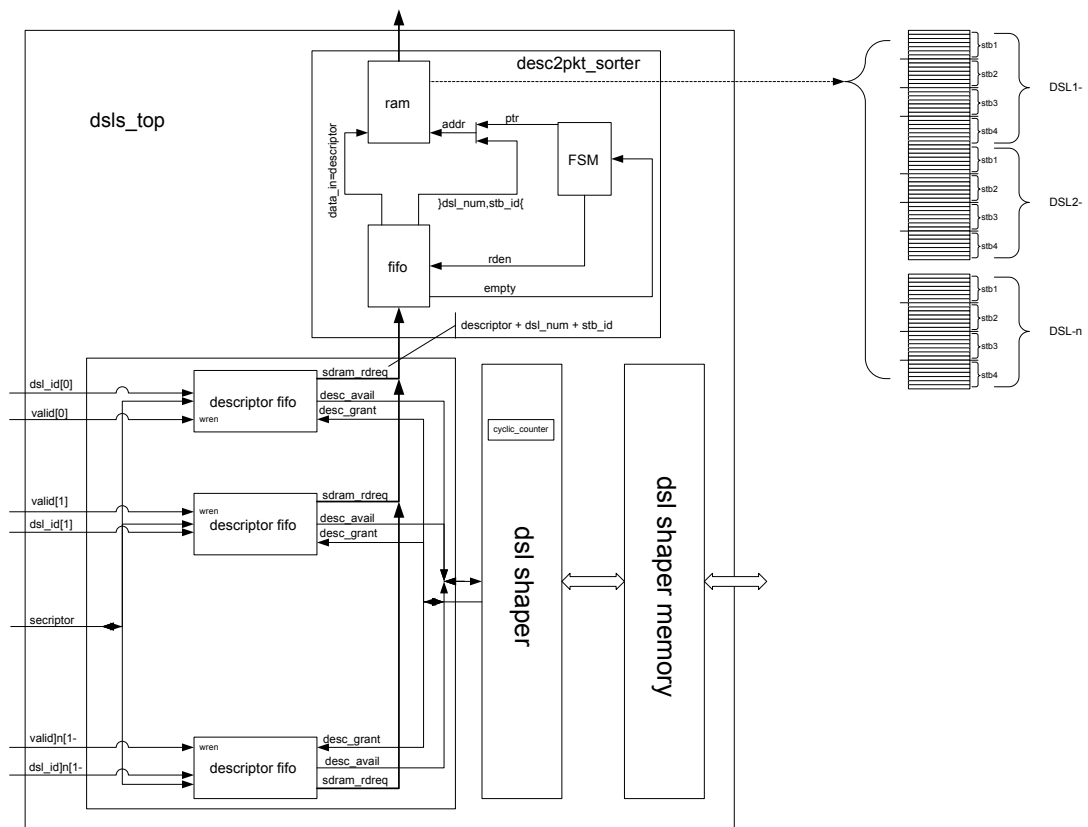
Figure 4 - DSLS module description

For each ADSL line, the DSLS maintains 2 registers:

- CreditVal
- CreditCounter

CreditVal indicates how many credits the ADSL line is deserve for transmission every credit interval, while CreditCounter indicates how many credits the ADSL line accumulated.

For example, for ADSL line of 10Mbps, and credit interval of 12.5usec:

$$CreditVal = 10Mbps * 12.5usec = 125 bits$$

If the credits were not exploited by the ADSL line for MPEG-TS packets transmission for 1msec for example, then the ADSL line accumulated credits to transmit:

$$CreditCounter = 125bits * (1msec / 12.5usec) = 10000 bits$$

The ADSL rate shaper allows a transmission of IPTV packet (that carries 7 MPEG-TS packets) only when the ADSL CreditCounter accumulated credits to transmit at least:

*8[bits/byte] * 188[bytes/MPEG-TS] * 7[MPEG-TS/IP-Packet] = 10528 bits*

Packet Modifier Module (PMM)

The PMM is the module at the output of the PDD, it modifies the MPEG-TS header of video MPEG-TS packets and the Ethernet headers of the transmitted IP packet per destination STB, as follows:

The PDA decides which of the video stream (among the available multi bitrate streams distributed at the network) to select and forward per STB per ADSL. In some cases, the PDA may change its decisions and switch between video streams within the same channel. For example, when one STB zaps between channels, the other STB on same ADSL line may have different bandwidth utilization opportunities.

The switching between the video streams must be transparent to the STB that keeps decoding the received video channel. To avoid changes in the video stream, the PMM has to modify the following fields at the MPEG-TS header:

8 bits	1 bit	1 bit	1 bit	13 bits	2 bits	2 bits	4 bits
Sync Byte	TEI	PUSI	TP	PID	SC	AF	CC
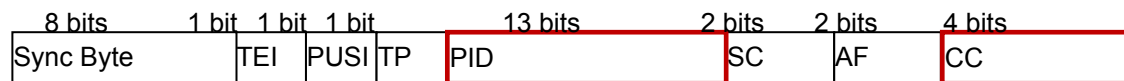
Figure 5 - MPEG-TS header fields

Table 6 - PMM MPEG-TS fields modification

Field	Description
PID	Each video stream at the multi bitrate channel has a different PID value.
	The PID value of the video stream received at the STB has to be constant regardless of the video bitrate switching done by the PDA.
	The PMM modifies the PID value of video MPEG-TS packets to be same and constant per each destination STB.
	The PID value that the PMM uses for the video MPEG-TS towards the STB is selected to be one of the multi bitrate streams PID, so that the PID will be specified in the PMT table and no modifications are required at the PMT entries.
CC	Each video stream at the multi bitrate channel has independent CC counter
	The CC value of the video stream received at the STB has to be fluent regardless of the video bitrate switching done by the PDA
	The PMM modifies the CC counter field of video MPEG-TS packets to be compliant with ISO/IEC-13818-1 standard per each destination STB[*].

[*] The CC is a 4-bit field incrementing with each Transport Stream packet with the same PID. The CC wraps around to 0 after its maximum value. The CC shall not be incremented when the AFC field at the MPEG-TS packet header equals '00' or '10'.

The PDA is actually personalizes the channels to each requesting STB by adapting the channel bitrate to the STB based on the ADSL bandwidth limitation and the combination of competing channels that share the same ADSL link.

The adapted channel has to be directed to the targeted STB as a unicast video stream, and shouldn't be sent as a multicast to all STBs that watch the same channel and are connected to the same DSLAM access device. It should be noted that the video stream has to be transmitted

from the PDA to the STB as a unicast even if the channel is a BCTV channel that has multicast IP address.

This conflict was described in D1.1, and is resolved by applying IP multicast addressing over Layer2 Ethernet unicast addressing, where the IP multicast address is the BCTV channel address and the Ethernet destination MAC address is the destination STB MAC address.

The DSLAM access device is a layer2 device that forwards the IP packets based on their Ethernet destination MAC address, which means that the DSLAM will forward each adapted channel to its destination STB without interfering with other STBs that watch the same channel.

2.1.3 Interaction with Network Management System (EMS/NMS)

As previously described, the PDA prototype integrates two functionalities: A network edge device, and an IPTV PDA.

The interaction between the PDA prototype and the EMS/NMS is only needed for the network related operations, meaning, for the network edge device functionality.

To apply the OptiBand solution on an existing IPTV network, the PDA prototype has to be aware of the ADSL links rate that the device serves (see Figure 1).

Deliverable D5.2 specified two mechanisms for providing the ADSL link rate information to the PDA device, for the prototype, the NMS approach will be adapted for simplicity.

It is assumed that the NMS that manages the DSLAM devices has the information base of its ADSL links and can "get" their rate values, using SNMP or CLI, and "set" the values to the PDA prototype device, per ADSL link. Values can be in 10Kbps resolution.

For the prototype phase, the ADSL rate configuration by the NMS will be done manually via command line interface. The DSLAM will be queried, and the retrieved information will be manually configured to the PDA prototype device, as illustrated below:
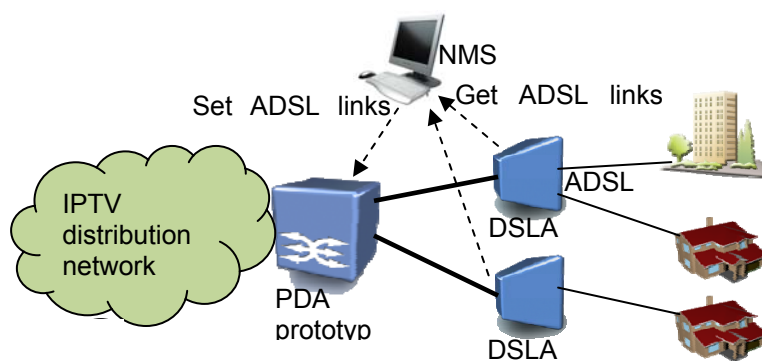
Figure 6 - Prototype interface with NMS

2.1.4 Interaction with MiddleWare

The IPTV middleware is client-server software that connects to the end user STB through the middleware client running at the STB devices, it allows end users to browse media content available in the network, watch videos from the streaming servers, and manage their own preferences such as GUI language, video language, etc. In the server side, the middleware

controls user access rights (so that users only have access to what they have paid for), media acquisition (registering or deleting content from the IPTV platform) and the miscellaneous services that could integrate in an IPTV platform like server-side PVR, EPG, teletext, etc.
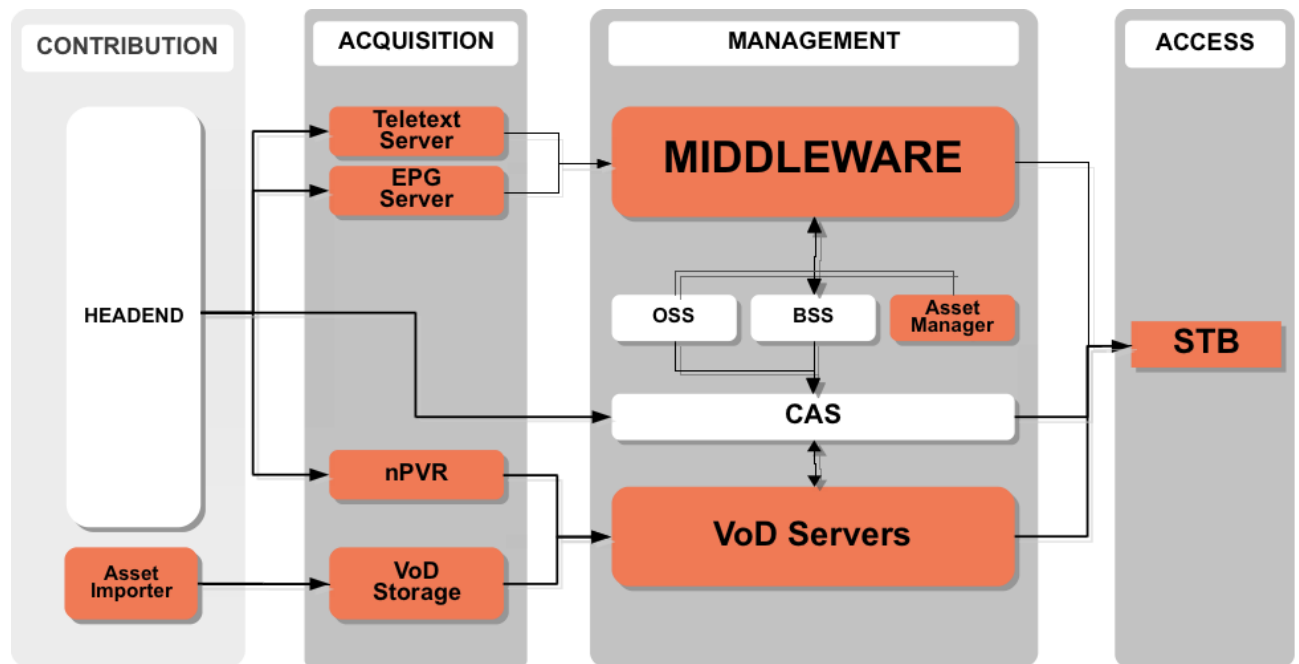
Figure 7 - Middleware deployment

Content providers must introduce the media metadata in the middleware in order to make them available in the service. If those metadata are in an already supported format the process is automatic (using the asset importer), otherwise it should be uploaded manually to the asset manager. For this prototype this is not a serious issue since the amount of assets that would be available is very small so that uploading metadata manually is not a burden. In real scenarios, the content provider and the middleware vendor must integrate to implement an automatic metadata uploading process.

VoD Servers and VoD Storage

Almost any video server will use standard protocols such as HTTP, transport stream over UDP or transport stream over RTP (with RTSP as session control protocol). The integration with the middleware simply requires that an URL is associated to video metadata so that the STB can request the appropriate stream to the VoD server when the user selects a video for watching. Even though both RTSP and HTTP have registered official URL formats in the IANA, transport stream over UDP does not. LambdaStream middleware uses the format `udp://<multicast-address>:port` for that purposes. This doesn't affect to the video servers, but it must be taken into account when generating the metadata to connect video information to the actual stream source.

The Middleware device communicates with the prototype for delivering parameters that are related to the IPTV service functionality and additional parameters to extend the IPTV service to support the OptiBand features.

Practically, the Middleware provides parameters per channel and per STB that will contribute to get better performance of the PDA, leading to better user subjective experiences. The parameters for supporting OptiBand features are specified in Table 7:

Table 7 - Prototype interface with Middleware

Scope	Information	Description
Per channel	Video channel quality	Indicates the quality of the channel using LUT
Per STB	MAC address	The STB MAC address
	Per genre Priority (preference)	End users can configure per each STB the priority of each of the possible genres. For example, users will probably set higher quality for the STB connected to the larger TV in their living room for all genres than for the smaller TV in the kids' room, but less priority for cartoon in the living room than at kid's.

As for the current first release of the prototype, the entire configuration which should be made by the middleware will be made manually to the PDD. For the second release, the middleware will be integrated.

2.1.5 Interaction with Head-End

The Packet Dropping Algorithm (PDA) switches between the different bit rates streams at IDR boundaries. The multi bitrate approach requires that all H.264/AVC streams of same source are IDR-frame synchronized, the same frame has to be encoded by all encoders and the IDR-frame shall start at the same time to allow the PDA to switch between the different bit rate streams transparently to the client.

The H.264/AVC encoded streams are encapsulated with MPEG-TS in an embedded multiplexer. The Multi-video SPTS is provided as described in D3.2 chap 3.3.1.1
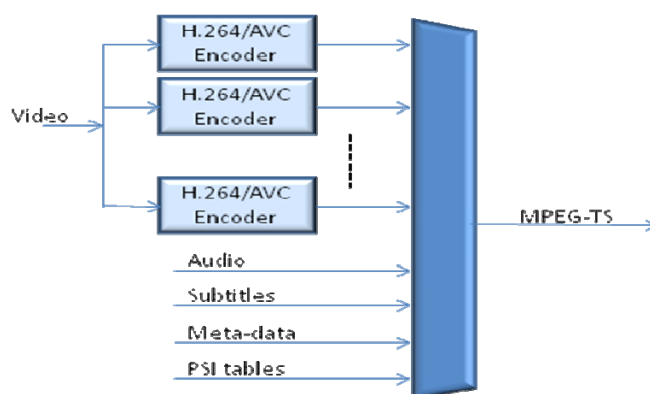
Figure 8 - MPEG-TS Multiplexer

In order to clearly identify Random Access Points (RAP) for the PDD, the Transport priority flag in TS packet header is used. The encoder will flag the first TS packet of each RAP which consists of a SPS NAL unit, a PPS NAL unit and an IDR-frame, indicating that this first packet can be used as a switching point.

For this multi video streams method, all video stream rates of same channel are multiplexed to a single MPEG-TS container. Actually, the MPEG-TS is still considered Single Program Transport Stream (SPTS) and not Multi Program Transport Stream (MPTS) as it transports a single program, with different rate options. Therefore, it is called Multi-video SPTS solution.

The following figure illustrates the MPEG-TS tables to support this method:
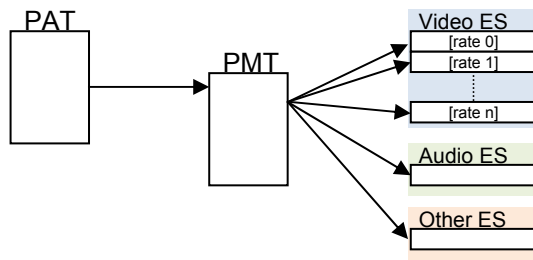
Figure 9 - Multi-video SPTS

The PAT has a single entry for single PMT table, and the PMT has entry per video stream rate.

By transporting all video streams of same program on a single TS/UDP, we can keep a single audio elementary stream (ES) and single subtitle ES (or one per supported language) for all video streams, no need for unnecessary duplication as done for the video ES.

For Multi-video SPTS, the PAT table specifies that the MPEG-TS carry a single program, meaning a single PMT table.

The PMT table specifies all ES within the transported program, as illustrated by the following tables:

PAT (PID = 0)

Program	PID
0	PMT_PID

PMT (PID = PMT_PID)

StreamType	PID
Video rate 0	VID_PID_1
Video rate 1	VID_PID_2
…	…
Video rate k	VID_PID_K
Audio	AUD_PID
PCR	VID_PID_1

Figure 10 - MPEG-2 transport stream Program Specific Information

Synchronization and Timestamps

The MPEG-TS distributes the Program Clock Reference (PCR) that is used for content synchronization at the decoder. The PCR is transmitted in the adaptation field of the MPEG-TS and the PCR_PID is indicated at the PMT table.

The PCR is appended at the adaptation field of each video streams, hence the STB will systematically receive the PCR and will not loss synchronization.

The PCR reference clock is set to be identical for all streams of same contents (at different bitrates), and therefore, the PTS and DTS timestamps are also set at the same value at the picture in all stream rates, to allow the PDA to switch between the MPEG-TS of different video rates, the switching has to maintain the reference clock at the STB.

Continuity-Counter (CC)

The CC is a 4-bit field incrementing with each Transport Stream packet with the same PID. The CC wraps around to 0 after its maximum value.

The Packet Dropping Device (PDD) modifies the CC at the output of the PDA to continually count per PID regardless of the selected video elementary stream.

Video streams PID

Each video stream rate has a different and unique PID that is specified in the PMT table of this channel.

PID of the selected video stream is pre-configured to a value. The Packet Dropping Device (PDD) modifies the PID at the output of the PDA to ensure a constant PID will be received by the STB, making it transparent to the PDA switching decisions.

UDP/IP Specifications

Every video stream rates are transported on the same MPEG-TS and hence on the same UDP/IP connection.

2.1.5.1 Head-End Architectures

The multi bitrate PDA prototype will be integrates with two types of head-end architectures:

Nx [H.264/AVC Single video SPTS] with Multiplexer

H.264/AVC Multi-video SPTS with embedded Multiplexer.

In the 1st architecture, the N encoders are synchronizing the Multi-bitrate channels, and are tagging the sequences with a start of packet indication for PDD mechanism. The multiplexer is used to convert the Single-video SPTS into Multi-video SPTS so as to have a compliant and standardized solution towards the PDD.

This head-end architecture is illustrated in Figure 11:
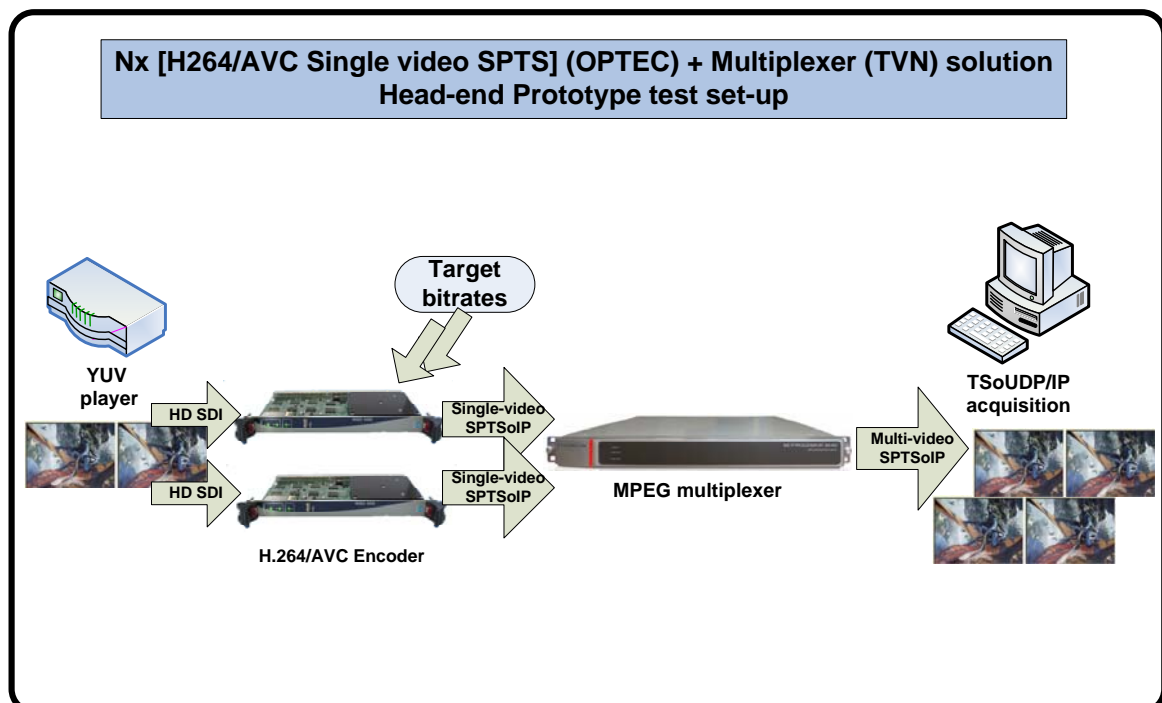
Figure 11 - Nx [H.264/AVC Single video SPTS] with Multiplexer HeadEnd

In the 2nd architecture, the H264/AVC Multi bitrate encoders are used with embedded MPEG-TS Multiplexer. The first target is a Multi-video SPTS prototype with two SD encoded streams with IDR-frames flagged for their first TS packet in order to validate the PDA based on MPEG2 TS system layer, the SD deviation being transparent in the first approach. This head-end architecture is illustrated in Figure 12:
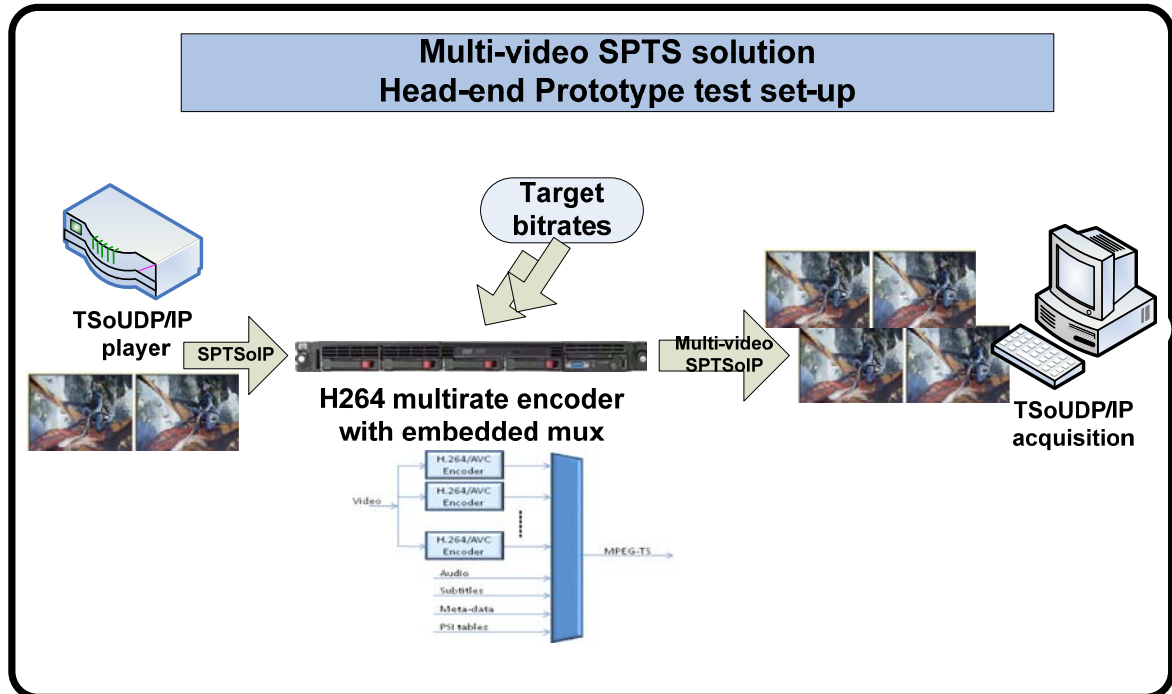
Figure 12 – Nx H264/AVC Multi-video SPTS with embedded Multiplexer

Three SD Multi-video SPTS channel sequences will be used to validate functionality of the 1st PDD prototype version and the interoperability of the multi-stream video with the PDD.

2.1.6 Interaction with STB

The STB devices used for decoding the video stream at the Packet Dropping Algorithm (PDA) output have to be same devices as used in a standard IPTV service that doesn't integrate OptiBand solution since it is not possible to adapt anything in the decoding process without incurring in huge cost requirements, since that will force any operator to replace all their STBs (at least one per end user). For this reason, the multi-rate solution must ensure that the streams the STB will receive comply with the constraints imposed by AVC standard profiles

There are several buffers involved in video playback. The STB uses the reader buffer to absorb possible jitter from the network. Once it starts sending the stream for video decoding, the clock synchronisation mechanism starts and the video processor begins to move data across its internal buffers, assuming the buffering constraints imposed by the MPEG standards will hold. There are several sources of potential problems:

- If the network stalls for too long, the reader buffer will empty and will fail to deliver data to the video engine at the required rate. The video engine will also stall, and the user will observe video artefacts. In this case, the most likely outcome is jerkiness due to images freezing in the screen

- If the network has a bottleneck, it will lose UDP packets, damaged input will be provided to the video processor. Different video engines will deal differently with this problem, but, inevitably, the user will perceive much lower quality due to lost frames or decoding artefacts.

The Packet Dropping Device (PDD) prototype is required to perform the dropping algorithm in a real time avoiding any change in the video stream timing. The algorithm should have a constant latency, otherwise introducing delay variation on the video stream may result in STB buffer overflow or underflow, which will affect the video quality of experience as previously described.

The PDD uses buffers only to build a proper IP/UDP packet with 7 MPEG-TS packets. This functionality is required since packet received from the network interface carries IP/UDP packet with 7 MPEG-TS packets of multiple video streams rate where only one video stream rate is selected by the algorithm while the rest are dropped.

The dropped MPEG-TS packets shorten the IP/UDP packet making the IP/UDP packet at the PDD output not efficient for delivering over the ADSL link due to inefficient overhead per packet. The algorithm aggregates 7 MPEG-TS packets that are allowed to be forwarded to the STB and only then it forwards the packet towards the STB as described in Figure 13:
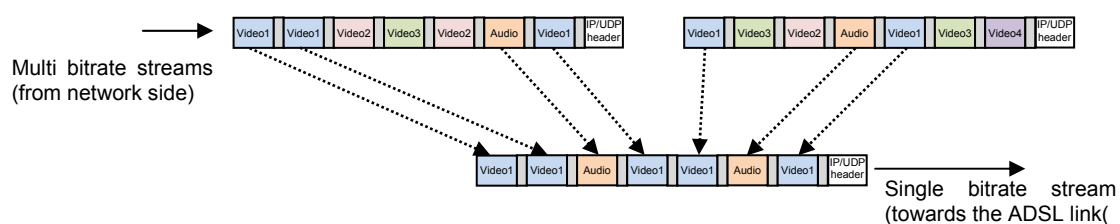
Figure 13 - IP/UDP construction by PDA

For broadcast TV (BCTV) channels, the IPTV STBs are configured with the IP multicast address associated with each broadcast channel. When end user zap between channels, its STB sends IGMP messages towards the network to inform the IPTV network that it changed channel, meaning, it leaved the multicast group associated to previously viewed channel and joined a new multicast group associated to the new requested channel

To allow the PDD prototype to perform the PDA correctly, it is required that the DSLAM devices are configured to operate in IGMP snooping mode, meaning, the DSLAM device that receives the IGMP messages from the user's STB will only "sniff" the message, and will forward the IGMP message unmodified upward to the PDD prototype.

By forwarding the IGMP messages to the PDD prototype, the PDD will be able to identify to which channel each STB is tuned, hence, will have the knowledge how many STBs behind each DSL link and which channel is currently viewed by each STB.

To minimize the overload of IGMP messages on the network and on the multicast router, the PDD prototype will be configured as IGMP proxy, it sends a single IGMP message towards the multicast router for all IGMP messages received on DSLAM interfaces for each multicast channel.

Each STB has a unique MAC address, the multicast channel access a group of STBs (those that were registered to this multicast group), by having multicast MAC address that is derived from the IP multicast address.

The PDD prototype implements Layer-3 multicast IP over Layer-2 unicast MAC. Where the Layer-3 multicast address is the channel multicast group identifier, all STBs on the IPTV network are configured with this multicast address for each multicast channel. The Layer-2 MAC is replaced by the PDD from multicast MAC to unicast MAC per destination STB. The PDD performs the MAC replacement for each STB, The MAC address information per STB is retrieved from the Middleware as previously described.

The STB that receives the multicast IP channel, over unicast MAC address will pick up the packets and will decode the video content and will present it at the appropriate channel according to the IP-Multicast address.

2.1.7 Encryption Consideration

Multiple AVC streams encode video in multiple qualities, deliver all streams to data dropping device and allow the device to drop parts of the unnecessary streams (allowing only the data from a selected stream to pass through). The solution multiplexes multiple streams into a single MPEG-TS stream with each stream having a different PID so the data dropping device rewrites PID of the selected stream to a single pre-defined value that the client device is tuned to and removes packets belonging to other PIDs.

The synchronization difficulty involved in this technique is little-to-nothing. The scrambler that encrypts the streams needs to support manual input of control words, so the same control word can be used to scramble all streams, moreover, since the streams are in the same MPEG-TS, their control periods can be easily synchronized, allowing seamless substitution of videos from different PIDs.

A scrambler application takes the input MPEG-TS, scrambles the contained programs and includes ECMs in the stream for the descrambler. The output of the scrambler is a scrambled MPEG-TS. The PDA selects per client STB the most suitable video stream in terms of quality and bitrate, dropping all other video streams. The scrambler uses the same control word for all streams, hence, the descrambler application can take the scrambled TS, extract the ECM information and descramble the contained programs. The output of the descrambler is a clear TS. The clear TS is equal to the input TS with the exception of the added ECM packets and possible altered PSI table information (the scrambler adds the CA Descriptor, thus modifying the tables).

The scrambler and descrambler applications may use a file or a UDP connection as an input and it outputs data to a file or UDP connection. The scrambler application requires the bit-rate of the incoming stream to calculate the ECM injection rate and Control Word period. Taking an appropriate value from the PCR contained in the stream will add a significant delay in the moment at which scrambling can start.

2.2 SVC with MGS Scalability Prototype [HHI]

This section gives an overview of the SVC based IPTV solution for bandwidth adaptation and presents the SVC based PDD prototype.

The SVC standard [1][4] with MGS SNR scalability is used in order to achieve the objective of the OptiBand project for adapting the video stream rate to the user access rate, which is performed by the packet dropping algorithm. This approach is based on the fact that a single SVC encoded bit stream can be arranged into several operation points which correspond to different bit rates (see example in Table 10). In this case it is not necessary to provide multiple versions of a single video source with the same video content at different bit rates to an adaptation unit in the network side. Therefore, it reduces the load at the head-end and the bandwidth consumption at the metro network. Furthermore, SVC capable STBs are currently being developed, see [1]. All these reasons make the SVC based approach very interesting for future IPTV services.

This kind of video adaptation can be implemented in a very efficient way. The adaptation is performed by a proxy application that runs on the network device and adapts the video stream on-the-fly as it is streamed over the network. The SVC based adaptation just requires dropping of selected IP packets [2]. An interesting approach which improves IPTV services by SVC Adaptation is also presented in [3].

2.2.1 General Description

This section provides an end-to-end general network description to allow the reader to understand the concept of the SVC based solution, and gives an overview of the SVC based Packet Dropping Device.

Figure 14 shows the main components of the SVC based IPTV solution for packet dropping like the Head End, the Packet Dropping Device and the STB emulator for SVC connected to an HD display. These components are used in order to achieve the objective of the project, i.e. to reduce the bandwidth of HD video by 33%.
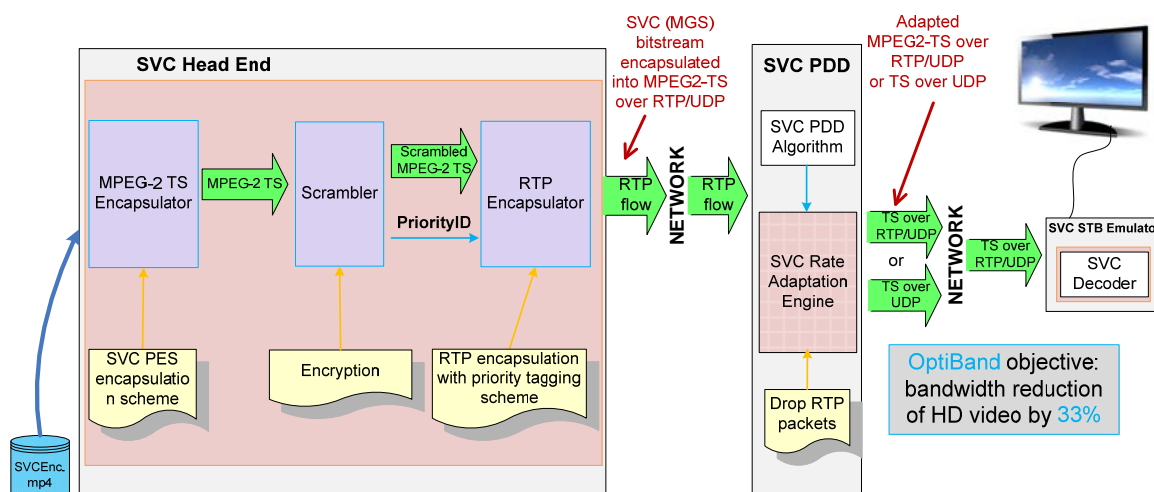
Figure 14 - SVC based IPTV solution for OptiBand

The Head End provides to the SVC based Packet Dropping Device several valid H.264/SVC bit streams which are encapsulated into MPEG2-TS and additionally into RTP packets. The SVC based Packet Dropping Device can either keep or drop a complete RTP packet to adapt HD video streams. The metadata needed for the decision whether to keep or drop an RTP packet is signalled using the SSRC field in the RTP header. The adapted stream, output of the SVC PDD,

is transmitted to a certain STB (PC based STB emulator in case of SVC) to decode the SVC bit stream. Two possible outputs of the SVC PDD are considered:

a) TS over RTP/UDP

b) TS over UDP

The presented approach is expected to achieve 33% bandwidth reduction with minimal degradation in QoE. Further simulations of the SVC based Packet Dropping Algorithms and the prototype implementation will be used to evaluate the proposed adaptation algorithm. These activities will be carried out as planned during the next tasks in WP3.

The Packet Dropping Device needs information about the current status of each DSL link (the bit rate allocated for IPTV services) of each user and the MAC addresses of the connected STBs. We assume in our approach that it will be accordingly provided from a Network Management System and Middleware. The list of video channels with their corresponding multicast addresses must also be provided to the PDD for further forwarding of IGMP requests from STBs. Table 8 summarizes which information the PDD needs.

Table 8 - Necessary information for the SVC Packet Dropping Device

#	From	Destination	Description	Format
1	Network Management System	SVC PDD	Information on available data rate required for each single IPTV service customer	Text string *user: ID: allocated_bitrate_DSL line (MB)*
2	Middleware	SVC PDD	List of MAC addresses of STBs for each single IPTV service customer	Text string *STB: ID: userID : STB MAC-address : IP : port*
3	Middleware	SVC PDD	List of available video channels with their corresponding names, multicast addresses and content types	Text string *channel:ID:Name:MulticastAddress:ContentType*

The parameters that change dynamically are included in the RTP streams from the head-end resp. from the video encoders. For the SVC solution, RTP packets are tagged with a priority that allows for dropping individual packets, respecting coding dependencies and thus keeping the remaining bitstream decodable.

specifies the SVC Packet Dropping Device interfaces. The PDD is adaptingincoming RTP flows in order to reduce bit rate at a single DSL link.

Table 9 - SVC Packet Dropping Device Interfaces

#	Source	Destination	Description	Format
1	SVC based Head End (RTP tagging)	SVC PDD	RTP flows (MPEG-2 TS over RTP/IP) of SVC encoded channels from Head End	RTP flows (MPEG-2 TS over RTP/IP)
2	SVC PDD	DSLAM	SVC bitrate adaptation	Adapted RTP flows (MPEG-2 TS over RTP/UDP/IP)

SVC based Packet Dropping Algorithm

The SVC bit rate adaptation approach is based on multiple Operation Points within the same MGS-encoded scalable stream. An Operation Point (OP) is defined as a unique combination of temporal and quality levels, where reasonable combinations of different pairs of Temporal ID and Quality ID are selected in a way that lower OPs can be easily obtained by dropping data from the bit stream with higher quality and temporal levels than the ones corresponding to the given desired OP.

Each OP refers to a valuable stream at a certain quality level with a corresponding bit rate. In the Head-End, the data units are tagged with priority levels. When passing the PDD, the data rate of the stream is monitored and compared with the data rate available on the DSL link to the SVC decoder in the client device (done by the SVC PDD Controller, see below).

If the incoming data rate exceeds the available data rate, the PDD drops part of the data stream. Dropping starts with NAL units belonging to the quality layer which are tagged with the highest value of Priority ID, i.e. in our case, packets with Priority ID equal to 8 are dropped first (see the Table 10). If the data rate is still too high, the next priority level is discarded. This process continues until the outgoing data rate is within the bounds of the link capacity.

A lookup table (LUT) and the content type are used at the Packet Dropping Device in order to prioritize video streams for dropping. Content types like "Soccer" and "Action Movie" are handled with the highest priority, while other types of content with more static content have less priority.

The following table shows an example how rate adaptation can be achieved using SVC (video sequence "Avatar" 10s long 1080p@25). The Priority IDs are assigned in a way that dropping starts with NAL units belonging to higher quality layers (Q in the table). All temporal levels (T in the table) of the base layer are always forwarded. This resulting number of OPs in this example is 9 (0..8).

Table 10 - Priority assignment to each OP (T, Q layer combination)

OP	T: 0			1			2			3			Rate [kbps]	PSNR [dB]
	Q: 0	1	2	0	1	2	0	1	2	0	1	2		
1	•	•	•	•	•	•	•	•	•	•	•	•	8225,4	40,841
2	•	•	•	•	•	•	•	•	•	•	•	x	7121,6	39,847
3	•	•	•	•	•	•	•	•	x	•	•	x	6295,9	39,211
4	•	•	•	•	•	x	•	•	x	•	•	x	5703,1	38,836
5	•	•	x	•	•	x	•	•	x	•	•	x	4559,6	38,506
6	•	•	x	•	•	x	•	•	x	•	x	x	3960,2	37,764
7	•	•	x	•	•	x	•	x	x	•	x	x	3521,5	37,201
8	•	•	x	•	x	x	•	x	x	•	x	x	3220,2	36,835
9	•	x	x	•	x	x	•	x	x	•	x	x	2670,7	36,437
PriorityID	0	1	5	0	2	6	0	3	7	0	4	8		

The SVC based Packet Dropping Algorithm is presented in pseudo code in Figure 15.

The SVC Packet Dropping Algorithm analyzes periodically DSL links of each authorized single IPTV costumer which has ordered at least one media service over PDD per IGMP request JOIN. For each DSL link, it uses parameters and meta-data information like

- the reserved DSL line rate for an IPTV service,
- a list of all STBs and PC based client devices,
- media streams ordered from STBs or user home devices,
- their receiving rates,
- appropriate rates for each Operation Point,
- content type of each media service.

When a bandwidth bottleneck is detected on a DSL link (see case 1 in the algorithm description), the SVC Packet Dropping Algorithm will decrease the bit rate of affected video streams in a fair way. The Packet Dropping Algorithm begins the rate adaptation process with a stream which has the highest MOS value in the look up table for a given receiving video rate and the content type of a stream. That means that the bit rate of a stream with the best quality regarding MOS value will be decreased first. The constraint of this strategy is to decrease rates of concurrent streams in such a way that the highest possible MOS provided by the lookup table is obtained for each stream, keeping the values for all concurrent streams as similar as possible.

The bit rate of a selected stream is decreased by choosing of the next highest Operation Point (OP) for this stream and dropping all packets which are not needed for decoding the content at this new OP. Information about bit rates for several Operation Points of a stream is provided from the MPEG Analyzer to the SVC Rate Adaptation Engine. The rate adaptation process for a single DSL link is continued until the total amount of bit rates on the DSL link is less than or equal to the reserved DSL line rate. Additionally, the Priority ID is set up for an adapted stream which will be transmitted to certain STB on the single DSL link. The maximum Priority ID is set for a unique combination of adapted stream and STB or SVC decoder in the client device. SSRC field information of the RTP header is used as the signalling information to decide which RTP packets will be dropped at the PDD.

Similarly to the bandwidth bottleneck case, when it is detected that a DSL link does not use the full capacity of the reserved DSL line rate (see case 2 in the algorithm description), the SVC Packet Dropping Algorithm will increase the bit rate of video streams switching to a higher OP. In this case, the algorithm starts rate adaptation from a stream which has a minimal MOS value according to the lookup table and increases bit rates of video streams until the total amount of bit rates on the DSL link is slightly less than or equal to the reserved DSL line rate.

The PDA is also able to detect when a single user changes channels (per receiving IGMP request LEAVE for an already ordered video channel and following JOIN for another video service) and dynamically adapt streams. This functionality is already implicitly realized in the algorithm.

Procedure **SVCPacketDroppingAlgorithm**

#analyze DSL links of each authorized user
For (DSL_link_i){

   # following parameters are known at the PDD for each DSL link:
   # reserved DSL link rate for IPTV services ($DSL\_link_i.rate$ *)*

   #list of all connected STBs and their ordered media services with receiving rates ($STB_i.rate$ *)*

   #appropriate rates for each OP ($OP_j.rate$ *) for each media service*

   #1 Bandwidth bottleneck. Not enough bandwidth on the DSL link

   While ($\sum_{i=1}^{STBs} DSL\_link_l.STB_i.rate > DSL\_link_l.rate$){

      #find a media service with max. MOS value for a given receiving rate on STBs
      *#Constraint: stream adaptation is based on **max** MOS values regarding concurrently streams*
      $j = k / MOSk = \max(MOS_{1..i}(STB_{1..i}.rate, STB_{1..i}.CT));$

      #use look up table for MOS value identification $MOS = f(STB_s.rate, CT)$

      $OP_j++;$ *#use OP_j rate for media service j which STB_j receives*

      $STB_j.rate = OP_j.rate;$ $STB_j.PriorID = OP_j.PriorID$

      If ($STB_j.PriorID > OP_{max} - OP_j$) {

              Drop RTP packets of stream j with $SSRC > OP_{max} - OP_j$ **sent to** STB_j

      } **Else** { **Transmit video service without dropping** }
   }*# end while*

   #2 Enough bandwidth on the DSL link. Increase bit rate of concurrent streams for STBs
   While($\sum_{i=1}^{STBs} DSL\_link_l.STB_i.rate < DSL\_link_l.rate$){

      *#find a media service with **min** MOS value for a given receiving rate on STBs*
      #Constraint: stream adaptation is based on min. MOS values regarding concurrently streams
      $j = k / MOSk = \min(MOS_{1..i}(STB_{1..i}.rate, STB_{1..i}.CT));$

      #use look up table for MOS value identification $MOS = f(STB_s.rate, CT)$

      $OP_j--;$ *# use OP_j rate for media service j which STB_j receives*

      $STB_j.rate = OP_j.rate;$ $STB_j.PriorID = OP_j.PriorID$

      If ($STB_j.PriorID > OP_{max} - OP_j$) {

              Drop RTP packets of stream j with $SSRC > OP_{max} - OP_j$ **sent to** STB_j

      } **Else** { **Transmit video service without dropping** }
   }*# end while*

} *# end for*

Figure 15 - SVC Packet Dropping Algorithm

This section describes the software architecture of the SVC-based Packet Dropping Device. The SVC based PDD is located at the metro network and connects DSLAMs to the transport network. Figure 16 illustrates the function of the SVC Packet Dropping Device and its main components.
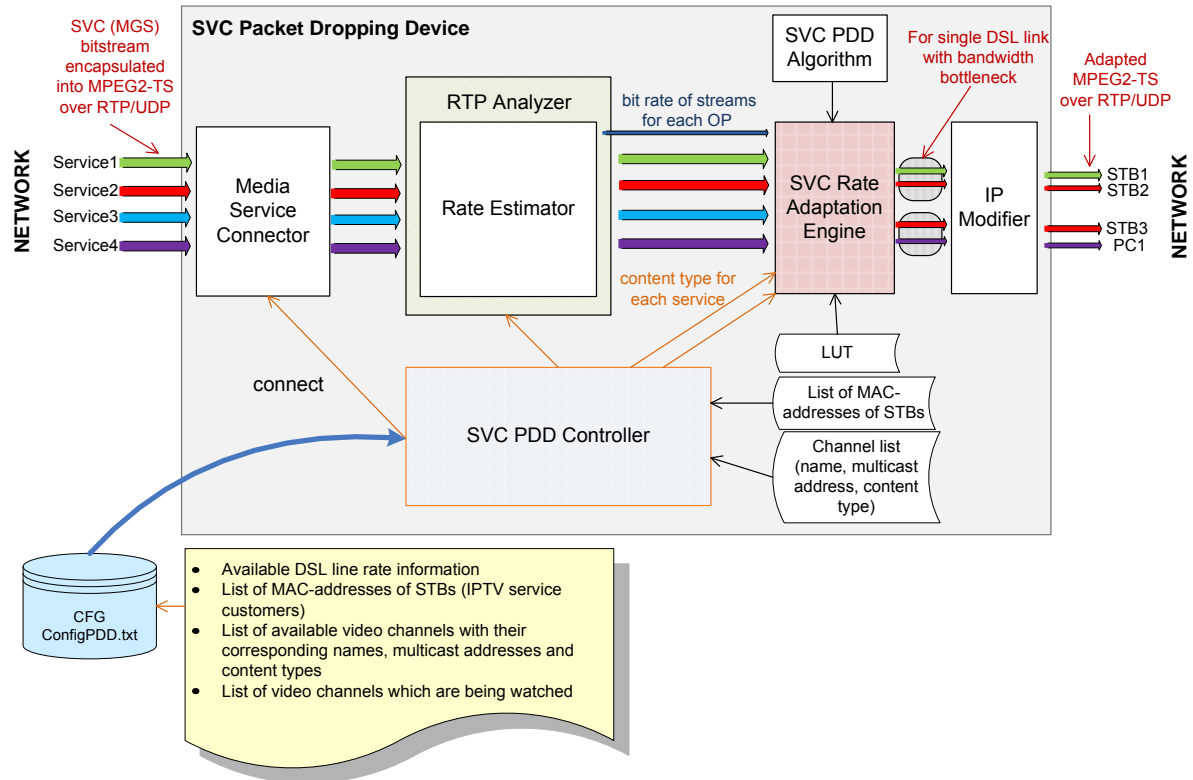
Figure 16 - SVC Packet Dropping Device

The PDD prototype is implemented in C++ using Boost [10] and Qt [11] frameworks. The functionality of each used component in the software architecture is explained below.

SVC PDD Controller:

The component SVC PDD Controller is the control unit of the PDD. This unit controls and monitors the components Media Service Connector, RTP Analyzer and SVC Rate Adaptation Engine. It has also an access to a configuration file ("ConfigPDD.txt") with important configuration parameters and manages the rate adaptation process. Configuration parameters are presented in Table 11. SVC PDD Controller has the knowledge of how many STBs are behind each DSL single link, which DSL links have a bottleneck and which channel is currently received and decoded by a certain STB. This information is used in the SVC Rate Adaptation Engine to decide which streams will be adapted and which STBs and DSL links are affected in this way.

Table 11 – Configuration parameters for SVC based PDD prototype (ConfigPDD.txt)

#	Item	Description	Format
1	Available DSL line rate information	Information on available data rate required for each single	Text string *user: ID: allocated_bitrate_DSL line (MB)* Example: **user:1:13;**

#	Item	Description	Format
		IPTV service customer	**user:2:13;** **user:3:20;**
2	List of MAC-addresses of STBs	List of MAC addresses of STBs for each single IPTV service customer	Text string *STB: ID: userID : STB MAC-address : IP : port* Example: **STB:1:1:00-01-38-6D-02-02: IP: port;** **STB:2:1:00-01-38-6D-02-03: IP: port;** **STB:3:2:00-01-38-6D-02-04 : IP: port;** **STB:4:2:00-01-38-6D-02-05: IP: port;** **STB:5:3:00-01-38-6D-02-06: IP: port;** **STB:6:3:00-01-38-6D-02-07: IP: port;** **STB:7:3:00-01-38-6D-02-08: IP: port;**
3	Channel list	List of available video channels with their corresponding names, multicast addresses and content types	Text string *channel:ID:Name:MulticastAddress:ContentType* Example: **channel:1:Name1:225.0.0.1:action;** **channel:2:Name2:225.0.0.2:action;** **channel:3:Name3:225.0.0.3:action;** **channel:4:Name4:225.0.0.4:soccer;** **channel:5:Name5:225.0.0.5:soccer;**
4	Watching list	List of video channels which are being watched	Text string *watching: UserID : STBID : ChannelID* Example: **watching:1:1:1** **watching:1:2:2** **watching:2:3:2** **watching:2:4:3** …

The Watching list from the configuration file presents a list of video channels which has been ordered by IPTV costumers. It simulates signalling of IGMP requests (JOIN and LEAVE).

The configuration file "ConfigPDD.txt" (see

Table *12*) provides enough information for testing of the dynamic rate adaptation based on the SVC approach.

Table 12 – Configuration file for the SVC PDD prototype

```
#user: ID: allocated_bitrate_DSL line (MB)

user:1:13;

user:2:13;

user:3:20;

# STB: ID: userID : STB MAC-address : IP : port

STB:1:1:00-01-38-6D-02-02: IP: port;

STB:2:1:00-01-38-6D-02-03: IP: port;

STB:3:2:00-01-38-6D-02-04 : IP: port;

STB:4:2:00-01-38-6D-02-05: IP: port;

STB:5:3:00-01-38-6D-02-06: IP: port;

STB:6:3:00-01-38-6D-02-07: IP: port;
STB:7:3:00-01-38-6D-02-08: IP: port;

# channel:ID:Name:MulticastAddress:ContentType

channel:1:Name1:225.0.0.1:action;

channel:2:Name2:225.0.0.2:action;

channel:3:Name3:225.0.0.3:action;

channel:4:Name4:225.0.0.4:soccer;
channel:5:Name5:225.0.0.5:soccer;

# watching: UserID : STBID : ChannelID

watching:1:1:1

watching:1:2:2

watching:2:3:2

watching:2:4:3

…
```

Media Service Connector:

This module receives SVC (MGS) encoded bit streams with 3 quality layers which are MPEG2-TS encapsulated and transmitted over RTP/UDP/IP. It monitors receiving streams (channels). It is able to start and to terminate receiving of video streams from the Head End. Its interface is given in Table 13

Table 13 - Media Service Connector Interface

#	From	Destination	Description	Format
1	IPTV Head End	Media Service Connector	Receive media streams which have been ordered from STBs	Program call, (SVC bitstream over MPEG2-TS/RTP/UDP/IP)
2	SVC PDD	Media Service	a) Connect to the Head	Program call

#	From	Destination	Description	Format
	Controller	Connector	End and receive media data

b) Terminate receiving of video streams from the Head End. | |

RTP Analyzer:

This component receives SVC streams which are encapsulated into MPEG2-TS and transmitted over RTP/UDP/IP. Its interface is given in Table 14. The Rate Estimator module receives RTP packets and analyzes packets that were identified by different SSRC values and calculates the bitrate of video streams for each Operation Point. Afterwards the RTP Analyzer forwards video streams unchanged to the SVC Adaptation Engine as well as the current content type and the estimated rates for each receiving video service with its appropriate Operation Points.

Table 14 - RTP Analyzer Interface

#	From	Destination	Description	Format
1	Media Service Connector	RTP Analyzer	Forward media streams which have been ordered from STBs	Program call, (SVC bitstream over MPEG2-TS/RTP/UDP/IP)
2	RTP Analyzer	SVC Rate Adaptation Engine	Forward media with the parsed content type and the estimated rate of each receiving video stream.	Program call, (SVC bitstream over MPEG2-TS/RTP/UDP/IP)

The components SVC PDD Controller and MPEG Analyzer are handling following information at the PDD:

- the reserved DSL line rate for an IPTV service,
- a list of all STBs and PC based client devices,
- media streams ordered from STBs or user home devices,
- their receiving rates,
- appropriate rates for each Operation Point,
- content type of each media service.

SVC Rate Adaptation Engine:

The SVC Rate Adaptation Engine component is responsible for rate adaptation of receiving streams for each single DSL link with potential bandwidth variations, i.e. bottlenecks or additional bandwidth due to disconnection of other users. This module uses the SVC Packet Dropping Algorithm, additionally information from MPEG Analyzer regarding the content type and estimated rates regarding OPs of each receiving video stream as well as a lookup table (LUT) with video quality information. A lookup table provides information about average MOS values for each certain content type and bandwidth level (BL) and connects bitrate values of a specific content type with MOS values.

The SVC PDD Controller controls the SVC Rate Adaptation Engine module and provides to it for that reason the current information about the status of DSL link as well as about connected STBs and their ordered video services. The SVC Rate Adaptation Engine module then decides

autonomous using the SVC Packet Dropping Algorithm in which way video streams will be adapted.

Table 15 - SVC Rate Adaptation Engine Interface

#	From	Destination	Description	Format
1	RTP Analyzer	SVC Rate Adaptation Engine	Forward media with the estimated rate of each receiving video stream. The *SVC Rate Adaptation Engine* will adapt streams if necessary.	Program call, (SVC bitstream over MPEG2-TS/RTP/UDP/IP)
2	SVC PDD Controller	SVC Rate Adaptation Engine	a) notifies adaptation of streams for some STBs b) provides necessary information about STB, data line on the DSL link with bandwidth bottleneck	Program call
3	SVC Rate Adaptation Engine	IP Modifier	Forward adapted media streams to modify IP packet header	Program call. (SVC bitstream over MPEG2-TS/RTP/UDP/IP)

The SVC Rate Adaptation Engine uses the SVC Packet Dropping Algorithm in order to adapt video streams of authorized users which have a bottleneck on their DSL links.

IP Modifier

This module receives adapted MPEG2-TS streams which are encapsulated into RTP/UDP/IP packets and modifies the destination information of the IP header to ensure that the stream adaptation is done transparently for the end user. IP packets with the modified destination information are sent to appropriate STBs.

Table 16 - IP Modifier Interface

#	From	Destination	Description	Format
1	SVC Rate Adaptation Engine	IP Modifier	Receive media streams which have been ordered from STBs	Network interface. (SVC bitstream over MPEG2-TS/RTP/UDP/IP)
2	IP Modifier	STB	Forward adapted streams to STB	Network interface, (SVC bitstream over MPEG2-TS/RTP/UDP/IP)

A FIFO buffer is used in order to smooth the bit rate at the output of the PDD. It will be done as a subcomponent of IP Modifier by pulling packets from the buffer and pushing small chunks of high-bitrate burst into the network. When a video stream is encoded with variable bit rate (VBR), bit allocation and distribution varies depending on the complexity and motion of each scene. This is done to obtain an optimal video quality while not consuming more resources than needed. The video variability is difficult to measure and depends on the chosen encoding parameters of the video clip, mainly the mean encoding bit rate (CER) and the peak encoding bit rate (PER). The greater the difference between these two parameters, the greater is the

assumed variability in the video stream which results in great frame size variability. To mitigate for this variability for the transmission of video over a network and for better provisioning of network resources, certain measures are needed [6][7][8].

Graphical User Interface (GUI) for SVC PDD:

The Qt [11] framework is used for GUI implementation of the SVC based Packet Dropping Device.
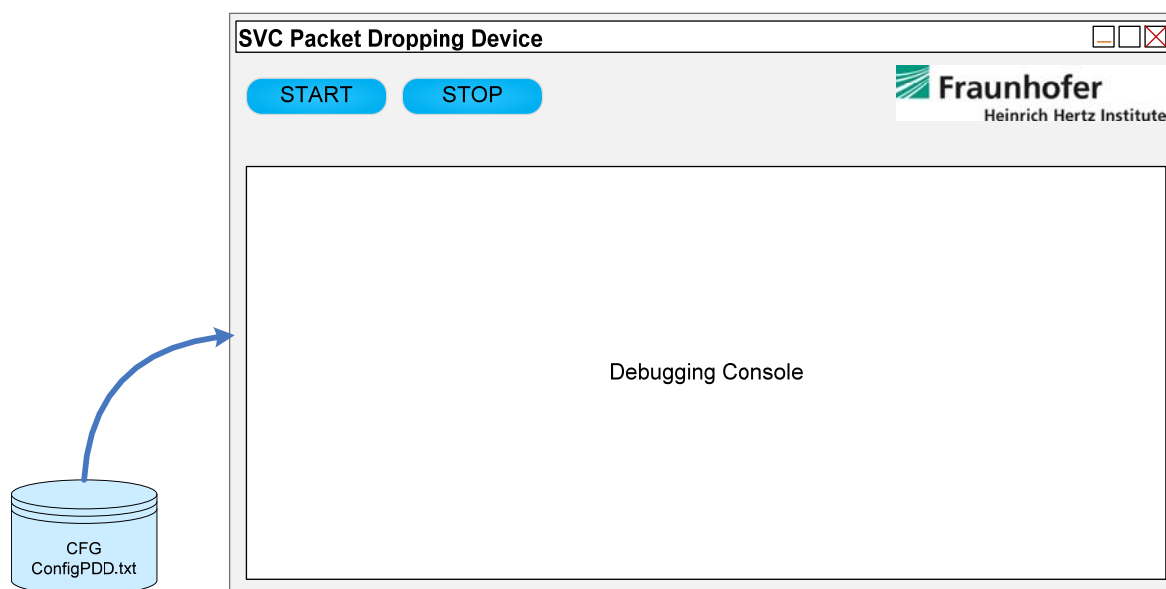

Figure 17 – GUI of the SVC Packet Dropping Device

The GUI enables:

- Start and stop of the PDD
- Debugging of the rate adaptation process using the debugging console

After starting the PDD by clicking the button START, all internal components will be initialized and the PDD immediately begins with the rate adaptation with request to the information from the configuration file.

2.2.3 Interaction with Network Management System (EMS/NMS)

The configuration file "ConfigPDD.txt" is used in order to simulate interaction between the Network Management System (EMS/NMS) and the SVC based PDD. We use the following format for the Interaction with Network Management System: *#user: ID: allocated_bitrate_DSL line (MB.)* The allocated bitrate on the DSL line of the single user (IPTV customer and STB owner) is needed at the PDD to demonstrate the general functionality of the SVC based solution for packet dropping.

2.2.4 Interaction with MiddleWare

The configuration file "ConfigPDD.txt" is used in order to simulate interaction between the Middleware and the SVC based PDD. Information delivered from the Middleware needed at the PDD is written into the configuration file.

2.2.5 Interaction with Head-End

This section describes the interaction of the SVC based PDD with the Head End.

As already mentioned above, the Head End provides to the SVC based Packet Dropping Device several valid encrypted H.264/SVC bit streams which are encapsulated into MPEG2-TS, scrambled and additionally encapsulated into RTP packets. The SVC based Packet Dropping Device can either keep or drop a complete RTP packet to adapt HD video streams. The metadata needed for the decision whether to keep or drop an RTP packet is signalled using the SSRC field in the RTP header.

The SVC based solution for optimisation of bandwidth for IPTV video streaming consists of five main processes which will be executed at the Head End:

- SVC MGS encoding of the video sequence and defining of several reasonable operation points (OP) within the SVC bit stream;
- Priority ID assignment for each SVC NAL (Network Abstraction Layer) units of the SVC bit stream. Priority ID is associated to the pair of { T, Q } provided by the NAL unit header;
- Encapsulation of the SVC NAL units into MPEG-2 Transport Stream (TS) using a specific PES Encapsulation due to Quality ID;
- Scrambling of the video content;
- The corresponding RTP encapsulation of TS packets which have the same Priority ID and tagging of the additionally metadata information such as priority information using SSRC field in the RTP header for multicasting over the IPTV System;

The SVC test bench consists of four main software components, namely: *the SVC Encoder*, the *MP4 File Creator*, the *MPEG-2 TS Encapsulator* and *RTP Encapsulator*. This test bench describes the technical basis of all the technologies involved in the SVC based Head End processing.

Additionally, an adapted stream will be transmitted to certain STB (PC based STB emulator in case of SVC) to decode the SVC bit stream.

The SVC based Head End is divided into two main functional modules as presented in Figure 18: **SVC Encoder and MP4 File Creator** and **MPEG2-TS and RTP Encapsulator.**
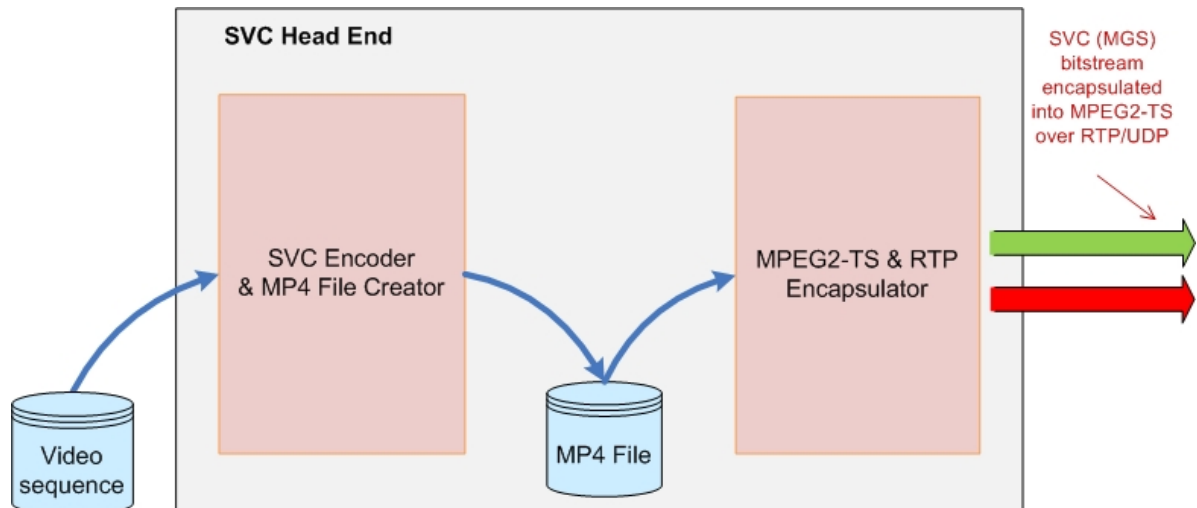
Figure 18 – SVC Head End

We assume, that the Head End provides to the SVC based Packet Dropping Device several valid H.264/SVC bit streams which are encapsulated into MPEG2-TS and additionally into RTP packets.

For this purpose, the MPEG-2 TS Encapsulator and RTP Encapsulator at the Head End for SVC based approach are already presented with more details in D5.2 (see section 3.2.1 "SVC encoder and packet tagging").

Only TS packets that contain NAL units with the same Quality ID and Temporal ID are sent within the same RTP packet. Packets that belong to different temporal levels are anyway encapsulated in separate RTP packets. By this means, the Packet Dropping Device can either keep or drop a complete RTP packet. The metadata needed for the decision whether to keep or drop an RTP packet is signalled using the SSRC field in the RTP header. Figure 19 illustrates the working principle of the encapsulation and the additional tagging process of SVC data using RTP header. More details about the SVC encapsulation approach for MPEG-2 TS is found in Deliverable D4.1 with corresponding diagrams (D4.1 Figures 22-24).
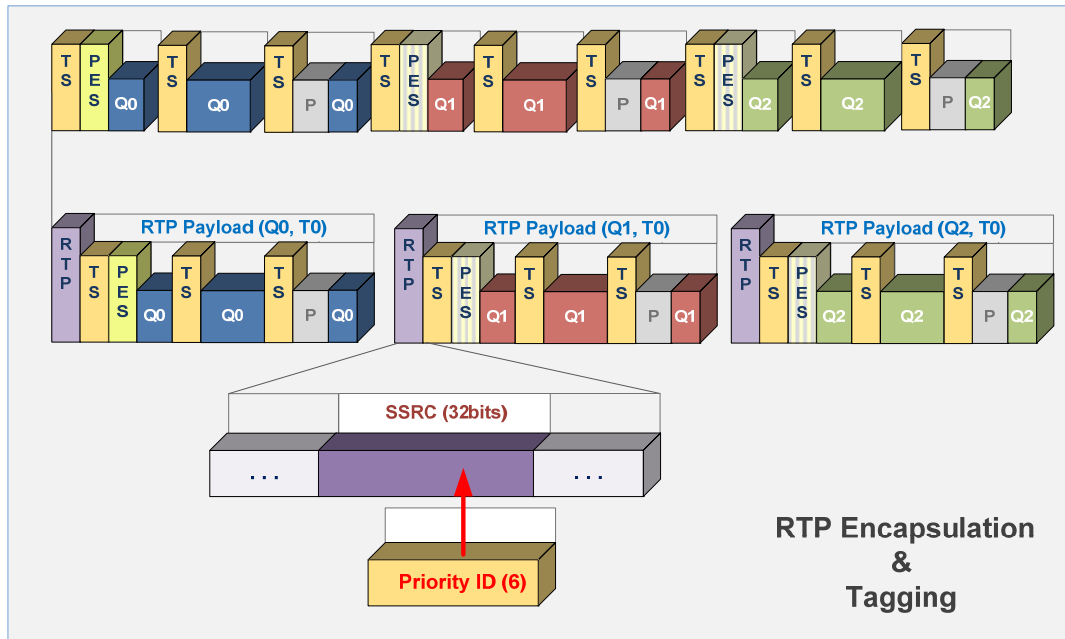
Figure 19 - RTP/MPEG2 TS encapsulation of SVC & Tagging with Priority ID

Table 17 lists properties of video sequences which are used for testing of the SVC based Head End.

Table 17 – Video format of streams for the SVC based Head End simulation

Video sequence	Standard	Content	Max bitrate	Video resolution	Frame rate (fps)	Video mode	GOP length
CC1_Action_ClipA	H.264/SVC	Action movie	7 ÷ 8.5 Mbps	1920x1080	25	VBR	8 (Intra frame every 24 frames)
CC1_Action_ClipB	H.264/SVC	Action movie	7 ÷ 8.5 Mbps	1920x1080	25	VBR	8 (Intra frame every 24 frames)
CC1_Action_ClipC	H.264/SVC	Action movie	7 ÷ 8.5 Mbps	1920x1080	25	VBR	8 (Intra frame every 24 frames)

Table 18 shows rate properties of SVC encoded streams (VBR) with respect to different Operation Points.

Table 18 - VBR SVC Streams - Rate Properties

	Avg. rate	Avg. rate	Avg. rate	Avg. rate	Avg. rate	Avg. rate	Avg. rate	Avg. rate	Avg. rate [kbps]

	[kbps] OP0	[kbps] OP1	[kbps] OP2	[kbps] OP3	[kbps] OP4	[kbps] OP5	[kbps] OP6	[kbps] OP7	OP8 /Base Layer
CC1_Action_ClipA	8225,3	7121,6	6295,9	5703,0	4559,6	3960,2	3521,5	3220,1	2670,7
CC1_Action_ClipB	7964,2	7024,9	6297,4	5708,4	4460,7	3909,9	3519,7	3185,9	2531,5
CC1_Action_ClipC	8476,3	7384,3	6515,6	5892,2	4697,6	4072,1	3590,2	3281,1	2652,5
CC1_Action_ClipD	8213,6	7349,7	6621,4	6011,9	4316,5	3853,0	3501,8	3190,7	2319,7

2.2.6 Interaction with STB

This section presents the interaction of the SVC based PDD with the SVC based STB emulator.

Figure 20 illustrates the SVC STB emulator. It based also on the VLC media player implementation [9] and is used to decode the SVC bit stream. RTP Access and Decoder Plug-ins as well as the SVC Decoder have been developed by Fraunhofer HHI. The PC-based STB emulator that embeds the RTP Access and Decoder Plug-ins and provides a set-top-like interfaces has been developed by UDC.
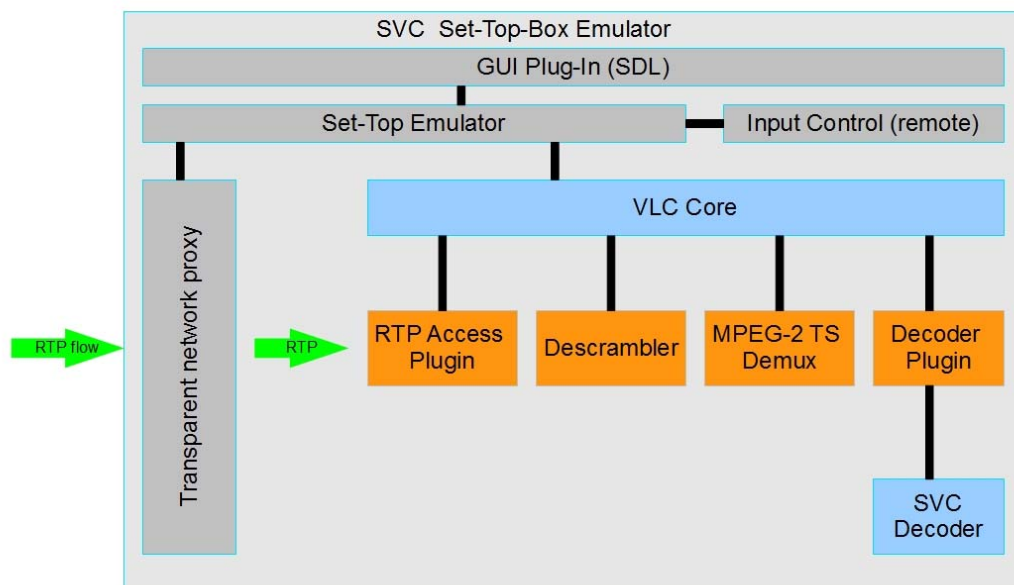

Figure 20 – SVC Set-Top-Box emulator

The STB emulator is fully implemented in software and running on a regular PC, with no hardware assistance for audio / video decoding; the lack of a specialized hardware unit is more than compensated by the massive difference in processing power over a real set-top box.

The emulator is a software engine that implements the same basic functionality as the embedded client in a real set-top box; currently it has support for retrieving video streams from UDP, switching channels under user control, displaying a graphical user interface, receiving input commands (key presses) from a remote control unit, and decoding and rendering the video.

The emulator engine is developed in Erlang / OTP (Open Telecom Platform), and the same byte-compiled code runs in any supported platform, including Linux and Windows.

Video decoding and rendering is performed by embedding a VLC player (VideoLan Client) running in slave mode, controlled by the set-top-box emulator. The emulator can start and then control (pause, play, switch media) VLC; it plays the same exact role as the demuxer / decoder chain in a physical set-top-box.

The RTP access and decoder plugins developed by HHI are dynamically loaded as modules by VLC. Without them, the set-top-box emulator does not work for SVC flows, but can be used for regular H.264 AVC content.

The Input Control module is implemented as a pluggable module in the emulator; currently it reads keyboard presses or buttons from a universal Gyration remote control. Alternate versions can be developed with ease to read an infrared port or a Bluetooth socket.

Similarly, we have tried to keep the Graphical User Interface module separate from the emulator engine; however it must be noticed that multiple aspects of the user interface in a set-top-box are alien to most graphics interfaces – there is no real concept of "windows", but rather there are "planes"; the "focus" is somewhat less clear, etc.

Currently, the Graphical User Interface is developed using the SDL library.

Additionally, the emulator may start an optional network proxy; it is intended to monitor the incoming streams and provide some simple insight on the traffic. Currently it is simply monitoring instantaneous bitrate in a short sliding window (1 second), and a realtime bitrate graph can be superimposed on the screen by pressing a key.

All components have been chosen to be easily ported. While the usual platform at the UDC group is Linux, the RTP plugin and SVC decoder are usually developed for Windows. We have chosen libraries and components (VLC, SDL, Erlang/OTP) that can be simply ported to any of these, or other, platforms. Right now the set-top-box emulator runs both on Windows and Linux – of course, on Linux only AVC streams can be decoded.

2.2.7 Encryption Consideration

The SVC based PDA works well in an encrypted domain. The Algorithm handles encrypted streams for bandwidth adaptation and uses only the RTP header (SSRC field) for signalling of metadata. The PDA is transparent to an encryption model and does not need an additional access to scrambled video streams for adaptation.

3. Prototype Testing

Deliverable D1.1 defined two typical test scenarios:

1. Three TV sets connected through 15Mbps (net available for IPTV) ADSL2+ line
2. Two TV sets connected through 10Mbps (net available for IPTV) ADSL2+ line

These two test scenarios can be characterised by statically allocating 5Mbps per TV device, which practically means that the OptiBand PDA goal is to reduce the channels bandwidth to 5Mbps on average, and to transport 3 HD channels over 15Mbps ADSL2+ line (or 2 HD channels over 10Mbps ADSL2+ line).

A 5Mbps per channel after 33% bandwidth reduction implies that the channels are originally encoded at ~8Mbps rate.

To validate the algorithms and to compare their QoE performance we used 3 action clips Action_ClipA, Action_ClipB, and Action_ClipC (each clip encoded to 8.2Mbps) as input video channels, and defined 3 test scenarios as specified in Table 19:

Table 19 - Simulation test scenarios

Test scenario	Number of active STBs		Required IPTV available ADSL rate	Bandwidth reduction
#1	2	{STB1 displays Action_ClipA channel, STB2 displays Action_ClipB channel}	13Mbps	~20%
#2	2	{STB1 displays Action_ClipB channel, STB2 displays Action_ClipC channel}	13Mbps	~20%
#3	3	{STB1 displays Action_ClipA channel, STB2 displays Action_ClipB channel, STB3 displays Action_ClipC channel}	19.5Mbps	~20%

3.1 AVC Multi Bitrate Streams Prototype

As mentioned in 3.1.6.1, three SD Multi-video SPTS channel sequences will be used to validate interoperability with the PDD on this prototype.

3.1.1 Prototype Test Description

In order to simulate the head-end network, a small PC application was written in order to read the above SD multi-video STPS channel sequences from files and to stream it toward the PDD while keeping certain timing constraints.

The PDD selects the right streams and forward them to the STB simulator (using VLC player on a PC).

At the current first version no IGMP is implemented those no zapping support.

The entire relevant information for the PDD, such as channels, addresses, DSL rates and etc, were provisioned through CLI commands.
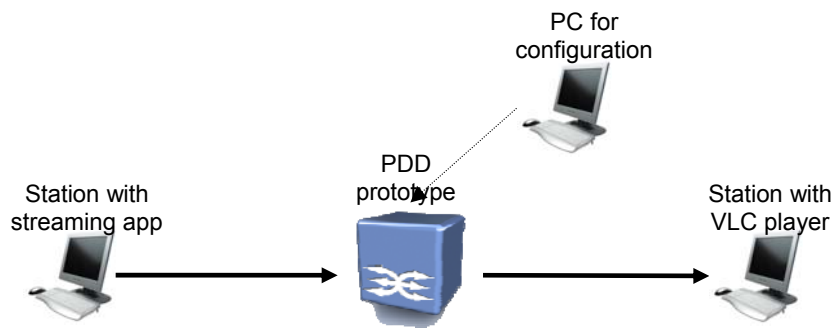
Figure 21 – Prototypes version 1 testing setup

The PDD will be used in order to measure the output rate from the PDD itself toward the PC VLC player. Since the measurement is done on the output port, after playing the PDA, this can be considered as the bit-rate that will arrive to the DSLAM.

The VLC player was configured to listen and play three channels, each on a specific unicast address: 10.10.10.101, 10.10.10.102, 10.10.10.103

3.1.2 Prototype Test Results

3.1.2.1 Test Scenario #1

ADSL rate is configured to 13mbps

Table 20 – Test Scenario #1

Step	playing channel	destination IP addresses	consuming output bitrate (mbps)		VLC playing scene
1	Action_ClipA	10.10.10.101	8.2	8.2	Action_ClipA
2	Action_ClipA Action_ClipB	10.10.10.101 10.10.10.102	6.5 6.5	13	Action_ClipA Action_ClipB
3	Action_ClipB	10.10.10.102	8.2	8.2	Action_ClipB
4	Action_ClipB Action_ClipA	10.10.10.102 10.10.10.101	6.5 6.5	13	Action_ClipB Action_ClipA
5	None	none	0	0	none

3.1.2.2 Test Scenario #2

ADSL rate is configured to 13mbps

Table 21 – Test Scenario #2

Step	playing channel	destination IP addresses	consuming output bitrate (mbps)		VLC playing scene
1	Action_ClipB	10.10.10.102	8.2	8.2	Action_ClipB

Step	playing channel	destination IP addresses	consuming output bitrate (mbps)		VLC playing scene
2	Action_ClipB Action_ClipC	10.10.10.102 10.10.10.103	6.5 6.5	13	Action_ClipB Action_ClipC
3	Action_ClipC	10.10.10.103	8.2	8.2	Action_ClipC
4	Action_ClipC Action_ClipB	10.10.10.103 10.10.10.102	6.5 6.5	13	Action_ClipC Action_ClipB
5	none	none	0	0	none

3.1.2.3 Test Scenario #3

ADSL rate is configured to 19.5mbps.

Table 22 – Test Scenario #3

Step	playing channel	destination IP addresses	consuming output bitrate (mbps)		VLC playing scene
1	Action_ClipA	10.10.10.101	8.2	8.2	Action_ClipA
2	Action_ClipA Action_ClipB	10.10.10.101 10.10.10.102	8.2 8.2	16.4	Action_ClipA Action_ClipB
3	Action_ClipB	10.10.10.102	8.2	8.2	Action_ClipB
4	Action_ClipB Action_ClipA	10.10.10.102 10.10.10.101	8.2 8.2	16.4	Action_ClipB Action_ClipA
5	Action_ClipB Action_ClipA Action_ClipC	10.10.10.102 10.10.10.101 10.10.10.103	6.5 6.5 6.5	19.5	Action_ClipB Action_ClipA Action_ClipC
6	Action_ClipB Action_ClipC	10.10.10.102 10.10.10.103	8.2 8.2	16.4	Action_ClipB Action_ClipC
7	Action_ClipC	10.10.10.103	8.2	8.2	Action_ClipC
8	none	None		0	none

3.2 SVC with MGS Scalability Prototype [HHI]

As mentioned before, SVC streams are encapsulated in MPEG2-TS and further in RTP with the priority signalling in the SSRC field. Action_ClipA, Action_ClipB and Action_ClipC are encoded with SVC with 3 layers and are used for being transmitted to the PDD and simulate the scenarios described in Table 19.

3.2.1 Prototype Test Description

The SVC head-end, as well as the PDD and the SVC VLC, are going to be running in PC. The same setup as shown in **Error! Reference source not found.** is used for testing the PDD, where the IP address and ports of the PC with the VLC client are set manually at the PDD and the head-end is configured to send the data to the PDD.

The results are shown in the following figures (Figure 22, Figure 23 and Figure 24), where the output rate of each of the streams (respectively receiving bitrate for each STB -R_STB_x) as well as the corresponding operation points or priority IDs are shown (OP_STB_x). Note that the

streams are coded VBR (see section 2.2.3 in D4.4.1) and therefore although only one STB is connected the streams set the limiting bitrate that can be provided as output at the PDD (e.g. for Figure 22 R_STB_2 at 11 seconds being alone and receiving 6 Mbps). Note also that the streams used are limited in time so they are repeated in a loop to extend the testing timeline for these results.

3.2.2 Prototype Test Results

3.2.2.1 Test Scenario #1

The output of the PDD is limited to 13 Mbps and two VLC clients are connected to receive Action_ClipA and Action_ClipB. Similar to **Error! Reference source not found.**, different steps are defined as follows:

- Step1: Only STB1 is connected receiving Action_ClipA (from 0 to 5 seconds)
- Step2: STB1 and STB2 are connected receiving Action_ClipA and Action_ClipB respectively (from 5 to 10 seconds)
- Step3: Only STB2 is connected receiving Action_ClipB (from 10 to 15 seconds)
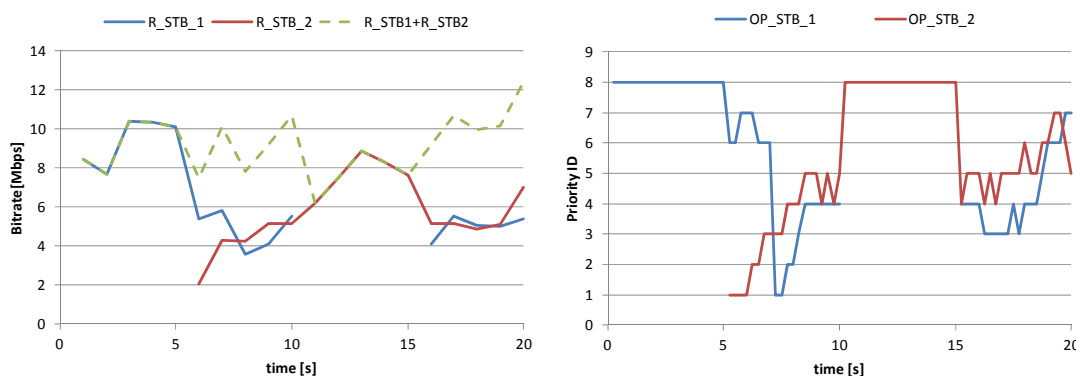- Step4: STB1 and STB2 are connected as for Step2 (from 15 to 20 seconds)

Figure 22: Bitrates and Operation points for test scenario#1

3.2.2.2 Test Scenario #2

For this test, again the output of the PDD is limited to 13 Mbps and two VLC clients are connected, but this time to receive Action_ClipB and Action_ClipC. Similar to the previous case steps are defined.

- Step1: Only STB1 is connected receiving Action_ClipB (from 0 to 5 seconds)
- Step2: STB1 and STB2 are connected receiving Action_ClipB and Action_ClipC respectively (from 5 to 10 seconds)
- Step3: Only STB2 is connected receiving Action_ClipB (from 10 to 15 seconds)
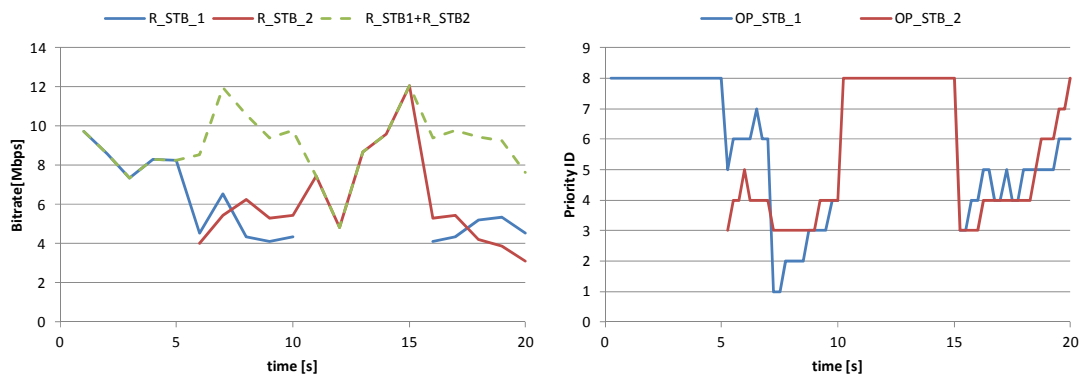- Step4: STB1 and STB2 are connected as for Step2 (from 15 to 20 seconds)

Figure 23: Bitrates and Operation points for test scenario#2

.

3.2.2.3 Test Scenario #3

In this case the output of the PDD is limited to 19.5 Mbps. In this case three STB are connected and Action_ClipA, Action_ClipB and Action_ClipC are transmitted. The steps simulated for this test scenario are defined as follows:

- Step1: Only STB1 is connected receiving Action_ClipA (from 0 to 4 seconds)
- Step2: STB1 and STB2 are connected receiving Action_ClipA and Action_ClipB respectively (from 4 to 8 seconds)
- Step3: Only STB2 is connected receiving Action_ClipB (from 8 to 12 seconds)
- Step4: STB1, STB2 and STB3 are connected receiving Action_ClipA, Action_ClipB and Action_ClipC respectively (from 12 to 16 seconds)
- Step 5: STB1 and STB3 are connected receiving Action_ClipA and Action_clipC respectively (from 16 to 20 seconds)
- Step 6: STB1, STB2 and STB3 are connected as described for Step4 (from 20 to 24 seconds)
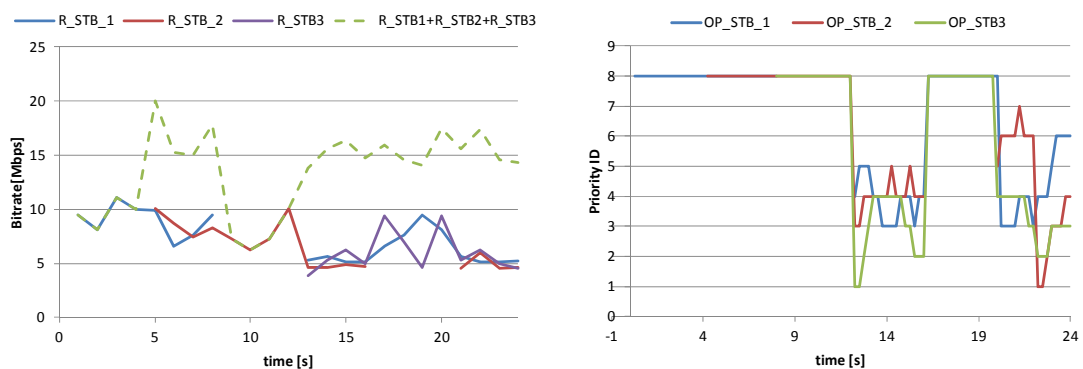

Figure 24: Bitrates and Operation points for test scenario#3

As shown in the results for the different test scenarios (see Figure 22, Figure 23 and Figure 24) the bitrates and operation points of the forwarded streams by the PDD, vary continuously to match the available bandwidth at the DSL link. However, for some of the time packet dropping is performed, while the available bandwidth is not effectively used and the forwarded data lies under the throughput limit. The reason for that is the fact that the used SVC stream is encoded with VBR and no delay for bitrate estimation of the layers is applied at the PDD, so the estimation of current characteristics of the data (bitrate of operation points) is based on the

measurements of the recent past (i.e. without look ahead). For the second version, a delay buffer will be built in the PDD allowing measuring the exact bitrate of the different operation points and forward with a higher accuracy a bitstream matching the available bandwidth of the DSL link. In this case, the estimation of future values based on the previous data shows anyway to give an acceptable solution. However, it is thought that in the second version with the described additional feature in the PDD the results will be much better.

4. Conclusions

Two preliminary versions of different PDD prototypes were carried out for different tagging approaches:

- AVC video encoding with multiple profiles,
- Scalable Video Coding with MGS SNR scalability

4.1 AVC Multi Bitrate Streams

The document describes the:

- Design of the PDD
- The implementation
- The building blocks
- The interfacing technologies (head-end, encryption, STB, middleware…)
- The test scenarios.

As for the test results of 3.1.2 above, one can see that the stream selection is quite constant. i.e. for given number of played channels, the same streams are always chosen.

This situation would change in case of VBR coding, where streams' bitrates change all the time while maintaining same quality level.

Moreover, it is assumed that with more streams per channel a better statistical multiplexing between the channels can be achieved. This will be evaluated and proven at the 2nd prototype version.

Another mean to improve results is the use of STB priority, where one STB gets better priority than other STBs on the same ADSL line. However, if priority were used in the above use-cases then nothing would change in the results due to the lack of more rates per channel.

This is also the case of remainder handling, i.e. with more streams there are better chances to use the remainder bitrate for channels with higher priority.

Since the current deliverable were using sequences developed under D4.5.1 AVC by TVN it wasn't possible to use other channels which have more streams. This should be targeted for D3.4.2 (2nd PDD prototype release), which supposed to be aligned with D4.5.2.

5. Bibliography

[1] 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*, Special Issue on Scalable Video Coding, Vol. 17, No. 9, pp. 1103-1120, September 2007.

[2] Thomas Wiegand, Ludovic Noblet, and Fabrizio Rovati: "Scalable Video Coding for IPTV Services", *IEEE TRANSACTIONS ON BROADCASTING, VOL. 55, NO. 2, JUNE 2009.*

[3] Ingo Kofler, Robert Kuschnig and Hermann Hellwagner: "Improving IPTV Services by H.264/SVC Adaptation and Traffic Control", *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB 2009)*, May 2009.

[4] S. Wenger, Y.-K. Wang, T. Schierl, and A. Eleftheriadis, "RTP Payload Format for SVC Video." Internet Draft, Nov. 2008

[5] Broadcom, System-on-Chip solution for Full-Resolution HD 3DTV supporting SVC (and MVC) http://www.broadcom.com/products/Satellite/Satellite-Set-Top-Box-Solutions/BCM7425 , (accessed April 29, 2011)

[6] Shuaib, K.; Sallabi, F, "Smoothing of Video Transmission Rates for an LTE Network", IEEE International Workshop on Selected Topics in Mobile and Wireless Computin*,* 29 November 2010

[7] Khaled Shuaib, Farag Sallabi, Liren Zhang: Smoothing and Modeling of Video Transmission Rates over a QoS Network with Limited Bandwidth Connections CoRR abs/1105.5515: (2011)

[8] Subhabrata Sen, Jennifer L. Rexford, Jayanta K. Dey, James F. Kurose, Donald F. Towsley: "Online Smoothing of Variable-Bit-Rate Streaming Video", *IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 2, NO. 1, MARCH 2000*

[9] VideoLAN, available at "http://www.videolan.org".

[10] Boost, C++ framework, http://www.boost.org/.

[11] Qt framework, http://qt.nokia.com/products.