IST R&D. FP6-Priority 2.
SPECIFIC TARGETED RESEARCH PROJECT
Project Deliverable

SUIT Doc Number	SUIT_429
Project Number	IST-4-028042
Project Acronym+Title	SUIT- Scalable, Ultra-fast and Interoperable Interactive Television
Deliverable Nature	Report
Deliverable Number	D4.5
Contractual Delivery Date	31 November 2007
Actual Delivery Date	14 December 2007
Title of Deliverable	**Playout Management**
Contributing Workpackage	WP4
Project Starting Date; Duration	01/02/2006; 27 months
Dissemination Level	PU
Author(s)	Víctor Domingo (URL), Francesc Enrich (URL) Kenneth Vermeirsch (IBBT) Michael Probst (IRT) Antonio Navarro (IT), Nuno Coelho (IT), Nelson Cabral (IT) Huseyin Oztoprak (UniS), Stewart Worrall (UniS), Stephane Villette (UniS)

Abstract

This document describes the modules of the SUIT system playout and the communication amongst them. Besides, it is also explained the role of the extractor module and the Intelligent Unit in order to perform optimization of services bitrates.

Keyword list: Bitrate, SOAP, UDP, Priority, Optimization, Intelligent Management.

Playout Management

SUIT_429

14 December 2007

Table of Contents

1 Introduction

SUIT makes use of two wireless broadband last mile networks in order to provide better quality of service and more flexible types of services. The advantages of a convergent solution (DVB-T/H + WiMAX) proposed by SUIT in comparison to any other divergent solution are:

- Supporting for return channels for interactive services as well as for conversational services like VoIP
- More robust reception for broadcasting signals in urban areas
- Supporting for broadcasting and unicasting services
- Mobility support at high speed above 150 km/h
- Supporting Quad-play (fixed and mobile)
- Intelligent routing of data (like hyperlinked video) using both networks.

Other singular features are:

- For rural areas, SUIT provides Internet services over UHF bands using DVB-T/RCT technology
- Support wide range of devices from HD TV sets to Mobile pocket units

The Intelligent management router, as mentioned above, is the main topic covered in this deliverable. Fig. 1 shows all components of the playout architecture including IP addresses. Service scenarios are illustrated and detailed in deliverable D1.5.

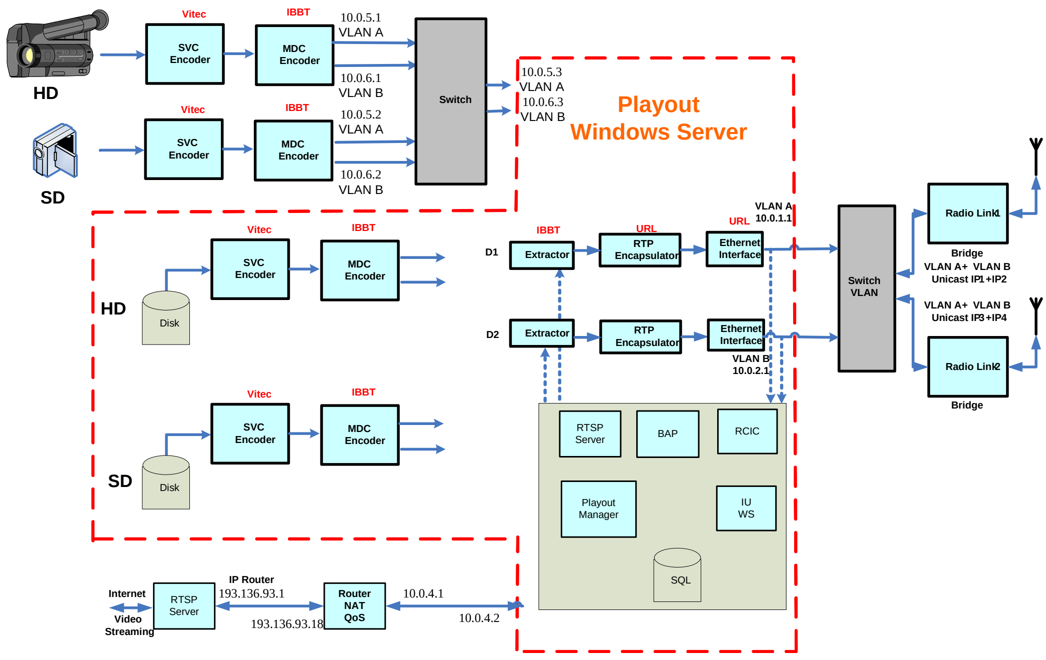

Fig. 1- Playout Architecture.

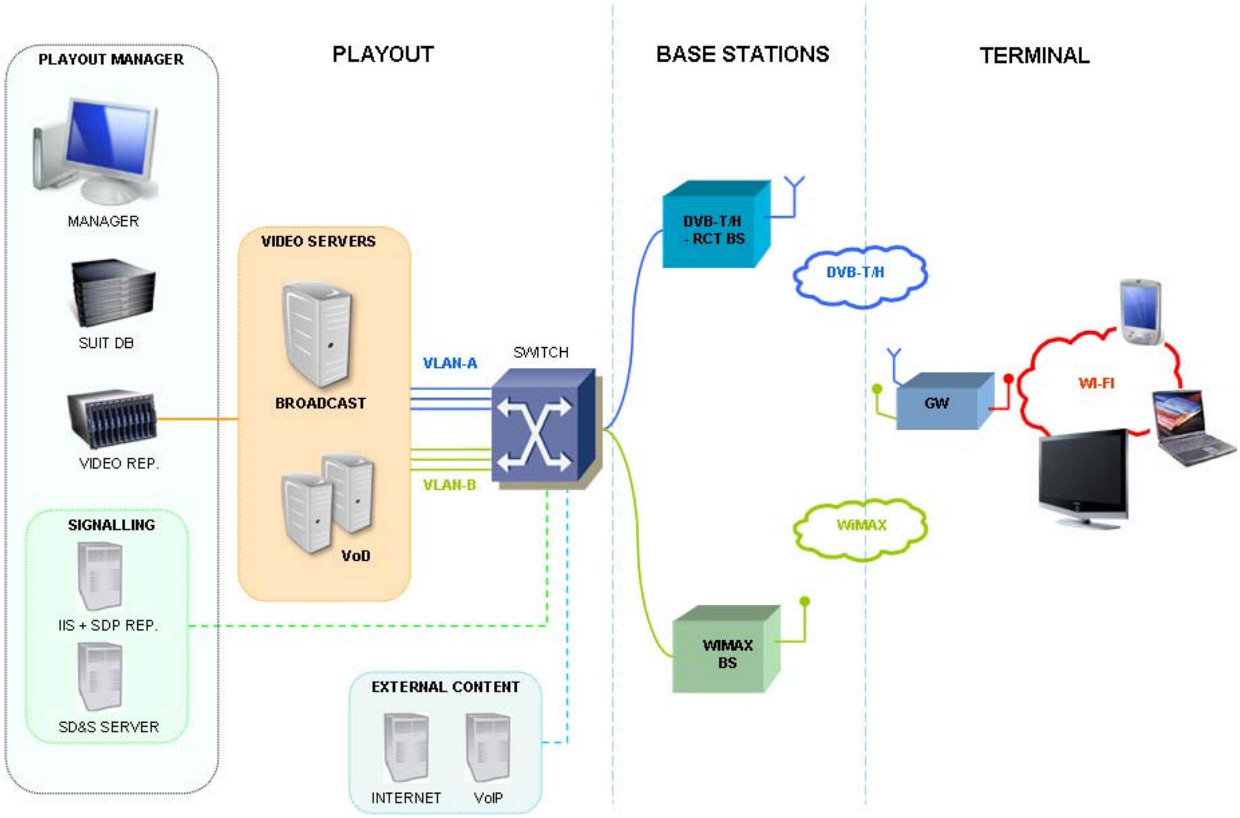

Fig. 2- SUIT Architecture.

The SUIT playout has been developed in a distributed way providing the managing of several multiplex and allowing video server load sharing. It consists of several modules, all managed at IP level: Playout Manager, Signalling, Video Servers, Platforms and Base Stations. The overlaid boxes show the linking between modules. The aim of the Playout Manager is to allow the Broadcaster Administrator to manage all the services in emission and configure all the elements in the playout. From this module all the emission signalling, platforms, contents and user connections can be monitored.

The SUIT playout delivers all the video services through the Video Servers. These elements are managed from the Playout Manager and can be deployed in a flexible way. The playout allows managing group of video servers located in different LANs. All the video servers in a group are connected to a switch that multiplexes all the input IP streams into two output IP streams, each one connected to a different base station. Each base station receives all the IP streams; these streams are multiplexed and encapsulated according the nature of the network through it would be delivered. The grouping of these elements is named “multiplex”.

The signalling of the total amount of services delivered by the playout to the user is done by the Signalling module. The SD&S[9] Server generates the information of all the available services to the user and links with the SDP[20] or RTSP[1] information needed by the terminal to receive the video service correctly.

The retrieving of the terminal information is performed by the RCIC module. This web service receives information from each terminal characteristics and notifies it to the playout. This module is also capable of obtaining information from the base stations.

All the information regarding to the playout is stored in a mysql database. This database is accessible from any module of the Playout.

In the next sections, these SUIT modules are detailed as well as all the communications between modules.

2 Playout

2.1 Playout Manager

This module is intended to manage all the modules of the SUIT playout. One of the requirements for this module is to be accessible from any PC inside the broadcaster intranet or extranet. So a web interface has been developed using .NET technologies and it has been deployed in an Internet Information Server (IIS) to make it available to all the broadcaster users. The web interface has been divided into several thematic web pages where different actions can be performed.
The Playout Manager communicates with all the playout elements through the web service Intelligent Unit.

2.1.1 Web Interface

The Playout Manager can be divided in three areas: Menu Bar (1), Action Area (2) and Log Area (3).

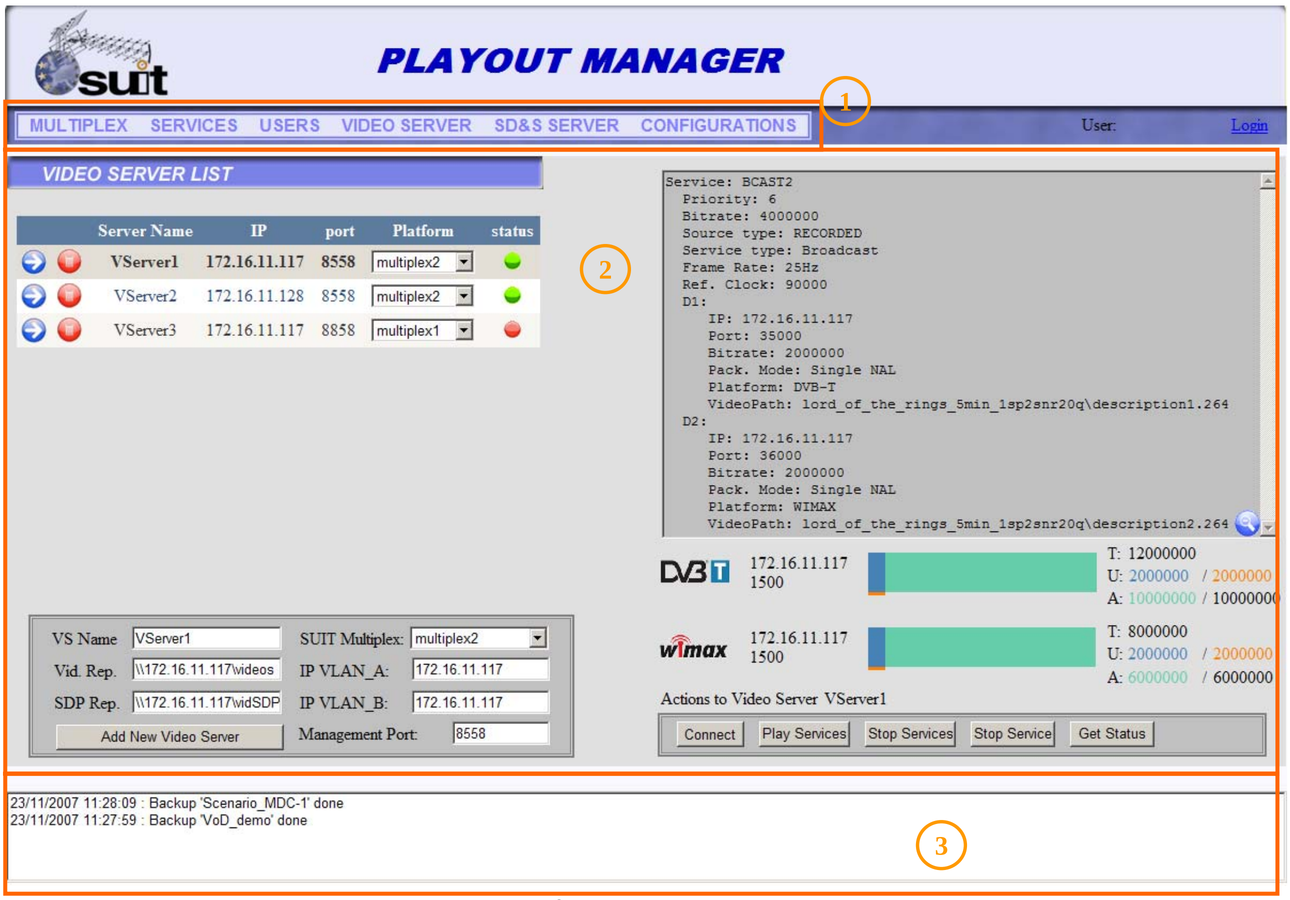

Fig. 3- Playout Manager.

From the Menu Bar (1) we can step to the different action web pages. These web pages are loaded in the Action Area (2). In the Action Area we can interact with the different elements of the web page inserting new services, performing actions to the playout elements, etc…

Finally, the information regarding to all the actions done in the action zone is shown in the Log Area (3).

2.1.2 Services Configuration

This web page allows the creation and modification of all the services in the playout and it also shows information of each service characteristics. We can distinguish three areas: Services Table (1), Descriptions information area (2), Action Toolbar (3).

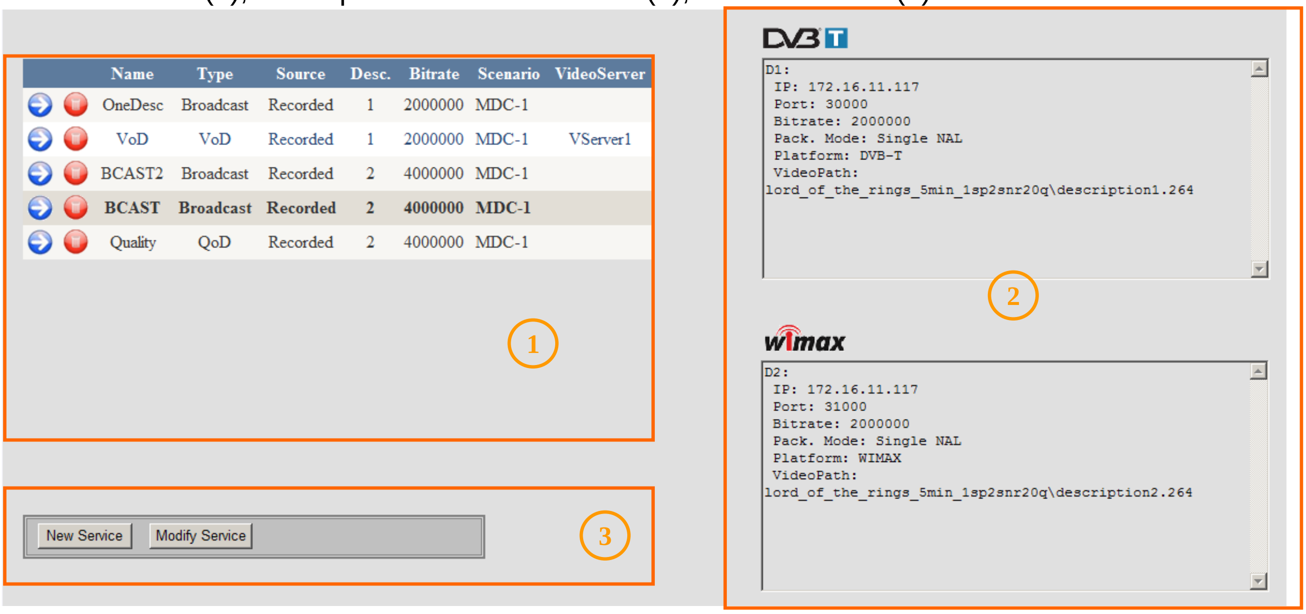

Fig. 4- Service Configuration.

- **Services table:** In this table all the available services in the playout are shown. This table shows information about the name, type, source, number of descriptions, bitrate, type of scenario and video server where the service is assigned. The button selects one service showing extended information of its descriptions in the Descriptions information zone. To delete one service we must use the button.
- **Descriptions information area:** It shows extended description information from the selected service. Here, we can see which kind of descriptions is delivered in each platform (DVB-T or WiMAX).
- **Action Toolbar:** These buttons open a wizard to create a new service or modify the selected one. When we modify a service the wizard loads the information from the service selected.

Service Wizard:

When the button **[New Service]** is pressed, the service wizard is loaded in the action area. The buttons **[Previous]** and **[Next]** allows navigation through the wizard. If some parameter is missing the wizard won't allow continuing forward.

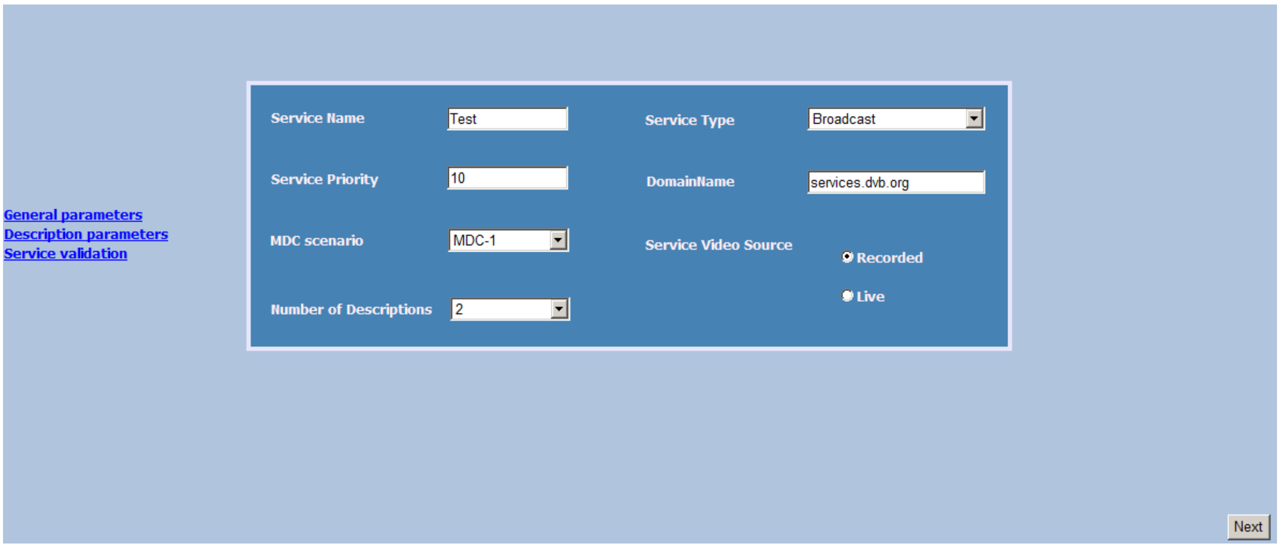

Fig. 5- Service Wizard.

- Service Name: Name that identifies the service.
- Service Priority: This value is used in the bitrate optimization algorithms to prioritize the services. Higher the value, more priority assigned to the service.
- MDC scenario: It selects the type of MDC used in that service.
- Number of description: Number of descriptions in the service.
- Service Type:
  - Broadcast
  - VoD
  - QoD
  - Hyperlinked
  - Unicast
- Domain Name: Used in the SD&S signalling.
- Video Source:
  - Recorded: Descriptions read from disk.
  - Live: Descriptions received in real-time.

When we have filled in all the parameters we press the **[Next]** button to go further. In the next wizard step we will configure the description parameters.

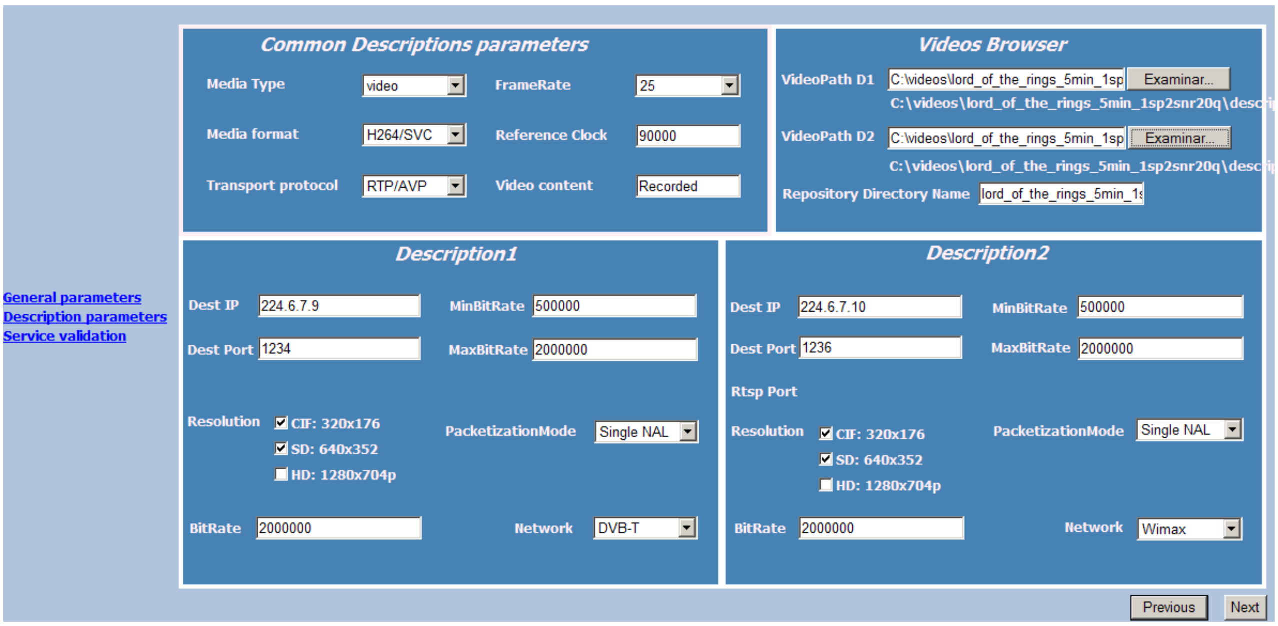

Fig. 6- Service Wizard - Description Parameters.

- Common Description Parameters:
  - Media Type
  - Media Format
  - Transport protocol
  - Frame Rate
  - Reference Clock
  - Video content
- Description:
  - Destination IP
  - Destination Port
  - Resolution
  - Packetization Mode
  - Bitrate
  - Network
- Videos Browser:
  - Videopath
  - Repository directory name

In the next step, a summary of the service configuration is shown:

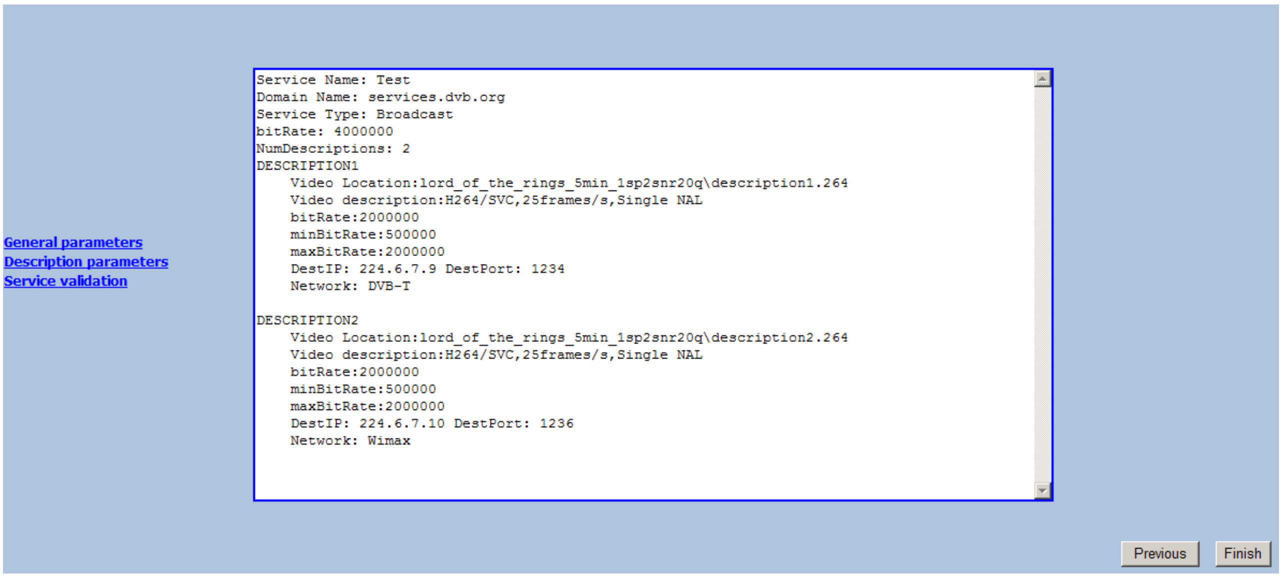

Fig. 7- Service Configuration Example.

To validate and store the new service in the SUIT database we must press **[Finish]**, otherwise we can press **[Previous]** button and change the parameters in previous steps.

2.1.3 Video Servers Configuration

The aim of this web page is to manage all the video servers of the playout. Five areas can be distinguished: **Video Server List Area (1), New Video Server (2), Available Services List (3), Platforms Bandwidth Area (4), Action ToolBar (5).**

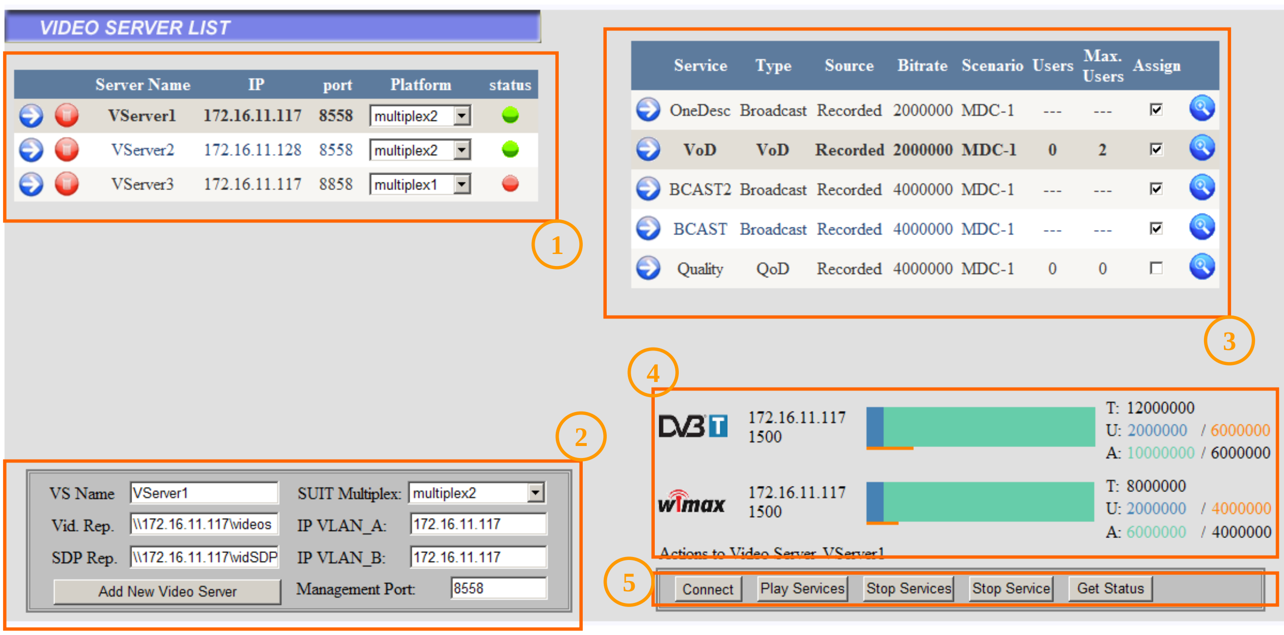

Fig. 8- Video Server List.

- **Video Server List Area:** In this table are listed all the video servers available in the playout.
  - Video Server Name
  - IP and Port
  - Platform
  - Status

The button selects one video server and it shows information of the video server in all the other areas.

- **New Video Server:** Shows information about the video server selected in the Video Server List Area and allows entering new video services to the list. Fields available:
  - Video Server Name
  - Video Repository: Path where the video files are stored.
  - SDP Repository: Path where the SDP files are stored.
  - SUIT Multiplex: Multiplex where the video server is assigned. When several video servers share the same multiplex this means that the bandwidth is shared too. This option allows load sharing between video servers to deliver all the services.
  - IP VLAN_A: IP for the first network interface of the video server.
  - IP VLAN_B: IP for the second network interface of the video server.
  - Management Port: Port to send the UPD commands to the video server.
- **Available Services List:** List showing the services that can be assigned to a one video server. When a service has been assigned to a video server, this service not appears as available to the others video servers.
  - Users: Number of Unicast or VoD users consuming this service.
  - Max. Users: Number of available users to consume this service.

If the button is pressed, extended information of the service is shown.

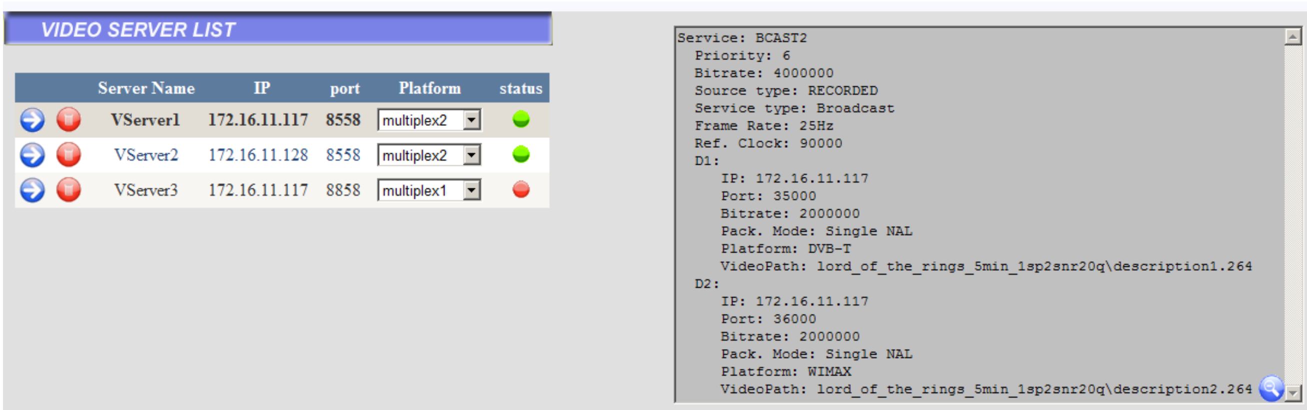

Fig. 9- Service Details.

- **Platforms Bandwidth Area:** These graphic bars give visual information of the used bitrate and available bitrate for each platform (DVB-T and WiMAX) for the multiplex of the selected video server.

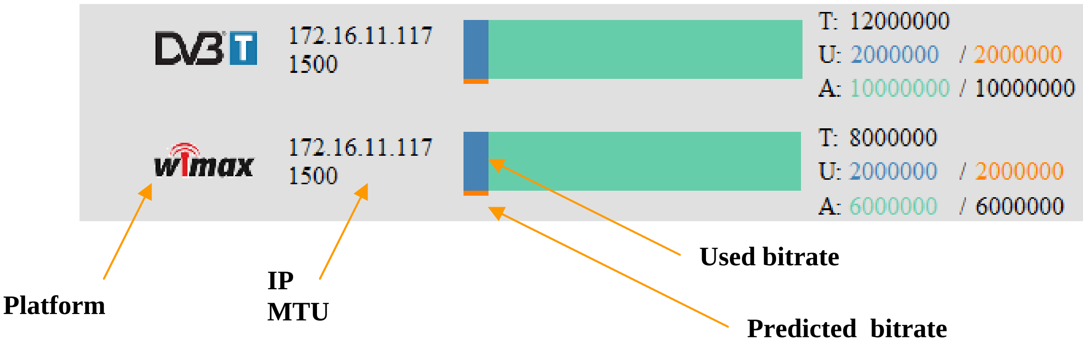

Fig. 10- Bandwidth Monitoring.

The blue values indicate the amount of bitrate already assigned into the video server.
The orange values indicate the prediction of the total amount of bitrate that will be assigned into the video server.

T: Total bitrate.
U: Used bitrate.
A: Available bitrate.

- **Action Tool Bar:** Buttons that interact with the video server selected in the Video Server List Area. The following actions are available:
  - **[Connect]:** It configures the video server with the IP for each network interface, MTU, Video and SDP repositories. The video server must be connected first in order to allow any other action.
  - **[Play Services]:** The video server starts the delivery of all the services selected in the manager.
  - **[Stop Services]:** The video server stops the delivery of all the services.
  - **[Play/Stop Service]:** starts or stops the single service selected.

- **[Get Status]:** Retrieves information of the status and all the services delivered in one video server.

2.1.4 Multiplex configuration

This web page allows the creation and managing of the multiplex. In the SUIT manager a multiplex can be defined as a bouquet of services delivered through two platforms. Here, the MTU and maximum bitrates for DVB-T and WiMAX can be defined.
When a video server is assigned to a multiplex in the Video Servers Manager, these values are taken into account limiting the bandwidth for each platform and the size of the IP packets generated in the RTP encapsulator.

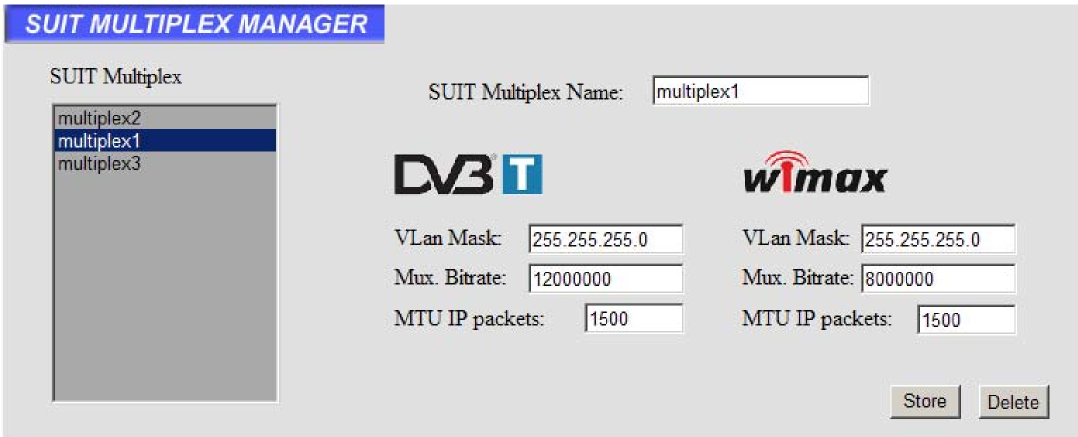

Fig. 11- Multiplex Manager.

2.1.5 User Accounting

From this web page we manage all the VoD and QoD user accounts. This page is divided into the User List Area (1) and the User Information Area (2).

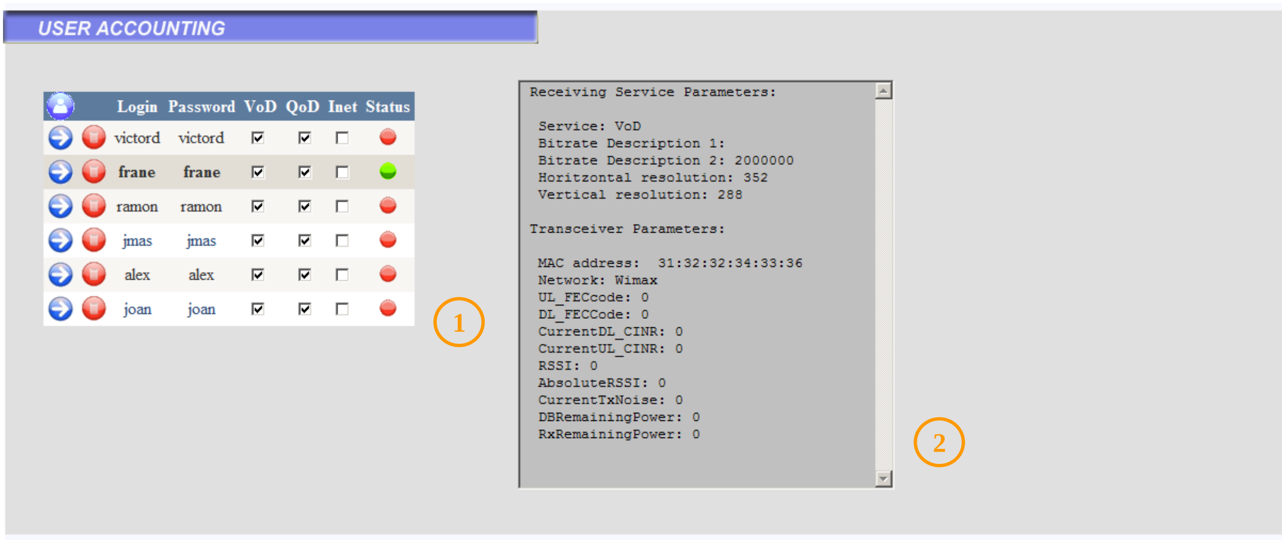

Fig. 12- User Manager.

- **User List Area:** Shows information about the user login, the service accounting and the status for VoD or Unicast services.
- **User Information Area:** Shows information about the terminal,the service delivered to the user and information from the transceiver.

To create a new user account we must press the  button and the **New User Creation Box** appears to the left as shown Fig. 13. Here, the login, password and allowed services must be entered.

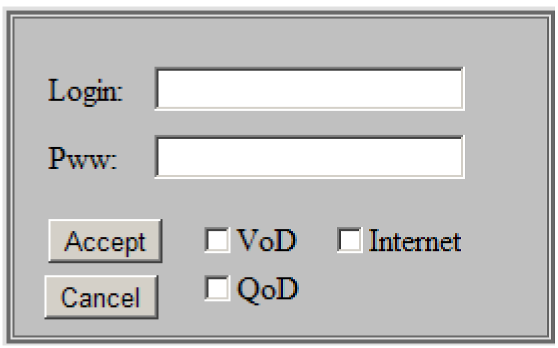

Fig. 13- New User.

To delete a user account, the button  must be pressed.

2.1.6 SD&S Server configuration

The signalling of the playout is generated in the SD&S server. This server reads the information from the SUIT MySQL database in order to generate the signalling information.

Fig. 14- SD&S Configuration.

From this web page we can establish the connection with the SD&S server and start or stop the service signalling.

- **[Setup]:** Establish connection with the SD&S server.
- **[Play SD&S]:** Starts the emission of SD&S signalling.
- **[Stop SD&S]:** Stops the emission off SD&S signalling.
- **[Update SD&S]:** Informs to the server from new changes in the database in order to update the SD&S signalling.

2.1.7 Backups managing

Another functionality of the Playout Manager is to backup configurations. This functionality is extremely useful to recreate different demos or scenarios. The interface is very easy to use. First we must indicate the path of the backup repository, where all the backups are available. Next, we must press the **[Refresh]** button to update the backup list.
To retrieve a configuration backup we must select a backup from the backups list and press the **[Retrieve]** button.
To backup a configuration we must enter the name of the backup and press the **[Backup]** button.

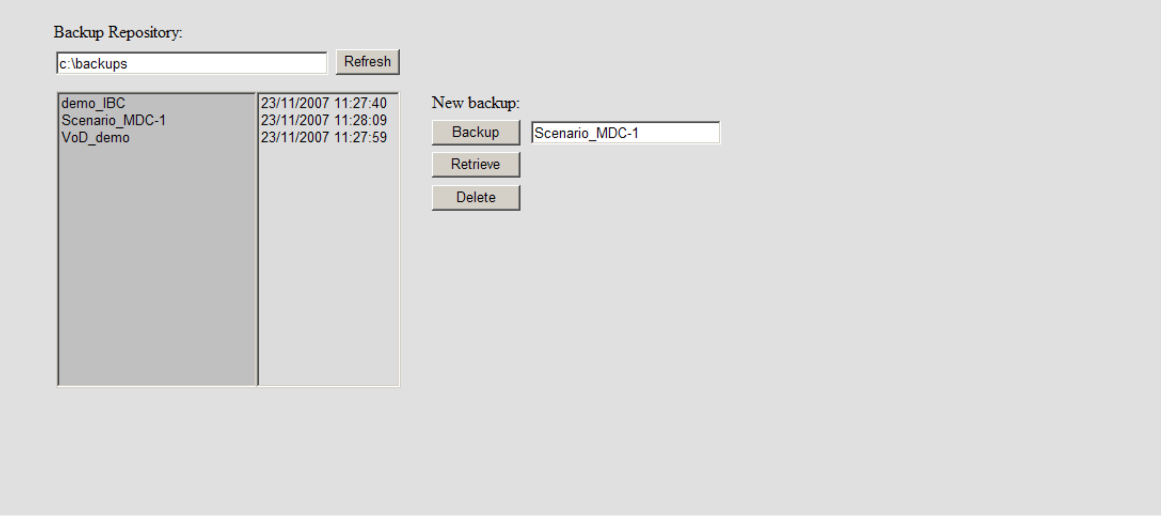

Fig. 15- Backup Managing.

Each backup configuration is stored in the repository folder inside a new folder with the name: SUIT_backup_[Name].

Ex: for the **demo_IBC** configuration showed in the picture above, the backup can be found in the folder path: **c:\backups\SUIT_backup_demo_IBC**

2.2 Intelligent Unit Web Service

2.2.1 Functionalities

This module works as a resident web service waiting for commands from the Playout Manager and requests from the Video Servers. The IU web service basically acts over the following modules:

- Control of the VideoServer
- Control of SD&S server

It also makes decisions about optimal services bitrates and decides which is the best network where to deliver additional Internet content.

2.2.2 Commands

2.2.2.1 Video Server control

The video server control is done by the playout manager through the IU. The IU converts SOAP[2] messages to UDP[8] messages sent to the video server.
The following UDP[8] commands are implemented in the IU:

- **ConnectVideoServer:** configures the video server Ethernet cards that belong to a specific platform. It also configures IP and RTP[17][19] parameters of the video server.
- **IsVideoServerConnected:** get the status of the video server
- **StopAllServices:** this command removes al sessions of the video server.
- **PlayAllServices:** some services configured at the playout manager are assigned to a specific video server and then the video server creates the RTP[17][19] sessions according the type of the services assigned.
- **PlayService:** this command is the same as PlayAllServices, but just for only one specific service.

- **StopService:** this is the same as StopAllServices, but just for only one specific service.
- **AddDestination:** this command is sent to the video server in order to start the emission of one video description to a unicast address. The session of the service is already created.
- **GetVideoServerServices:** the video server returns information about its configured services.
- **SetPacketSize:** this command allows changing the MTU size of the packets sent by the video server.
- **SetTargetBitRate:** this command is sent to the video server to set the target bitrate for each description. The extractor of each description is responsible set the specified bitrate.

The UDP[8] interface between the IU and the video server is described in Chapter 2.3.

2.2.2.2 SD&S control

The Web Service (IU) converts the Manager SOAP[2] commands into UDP[8] messages sent to the SD&S[9] server. This UDP[8] interface is described in chapter 2.4.

The following UDP commands are implemented in the IU:

- **SetupDiscoveryServer**
- **PlayDiscoveryServer**
- **StopDiscoveryServer**
- **UpdateDiscoveryServer**

2.2.2.3 Services requests

Another functionality of the Web Service is to receive information from the video servers when a user requests VoD services, QoD services, Internet, or whether is necessary to change services bitrates in order to maximize the available bitrate for each platform.

The IU implements some web methods to be called by the video servers:

- **VoDRequest:** when a user requests a VoD service using the RTSP[1] protocol, the video server calls this web method to get permission from the playout to serve the VoD service to the user.
  The IU has to authenticate the user and it has also to get the available bitrate of the Wimax platform in order to be aware if the service can be delivered. The IU will take into account the available bitrate and also the priority of this VoD service, therefore if there isn't available bitrate in the Wimax network and some video servers are broadcasting services over Wimax with lower priority than the VoD service, then the IU will try to command the extractors of the video servers in order to reduce the bitrates of broadcast services whether is possible in order to accommodate the new VoD service request. So the broadcast services bitrates will be only reduced until a minimum value that the video servers can ensure the video quality of these services.

- **QoDRequest:** when a user requests the second description of a broadcast service in order to get more service quality, an RTSP[1] request is done to the video server, so the video server calls this web method to get permission from the playout to serve the second description to the user.
  The IU has to authenticate the user and it has also to get the available bitrate of the Wimax platform in order to be aware if the description can be delivered. The IU will take into account the available bitrate and also the priority of this QoD service, therefore if there isn't available bitrate in the Wimax network and some video servers are broadcasting services over Wimax with lower priority than the QoD service, then the IU will try to command the extractors of the video servers in order to reduce the

bitrates of broadcast services whether is possible in order to accommodate the new description of the QoD service request. So the broadcast services bitrates will be only reduced until a minimum value that the video servers can ensure the video quality of these services.
If the second description is finally delivered to the user, it will be synchronized to the broadcast description sent through the DVB-T network.

- **NewServiceBitrateAdapted:** when a Video Server is delivering a unicast service like a VoD service, then the extractor will adapt the service bitrate according the terminal capabilities. It could be possible that the configured target bitrate for the VoD service does not fit all terminals requesting this service. Is for that reason that the video server has to be aware of the bitrate sent to each user in order to get more information about the amount of bitrate sent by the video server. Therefore, this web method will be called by the video server in order to inform the IU about the bitrate assigned to the video on demand session just to fit the terminal capabilities. Besides that, if the adapted bitrate is smaller than the target bitrate configured in the playout, the IU will try to reuse the available bitrate in order to maximize the broadcast services bitrates to deliver the best video quality as much as possible.

- **InternetRequest:** when a user requests internet service using the HTTP[22] protocol, the video server calls this web method to get permission from the playout to serve the internet service to the user. The IU has to authenticate the user and it has also to get the available bitrate of the Wimax and DVB-T platform in order to be aware if the service can be delivered. The IU will take into account the available bitrate of each network, the network conditions that modify the available bandwidth and also the priority of this internet service. Therefore, the IU has to choose the best network to delivers internet according the available bitrate. If there isn't available bitrate neither in Wimax nor in DVB-T networks and some video servers are broadcasting services over Wimax or DVB-T with lower priority than the Internet service, then the IU will try to command the extractors of the video servers in order to reduce the bitrates of broadcast services whether is possible in order to accommodate the internet service request.

- **DisconnectUserUnicast:** this web method will be called by the video server a user receiving a QoD service logouts the session. The IU has to increase the available bitrate of the Wimax platform with the amount of bitrate used by the QoD user. The IU has also to maximize the broadcasting services whenever is possible, following the described service policy priority.

- **DisconnectUserVoD:** this web method is like DisconnectUserUnicast but for a VoD service.

The algorithms of the IU about the decisions to be taken in certain scenarios have been described in deliverable 4.3.

2.3 Video Server

This module is intended to stream data from a repository over RTP/RTCP[16,17,18,19] protocols. It has been developed using C++ and Live555 libraries and supports currently streaming of Broadcast, VoD and QoD services.

2.3.1 Interfaces

Server has two different interfaces for Video Server-Playout Manager connection. They are used for commands and result/error messages exchange between them.

2.3.1.1 UDP interface

Video Server has a dedicated thread that implements an UDP[8] server to receive the different commands explained above in 2.2.2.1.
Its only function is to receive commands, report to streamer about these, and then return a response/error message to inform the Playout Manager about the results of the action requested.

Control commands

This section defines the syntax of the Video Server control commands. All message definitions are in network byte order, i.e. most significant byte first (MSBF).

2.3.1.1.1 Setup Command

This command is sent by IU to configure some parameters of Video Server before starting to stream services. It contains the address of the Video Server, DVB and WiMAX interfaces and Maximum Transfer Unit over these networks. It also specifies paths where Video Server can find video repository and where it has to create SDP[20] files to watch broadcast services. Finally it contains a list of users that have access to on demand services.

	Length (byte)	Value	Comment
Command ID	1	0x01	
Command length	2		Total command length not counting first three bytes
Web Service IP length	1		Length of the string with IU address
Web Service IP			String with IU IPv4 Address
Port Web Service	2		Port for the response message to IU
DVB IP length	1		Length of the string with DVB interface's address
DVB IP			String with DVB interface's IPv4 address
DVB MTU	2		DVB interface's MTU
WiMAX IP length	1		Length of the string with WiMAX interface address
WiMAX IP			String with WiMAX interface IPv4 address
WiMAX MTU	2		WiMAX interface MTU
Video repository length	1		Length of video repository
Video repository			Video repository's path
SDP repository length	1		Length of SDP repository path
SDP repository			SDP repository path
numUsers	1		Number of Users initiated for on demand services
User 1 Length	1		Length of User 1 Length
User 1 Name			First user's name
…			

2.3.1.1.2 Status

Status command is sent periodically by IU to monitor services active on Video Server. Server returns in a response message, a list of parameters from all services active.

	Length (byte)	Value	Comment
Command ID	1	0x02	
Command length	2		Total command length not counting first three bytes

2.3.1.1.3 InitServices Command

Command sent to server to initiate a number of services. It provides a number of services to add and then a list of services parameters. Services are sent one after the other from first parameter Service Name to Description BitRate. If one service contains more than one description they are sent continuously too. Descriptions specification comprises ten parameters from packetization mode to description bit rate parameters.

	Length (byte)	Value	Comment
Command ID	1	0x10	
Command length	2		Total command length not counting first three bytes
NumServices	1		Number of services initiated
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
BitRate	4		Service's bit rate
Type	1		Service's Type [Broadcast:1,Video on Demand:2, Quality on Demand:3]
Video source	1		Recorded [0] / Live [1]
FrameRate	1		Service's frame rate [fps]
ReferenceClock	1		Service's Reference clock [kHz]
RTSPPort	2		RTSP connection port (ONLY in VoD Services)
Num. descriptions	1		Number of descriptions of the service
Packetization mode	1		Non-Interleaved [0] / Interleaved [1]
Destination IP length	1		Length of the string with destination address
Destination IP			String with destination IPv4 address
Destination port	2		Destination port
Path length	1		Length of Path
Path			Path of descriptions relative to video repository
RTSPString length	1		Length of RTSPString
RTSPString			RTSP URI used for VoD Request
FUA	1		Fragmentation Unit deactivated [0] / activated [1]
Description BitRate	4		Description's bit rate
…			

2.3.1.1.4 addService Command

AddService is sent to add one service to Video Server and it is almost identically to initServices command. The only difference between them is that this one doesn't include the number of services and the list contains only parameters for one service.

	Length (byte)	Value	Comment
Command ID	1	0x20	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
BitRate	4		Service's bit rate
Type	1		Service's Type [Broadcast:1,Video on Demand:2, Quality on Demand:3]
Video source	1		Recorded [0] / Live [1]
FrameRate	1		Service's frame rate [fps]
ReferenceClock	1		Service's Reference clock [kHz]
RTSPPort	2		RTSP connection port (ONLY in VoD Services)
Num. descriptions	1		Number of descriptions of the service
Packetization mode	1		Non-Interleaved [0] / Interleaved [1]
Destination IP length	1		Length of the string with destination address
Destination IP			String with destination IPv4 address
Destination port	2		Destination port
Path length	1		Length of Path
Path			Path of descriptions relative to video repository
RTSPString length	1		Length of RTSPString
RTSPString			RTSP URI used for VoD Request
FUA	1		Fragmentation Unit deactivated [0] / activated [1]

Description BitRate	4		Descriptions bit rate

2.3.1.1.5 deleteService Command

Command sent by IU to delete one specified active service. It contains the name of service to delete and type of this one.

	Length (byte)	Value	Comment
Command ID	1	0x30	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Type	1		Type of Service

2.3.1.1.6 deleteAllServices Command

Similar to command just above it doesn't set any service name or type, deleting all services active on Video Server.

	Length (byte)	Value	Comment
Command ID	1	0x31	
Command Length	2	0	

2.3.1.1.7 addDestination Command

It is a Quality on Demand specific command. When Server receives it starts to send second description of a QoD Service. Main second description's parameters of the QoD services are included on service initialization but this command contains the name of requested service and address and port of terminal that demands it.

	Length (byte)	Value	Comment
Command ID	1	0x50	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Destination IP length	1		Length of the string with destination address
Destination IP			String with destination IPv4 address
Destination port	2		Destination port

2.3.1.1.8 stopFUA Command

It sends name, type and description's number of a service that has to be sent without using fragmentation of NAL Units by the Server. From then this command is received fragmentation function is carried out by the network.

	Length (byte)	Value	Comment
Command ID	1	0x60	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Type	1		Type of Service
Description Number	1		Number of description

2.3.1.1.9 startFUA Command

Reverse to stopFUA, when server receives this command it starts to use *Fragmentation Unit Type A* and dividing NAL Units in parts. These ones size are equal to MTU's value for network where they are sent.

	Length (byte)	Value	Comment
Command ID	1	0x61	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Type	1		Type of Service
Description Number	1		Number of description

2.3.1.1.10 changePacketSize Command

Command that specifies packets' size for a particular description of a service. When it is received server starts to fragment NAL Units of this description in established size parts.

	Length (byte)	Value	Comment
Command ID	1	0x62	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Type	1		Type of Service
Description Number	1		Number of description
Packet Size	2		Size of the RTP Packet

2.3.1.1.11 setTargetBitRateOneDescription Command

Command sent by IU to balance load of the network by re-adjust bit rates of a service's description. The Server receives it and calls the corresponding Extractor Interface function to modify its bit rate. When it specifies a VoD service, it also includes the user of the video that the bit rate will be changed.

	Length (byte)	Value	Comment
Command ID	1	0x63	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
Description Number	1		Number of description
BitRate	4		Description new bit rate
User length	1		Length of user name
User name			User name

2.3.1.1.12 setTargetBitRate Command

Likely last command, setTargetBitRate re-adjust bit rate of a service, but in this case, it modifies both descriptions. It doesn't matter if the service requested has only one description, so it will change only the first one, but command should include description bit rate bytes.

	Length (byte)	Value	Comment
Command ID	1	0x64	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name

Description 1 BitRate	4		Description 1 new bit rate
Description 2 BitRate	4		Description 2 new bit rate
User length	1		Length of user name
User name			User name

2.3.1.1.13 getTargetBitRate Command

Command used by IU to monitor bit rate of the different services active without using status command that involves more information exchange maybe not needed at this point. It includes a service's name and a user for video on demand services.

	Length (byte)	Value	Comment
Command ID	1	0x66	
Command length	2		Total command length not counting first three bytes
Service Name length	1		Length of the string with Service Name
Service Name			Service's Name
User length	1		Length of user name
User name			User name

2.3.1.1.14 Response Message

Response messages are sent in reverse direction than others specified above. These messages include command ID that response belongs, error value to action requested and, in some cases, some parameters to better identify this error. Also in return to status and getTargetBitRate commands return other parameters requested following Message length field. Status' response includes a list of services active on Video Server just like the structure of InitServices command. Other way, getTargetBitRate command's response returns the bit rate requested expressed in four bytes.

	Length (byte)	Value	Comment
Command ID	1		
Error	1		Error code
Message length	2		Total message length not counting first four bytes
Service Name length	1		Length of Service Name
Service Name			In response message to InitServices command, name of service that causes an error

<u>List of error codes</u>

Value	Comment
0x00	No error
0x01	DVB IP Address requested not accessible
0x02	WiMAX IP Address requested not accessible
0x11	Service addition request error: Description 1 port and address already taken
0x12	Service addition request error: Description 2 port and address already taken
0x1E	Service addition request error : Reached maximum load of services allowed (MAX_SERVICES)
0x1F	Service addition request error : Service name already taken
0x21	Service addition request error: Description 1 file not found
0x22	Service addition request error: Description 2 file not found
0x31	Service delete error: Error deleting description 1
0x32	Service delete error: Error deleting description 2
0x3F	Service delete error: Service Name not found
0xFF	Message transmission errors (length specified in command and received bytes don't match)

2.3.1.2 <u>SOAP interface</u>

UDP[8] commands are used to control Video Server from Manager, but there is another interface connection between them for a reverse notification. The UDP[8] messages flow isn't always Manager to Server, because there are response messages from Server to Manager, but it is needed a different connection to act directly on the IU web service. For that case it is used SOAP[2] interface, invoking from the Video Server different web-methods of IU. These ones are specified above in 2.2.2.3 of this deliverable.

2.3.2 Functionalities

One of its main requirements is the possibility to choose from over two operation modes. On first one, Video Server acts as a stand-alone application that initiates services specified in a *Commands.ini* file. On the other one, the application is controlled by Playout Manager, which can start, stop and modify services. On second mode it also implements user authentication to VoD and QoD services via Manager Database.
For both modes Server has the function to create corresponding SDP[20] files for each streamed broadcasting services and handle RTSP[1] requests for On Demand services.

2.3.3 Setup

Different working modes can be configured by *UDPServer.ini* using *webServiceAuthentication* parameter. A zero value indicates stand-alone mode, and one a Playout Manager/IU control. This file should also add *UDPaddress*, *UDPport* and *RTSPBroadcastPort* parameters. *UDPaddress* should be set to computer address where server is running. On a computer with various network interfaces (usual case on Video Server), it is also used to choose in what of them will be established the UDP server. *UDPport* indicates the port where Server will receive UDP commands. *RTSPBroadcastPort* show the port where will be received RTSP[1] request for QoD services. Name of this parameter is because of Server internally has some RTSP[1] Servers, one that manages broadcast and QoD services and others for VoD services.

On stand-alone mode, Video Server uses another file, *Commands.ini,* to set up extra information for his configuration and read the parameters of different services that user wants to start.

***Commands.ini* parameters:**

[Setup Command]

- **videoPath:** Repository video path.
- **SDPPath:** SDP repository path.
- **interface1_IP:** IP address network card for VLAN A.
- **interface1_MTU:** Maximum packet size for VLAN A.
- **interface2_IP:** IP address network card for VLAN B.
- **interface2_MTU:** Maximum packet size for VLAN B.
- **numUsers:** Num users to add (0:No authentication in stand-alone mode)

[InitServices Command]

- **numServices**: number of services to be delivered.

[Service N]

- **name:** Service name
- **bitRate:** Estimated service bitrate[bits/s] (not used)
- **type:** Number of the service type.
    - Broadcast [1]
    - VideoOnDemand [2]
    - QualityOnDemand [3]

- **videoSource:** Indicates if the video source is recorded or live (not used in this version)
- **frameRate:** Service frame rate [frames/second] (default 25)
- **referenceClock:** reference clock [kHz] (default 90)
- **RTSPPort:** Port to RTSP Connection (only at Video On Demand Services)
- **numDescriptions:** Number of descriptions in the service

[Description N]

- **packetizationMode:** Packetitzation type. [0] Non-interleaved
- **destinationIP:** Destination IP (not used in on demand services)
- **destinationPort:** Destination port (not used in on demand services)
- **path:** video name or relative path inside the videos' repository.
- **platform:** interface network card to deliver the description
  - interface VLAN A [0]
  - interface VLAN B [1]
- **RTSPString:** RTSP URI used in on demand services (not used in broadcast services)
- **FUA:** Nal units fragmentation type A.
  - The fragmentation is done at network level [0]
  - To fragment Nal units to the MTU of the VLAN [1]
- **descriptionBitRate:** description's estimated bitrate.

How to:

In order to use the SUIT Video Server Live application you must have two repository directories created. In our example we have the repositories **C:\videos** and **C:\vidsdp** and two networks interfaces with the **IP-A: 172.16.11.117** and **IP-B: 172.16.11.118**.

In the SDP[20] repository, the video server will store all the active services multiple-description and single-description SDP[20] files. The videos that we want to deliver must be stored inside the video repository and inside a directory with the name of the service with description1.264 and description2.264 names as shown in Fig. 16.

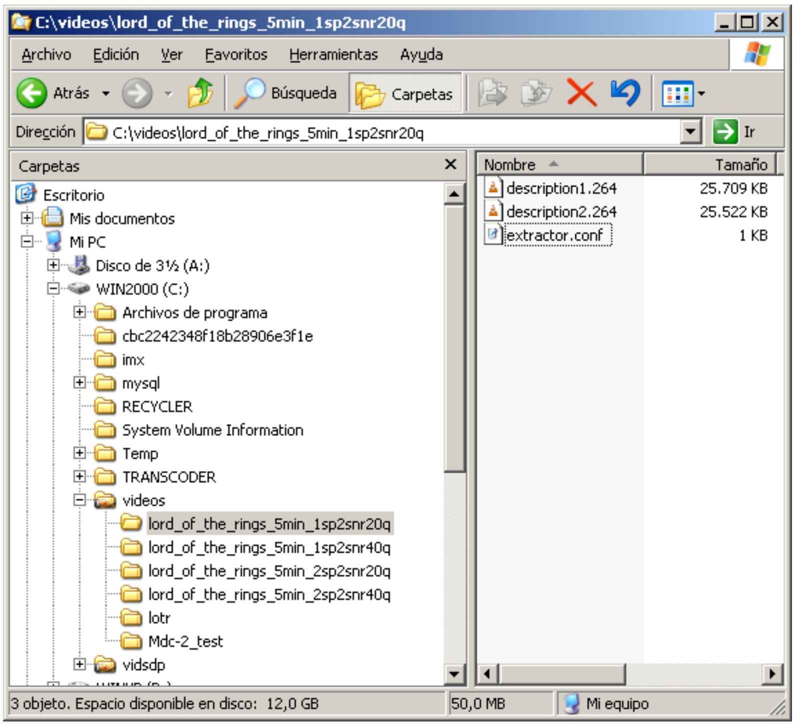

Fig. 16- Repositories.

2.3.4 Services Configuration

2.3.4.1 BroadcastServices

Broadcast Services can use one or two descriptions. Destination address IP for these ones can be set with a destination broadcast, unicast or multicast address, although if it is the same unicast address of the server, the video will not be streamed. It is caused because of Live555 implementation, what usually sends and receives one session's RTP[17][19] data using the same port number, so if the video is sent one port of Video Server, it can't receive the information on the same port. It is not an important obstacle, so Video Server will not usually be the client of services that streams, but in this special case multicast or broadcast address should be used. Client can use multiple or single-description SDP[20] files created in repository to watch these services.

2.3.4.2 VoD Services

Video on Demand Services remain inactive when they are initiated until a client reclaim one of them by calling server with a RTSP[1] request. When Video Server receives it, search service demanded and if it is found and sends a SOAP[2] message to IU, invoking VoDRequest web-method. IU will return an access confirmation or will refuse this user for service requested. If user has permission, server start RTSP[1] setup negotiation to establish the session and then terminal will start to receive video description. Also at this point, IU will send setTargetBitRate commands to Video Server to re-adjust bit rates to load of the network. When video is stopped by the user, Server Video receives a RTSP[1] Teardown message and informs IU with DisconnectUserVoD command that this user has closed his session and it has stopped transmission RTP/RTCP[16][17][18][19] packets to him.

2.3.4.3 QoD Services

Quality on Demand Services are formed by two descriptions. First one is sent when Manager initiates the service and the other one remains inactive until a client makes a request for this service using a RTSP[1] request. When it is received, server starts to stream second one to the address of user that reclaimed it. The process for authentication of this user to that service is the same used by VoD services but invoking QoDRequest instead VoD one. First description, as broadcast descriptions, can be set with a broadcast, unicast and multicast address with the same fault explained in this case.

2.3.5 Extractor

2.3.5.1 Interfaces

The Bitstream Extractor is designed as a static library, to be linked against by the playout video servers. The Extractor provides the following interfaces:

i. Front-end interface to the MDC encoder. In previous versions of the extractor, the MDC-1 description generator was built into the extractor; in the latest versions, the MDC-1 generator has been integrated into the encoder itself, in order to harmonize the MDC-1 and MDC-2 scenarios.

ii. Back-end interface to the rest of the playout (more specifically, the video servers). This interface consists of (a) a data path which delivers the extracted bit stream on a frame-per-frame basis, and (b) a control and configuration interface. The control and configuration interface includes the following functionalities:

   a. - Start-up configuration of the extractor: path to the two input descriptions. This configuration used to be in a separate extractor configuration file (extractor.conf); in order to centralize the playout configuration, this configuration file has been eliminated. The extractors are now configured at run-time by the supervising video servers.
   b. - Adaptation control interface, to allow the supervising video servers to set a target bit rate and the target spatial resolution, and to retrieve the actual realized bit rate.

2.3.5.2 Bit Rate Adaptation Process

The extractor incorporates support for the spatial and SNR scalability features offered by the SUIT codec platform. Network Abstraction Layer (NAL) units produced by the spatial extension are treated as follows.

i. Non-scalable (base layer) and non-VCL (SPS, PPS, SEI) NALUs are always extracted and sent to the output, even if doing so would cause the target bit rate to be exceeded for the duration of that frame.

ii. Base quality (non-FGS) NALUs are either extracted or dropped entirely, depending on the ID of the target spatial layer and on the target bit rate that has been set. Spatial layers (higher resolutions) that exceed the target spatial layer ID are pruned in their entirety, as are spatial layers that cannot be sent due to bandwidth shortage.

iii. FGS NALUs are sent as the target bit rate allows. Additionally, they are cut at byte granularity in order to make optimal use of the target bit rate that has been allocated to the extractor.

iv. A next spatial layer is sent only after all FGS NALUs for the current layer have been sent (and if there is still bandwidth available).

v. NALUs in an access unit are assumed to be in the correct order (non-VCL, base layer, 1st spatial enhancement layer, its FGS refinements, 2nd spatial enhancement layer, etc.).

The above behaviour allows for the actual bit rate to exceed the targeted bit rate for brief periods of time. We motivate this behaviour by remarking that this only happens when sending the most important data (from a decoder point of view), such as intra coded pictures and high-level bit stream information. As can be seen in the graph below, showing a stepped target bit rate and the actual response of the bitstream extractor, the target bit rate overstepping is very localized in time. It is expected that in a production environment the multiplexing of several programmes will smoothen these bandwidth peaks to a large extent. If necessary, the target bit rate could be set somewhat lower than the actual available bandwidth in order to compensate for this effect.

Note that the graph below shows the actual bit rate as calculated on an Access Unit (AU) basis. If the measured bit rate were to be integrated over a somewhat larger timeframe (e.g. 5~10 AU's), the actual bit rate curve would be much smoother.

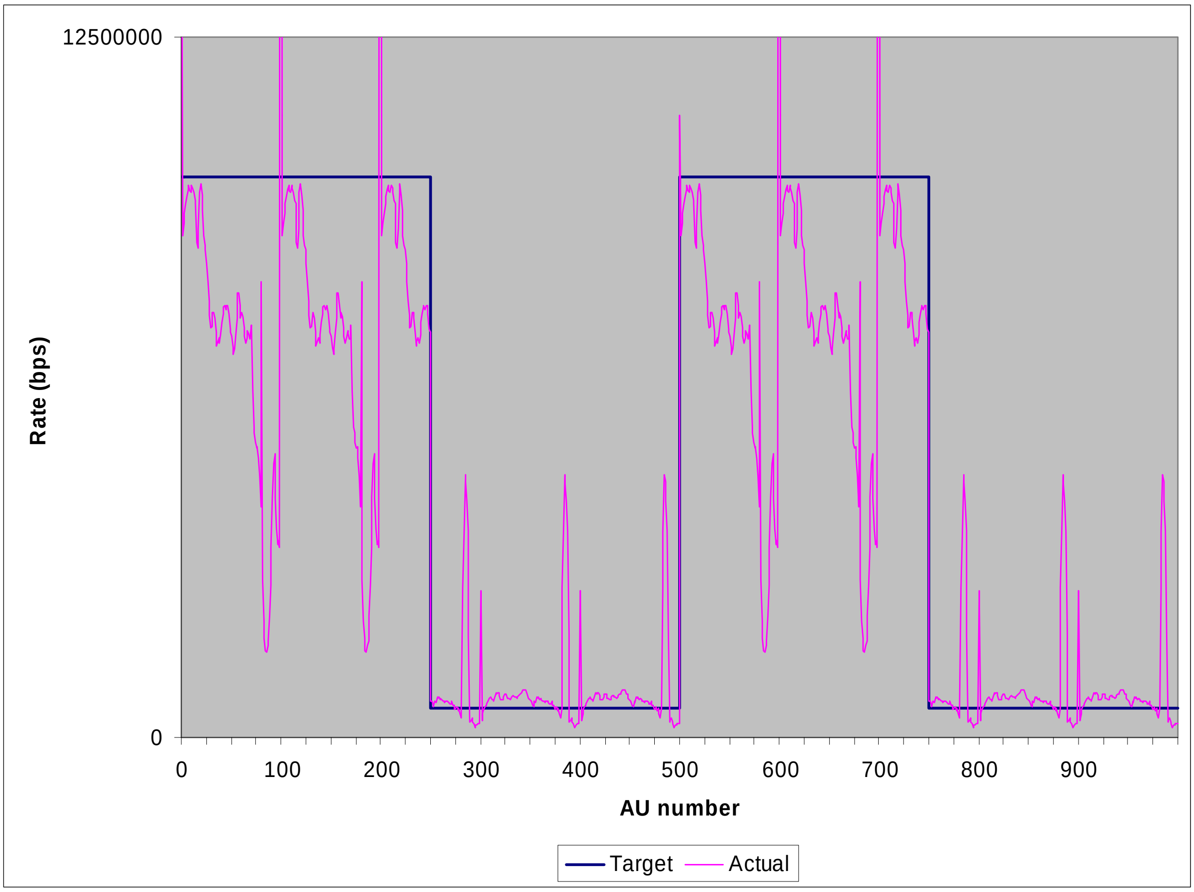

Fig. 17- Bit-rate.

The supplied Extractor demo program has been extended with the functionality to generate comma-separated value (CSV) files which contain, for each Access Unit (AU), the target and actual bit rate. This allows graphs such as the example above to be generated.

2.4 SD&S Server

The SD&S[9] server is used to announce SUIT services to SUIT client compliant to the DVB IPTV standard. Due to the fact that phase 1 of DVB-IPTV[9] supports only services based on the DVB transport stream, SUIT extends the standard to allow the use of different services. These types of services are in the scope of the upcoming Phase 2 of DVB-IPTV[9].

The SD&S server has two interfaces, one to the playout management in order to get configured and the other one is the service announcement to the SUIT clients. The former one is described in the next section, the latter in section 2.4.2.

2.4.1 Interfaces

The SD&S server can be controlled by the playout management via network (UDP[8]) messages. At the moment 4 different messages are defined – setup, start, stop, and update. The SD&S sends a short response message to each command containing an error code and message.

2.4.1.1 Control commands

This section defines the syntax of the SD&S server control commands. All message definitions are in network byte order, i.e. most significant byte first (MSBF).

2.4.1.1.1 Setup Command

The SETUP command is send by the playout management to set the address for response messages.

	Length (byte)	Value	Comment
Command ID	1	0x01	
Command Length	2	6	
IP Web Service	4		the raw IP address in network byte order
Port Web Service	2		Port for the response message

2.4.1.1.2 StartServer Command

The START command has to be sent in case the SD&S server is running in multicast mode. It starts the multicast of SD&S records encoded in DVB-STP sections. Note: In case of SD&S unicast this command is ignored by the server.

	Length (byte)	Value	Comment
Command ID	1	0x02	
Command Length	2	0	

2.4.1.1.3 StopServer Command

The STOP command stops the multicast of DVB-STP sections.
Note: In case of unicast this command is ignored by the server.

	Length (byte)	Value	Comment
Command ID	1	0x03	
Command Length	2	0	

2.4.1.1.4 UpdateServices Command

The UPDATE command triggers the SD&S server to integrate updates from the service information in the database to the playout.

	Length (byte)	Value	Comment
Command ID	1	0x04	
Command Length	2	0	

2.4.1.2 Response message

The response message is send after the command from the playout management has been executed.

	Length (byte)	Value	Comment
Command ID	1		The ID of the command for this response
Error Code	1		
Message Length	2	n	
Message	n		An ASCII coded string

Error code:

Value	Description
0x00	OK, the execution of the command succeeded.

0xFF	Error, see message for details

2.4.1.3 Server Command Line Interface

The SD&S server is implemented as a service for the Windows operating system. It is started automatically during start-up of Windows. The service can be controlled via the Windows System Control Panel.
A simple command line interface allows the service to get installed, uninstalled, started, and stopped. Call the executable with one of the following parameters:

- -Install
- -Uninstall
- -Start
- -Stop

2.4.1.4 Server Configuration

A configuration file for the server can be found in the program directory. The path to the file is cfg/ipi2.cfg (relative to the program directory). The configuration contains a set of parameters to control the playout. A detailed explanation for every parameter can be found inside the configuration file.

2.4.2 Extended Functionalities

The SD&S server is compliant to the DVB standard ETSI TS 102034. However, SUIT kind of video services are not covered by Phase 1 of DVB-IPTV. Therefore SUIT extended SD&S to cope with all the SUIT requirements, namely scalable video services. A detailed description of these additions can be found in SUIT Deliverable D4.2.

3 Return Channel

3.1.1 RCIC

3.1.1.1 Overview and functioning of the RCIC

The RCIC (Return Channel Information Collector) is part of the SUIT playout system. It acts as the glue logic for control information between the wireless base stations, the playout database, the video servers, and the terminals. It is a Java application that performs two different tasks:

1. Poll the WiMAX and DVB base stations at regular intervals and request link quality information pertaining to the different user terminals. Polling of the base stations takes place through a proprietary API based on message passing on top of the UDP[8] protocol;
2. Listen for MPEG-21[21] messages from the terminals. These are transferred using the TCP client/server model.

The received information is then parsed:

1. The base station information reply messages are coded in a flat raw data format and parsing is quite straightforward. This information includes a number of parameters pertaining to the MAC layer. One field of special importance is the FEC and Modulation code, which specifies the modulation scheme and FEC scheme currently in use. From these it is possible to calculate the maximum throughput, using a look-up table. A set of BST parameters is available for each active unicast User Terminal;
2. The MPEG-21[21] messages (which are structured XML conforming to Schema definitions) from the terminals are converted to an internal tree representation by a purpose-specific MPEG-21[21] DIA XML parser, and the fields of interest are then extracted from the internal tree.

The retrieved information is then sent to two distinct destinations:

1. All information is stored in the central Playout database, which is a MySQL RDBMS. From there it can be consulted by any Playout sub-entity for a variety of purposes;
2. Additionally, the terminal screen resolution and available bandwidth information is sent to the video server using UDP[8] messages; the video server then forwards this information to the extractors. This 'push model' eliminates the extra delay that would be introduced by having the video server poll the database for changes.

3.1.1.2 The MPEG-21 client software

To complete the MPEG-21[21] feedback chain a terminal-side client module is also provided. The MPEG-21[21] terminal functionality consists of a function that can be called to transmit an MPEG-21[21] message to the playout. This source code is designed to be integrated into the terminal session management software; however, if necessary it could be used in a stand-alone manner.

In order to function correctly, the MPEG-21[21] client needs to be supplied with the IP address and port of the playout machine where the RCIC is running. These parameters could be hard coded or could for example be supplied by the RTSP[21] or SD&S client operated on the terminal. In this case, it would be recommended to run the RCIC on the same machine as the RTSP[1] SD&S server.

3.1.1.3 Configuration of the RCIC

Reconfigurability has received special focus during implementation of the RCIC. Most of its parameters can be set in the human-readable configuration file, conf/rcic.properties . This file will be read at program startup time. The configuration options include:

IP addresses and ports for the IPC communication with the base stations;
Access parameters to the SUIT MySQL database;
IP address and port for sending the UDP messages to the video server;
The port to listen on for MPEG-21[21] messages from the terminal.

Below an example RCIC.properties is shown:

```
# Where to reach the base stations for UDP-based IPC.
rcic.basestation.dvb.host=10.0.10.1
rcic.basestation.dvb.port=50001
rcic.basestation.dvb.receivehost=127.0.0.1
rcic.basestation.dvb.receiveport=50000
rcic.basestation.wimax.host=10.0.20.1
rcic.basestation.wimax.port=51001
rcic.basestation.wimax.receivehost=127.0.0.2
rcic.basestation.wimax.receiveport=51000

# How to reach the SUIT database.
rcic.db.host=localhost
rcic.db.port=3306
rcic.db.database=suit
rcic.db.user=suituser
rcic.db.passwd=suitpassphrase

# Where to send the UDP messages for the video server to.
rcic.videoserver.host=localhost
rcic.videoserver.port=1050

# What port to listen on for MPEG-21 updates from UTs.
rcic.mpeg21server.listenport=20000
```

Additionally, the Log4j logging backend used by the Jakarta Commons Logging package can be configured through the conf/log4j.properties file. For testing and integration it is recommended to use the supplied file, which prints all log messages to standard output.

3.1.1.4 Running the RCIC

In the directory containing RCIC.jar, execute the command "java -jar RCIC.jar".
A "Java 5" (Java 1.5) compatible JVM is required.

If running the RCIC from the class tree (instead of using RCIC.jar), note that the .properties files need to be on the classpath in order to be read on startup. This remark also holds for the required libraries (which are all supplied with in the rcic/lib directory).

4 Overall Communication between modules

4.1 *Communication between manager-IU-VideoServer/SD&S Server*

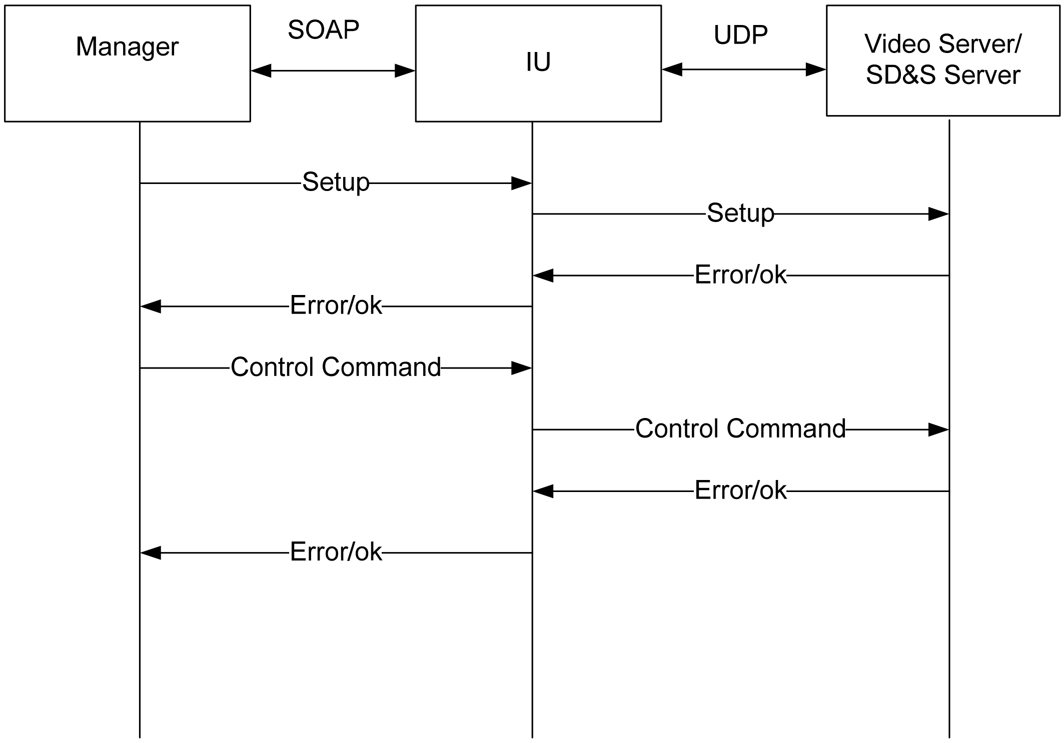

Fig. 18- Communication between manager-IU-VideoServer/SD&S Server.

4.2 *Communication between RCIC-BST*

For each terminal participating in a unicast session, the playout needs to be informed of the maximum bandwidth that is attainable between the base stations and that particular user terminal. In WiMAX and DVB, a number of parameters are signalled between the base stations and the terminals at the Media Access Control (MAC) level. These originate from the terminal, which measures certain link parameters, and signals them back to the base station, which changes its transmission characteristics accordingly. From these parameters, the available bandwidth may also be derived. Therefore, it is not necessary to measure the link quality at the receiver in a higher layer.

In order for the SUIT playout system to know the available bandwidth for every user terminal, the DVB and WiMAX base stations may be queried using a proprietary (non-standard) inter-process communication (IPC) protocol, based on User Datagram Protocol (UDP) messaging. In the base station, these packets are internally routed to the appropriate sector card. In the playout, the RCIC takes care of querying the base stations and reporting the retrieved information to other playout subsystems. This leads to the following service architecture:

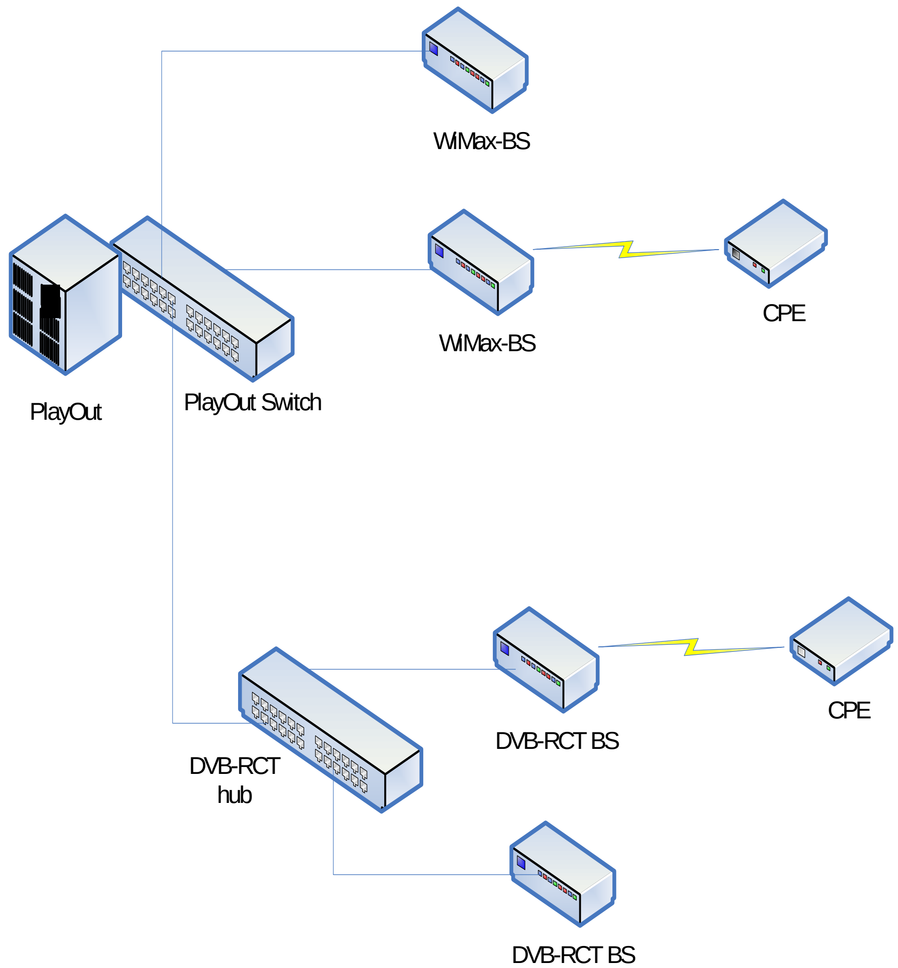

Fig. 19- Communication between Playout-BS.

A query to a base station consists of sending a PLAYOUT_IPC_LINK_STATUS_REQ message which allows individual, or all, user terminal's information to be accessed:

PLAYOUT_IPC_LINK_STATUS_REQ message format

Syntax	Size	Notes
UT number to poll	8bits	Number Ut's to poll (if eq. 0 poll all Ut's)
1^{st} UT MAC number to poll	48bits	Option
2^{nd} UT MAC number to poll	48bits	Option
…		

The base station replies with a PLAYOUT_IPC_LINK_STATUS_REP message, which carries, for every user terminal whose information was requested, the following payload:

PLAYOUT_IPC_LINK_STATUS_REP message format

Syntax	Size	Notes
UTMACAddress	48bits	
UlFecCodeType	8bits	Modulation and code-rate uplink
FecCodeType	8bits	Modulation and code-rate downlink
CurDownLinkCinr	8bits	Downlink CINR - UT – report.

CurUplinkCinr1_2db	16bits	UPLK as it measured in BS
RSSI	8bits	UPLK as it measured in BS
AbsoluteRSSI	16bits	UPLK as it measured in BS (effective RSSI)
CurTransmititerNoisePower	16bits	UT – report
dbRemainPower	8bits	UT remaining power
RxRemainPower	16bits	BS reception power

These magnitudes represent the MAC layer parameters, such as U/L and D/L FEC code and modulation employed, as well as link level parameters such as CINR, RSSI, and noise power values.

This data flow example is illustrated in the figure below:

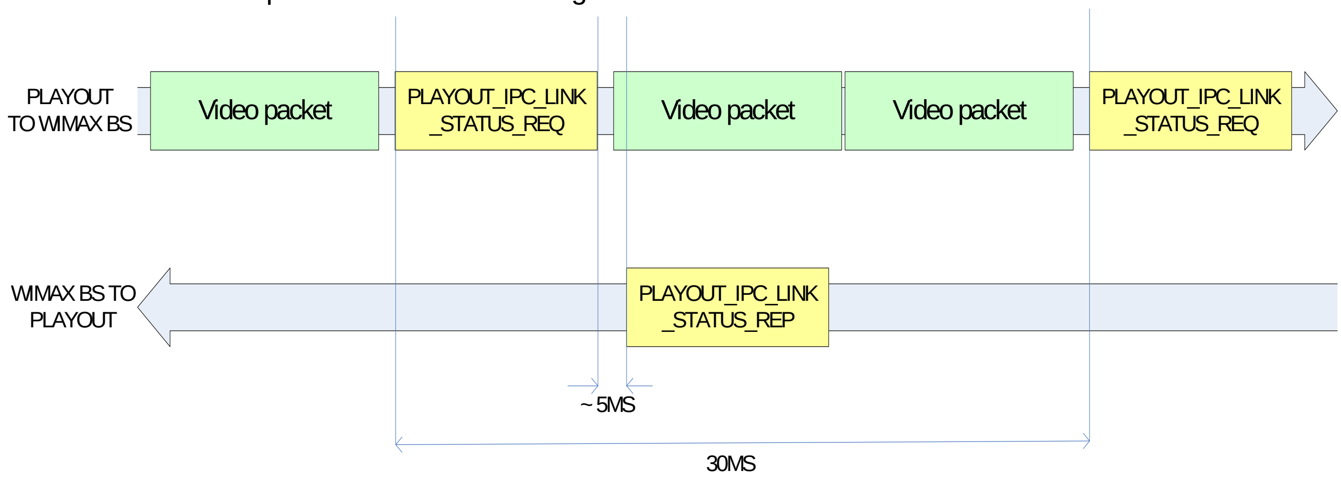

Fig. 20- Data flow between Playout-BS.

All retrieved parameters are stored in the SUIT playout database by the RCIC to allow monitoring or referencing by other subsystems. Most important as far as the RCIC and the playout are concerned is the downlink modulation and FEC code rate, which are signalled in the 'FecCodeType' field. The modulation and FEC code rate employed in the downlink directly determine the maximum D/L throughput, as indicated below:

Modulation	Code Rate	RSSI (BER=10^-5) dBm	CINR (BER=10^-5) dB	Throughput Mbit/sec
Down Link Using channel model Veh A at 60Km/h				
QPSK	1/2	-83	16	4.32
QPSK	3/4	-80	19	6.48
16QAM	1/2	-78	21	8.64
16QAM	3/4	-72	27	12.96
64QAM	1/2	-73	26	12.96
64QAM	2/3	-67	32	17.28
64QAM	3/4	-65	34	19.44
64QAM	5/6	-63	36	21.60
Up Link Using channel model Veh A at 60Km/h				
QPSK	1/2	-83	16	2.16
QPSK	3/4	-80	19	3.24

16QAM	1/2	-78	21	4.32
16QAM	3/4	-72	27	6.48

The RCIC will determine this throughput and store it separately in the database, so other subsystems do not need to repeat this mapping. Additionally, it is signalled directly to the video servers using a UDP message, avoiding any further polling delays.

4.3 *Communication between RCIC-Terminal*

Apart from the available downlink throughput, some other characteristics need to be signalled by the terminal to the playout. Most important are: the terminal's display resolution and the amount of hard disk capacity available. SUIT terminals convey these characteristics to the playout using Usage Environment Description (UED) messages. UED is a format from the MPEG-21[21] DIA (Digital Item Adaptation) tool suite. It is XML-based, hence structured, and is extensible to a large extent.

At the terminal side a function has been implemented that is to be integrated into the terminal program, allowing the latter to call this function whenever an updated message needs to be sent. At the playout side, the RCIC listens for incoming messages and upon reception, parses the received message, checks the SUIT database to see if the terminal is known to the system, and stores the parsed characteristics into the SUIT database. Additionally, characteristics which are important to the video adaptation process (in casu, the maximum display resolution, determining the highest spatial layer that should be sent), are signalled directly to the video servers using UDP[8] messaging.

The communication of UED messages from terminal to RCIC is based on the Transmission Control Protocol (TCP). The RCIC listens on a TCP socket to which the terminals can connect. These connections need to be routed through the respective SUIT gateways.
Below is shown a sample UED message passed from a terminal to the RCIC:

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS" id="loginname">
 <Item>
  <Descriptor>
   <Statement mimeType="text/xml">
    <DIA xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         schemaLocation="urn:mpeg:mpeg21:2003:01-DIA-NSUsageEnvironment.xsd"
         xmlns="urn:mpeg:mpeg21:2003:01-DIA-NS">
     <Description xsi:type="UsageEnvironmentType" id="sessionID">
      <UsageEnvironmentProperty xsi:type="TerminalsType">
       <Terminal>
        <TerminalCapability xsi:type="StoragesType">
         <Storage>
          <StorageCharacteristic id="suitstorage" xsi:type="StorageCharacteristicsType"
size="freeDiskSpace" writable="true" />
         </Storage>
        </TerminalCapability>
        <TerminalCapability xsi:type="DisplaysType">
         <Display id="suitdisplay">
          <DisplayCapability xsi:type="DisplayCapabilityType">
           <Mode refreshRate="refreshRate">
            <Resolution horizontal="horDisplayRes" vertical="vertDisplayRes" />
           </Mode>
          </DisplayCapability>
         </Display>
```

```
        </TerminalCapability>
       </Terminal>
      </UsageEnvironmentProperty>
     </Description>
    </DIA>
   </Statement>
  </Descriptor>
 </Item>
</DIDL>
```

Fig. 21- UED message Terminal-RCIC.

4.4 *Communication between SD&S Server-Gateway-Terminal*

This Section describes the communication between terminal, gateway and SD&S Server. As specified in the DVB-IP standard, the SD&S records can be transmitted either as multicast or unicast. In multicast case, Fig. 22, the DVB SD&S Transport Protocol (DVBSTP) is used to transport the records. In unicast case, Fig. 23, the HTTP[22] protocol is used to transport the records.
The DVBSTP [9] protocol is specified in the DVB-IP specification and is used when the SD&S information is transmitted using multicast UDP packets.
When SD&S information is available only via unicast, the terminal must use HTTP to communicate with the SD&S server(s)

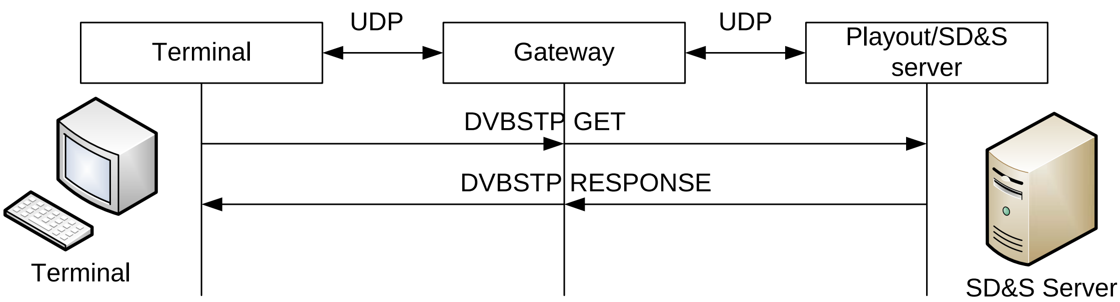

Fig. 22- Communication Diagram of Gateway and SD&S Server Multicast.

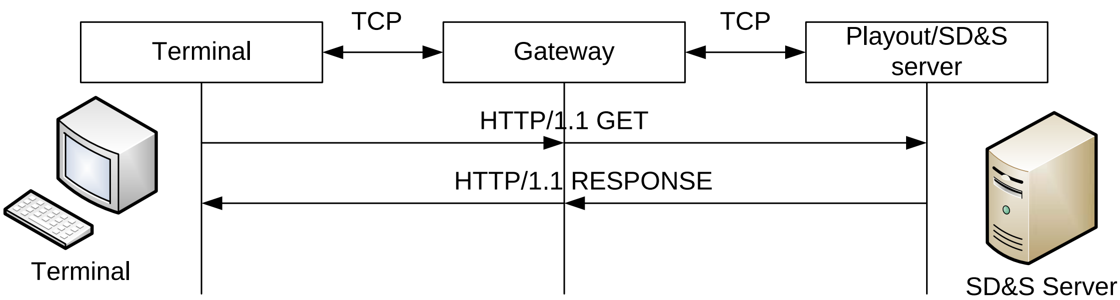

Fig. 23- Communication Diagram of gateway and SD&S Server Unicast

4.5 *Communication between IU-VideoServer-Terminal*

4.5.1 Video on Demand Scenario

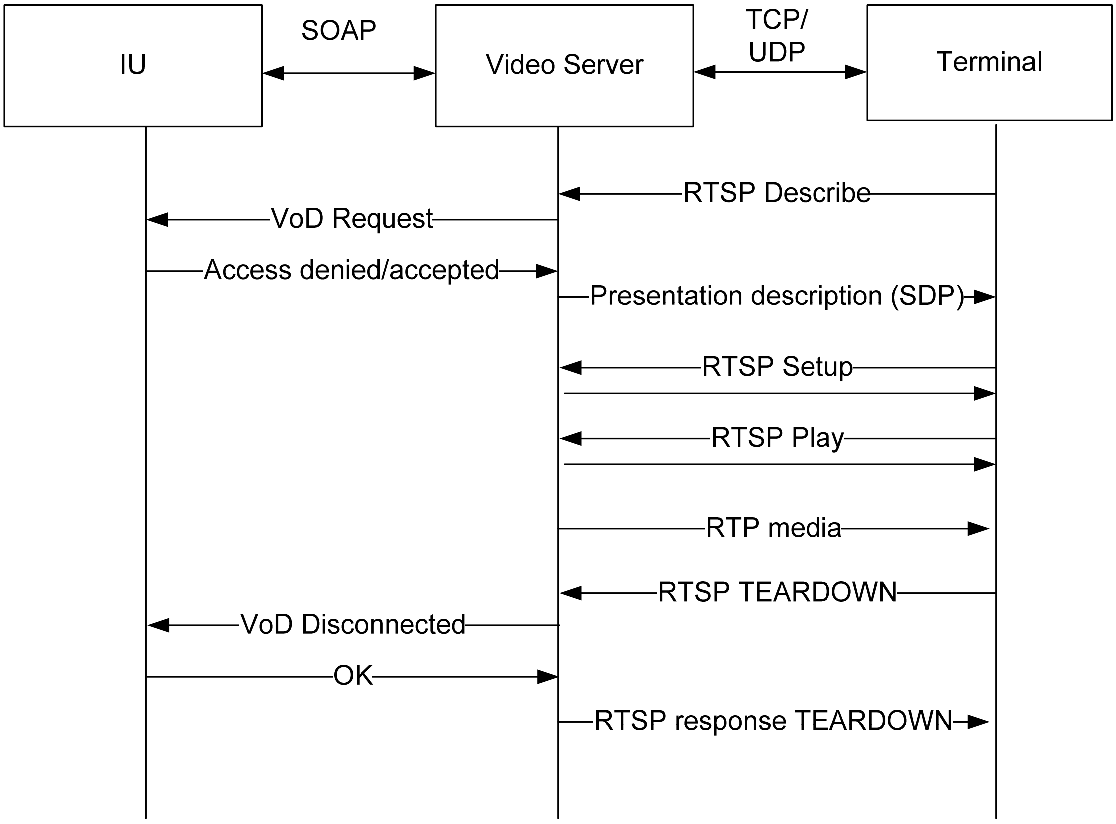

Fig. 24- Video on Demand scenario

Fig. 24 shows all communication and protocols used for Video on Demand in SUIT. Generally speaking, SUIT follows standards. The local Intelligent Unit manages VoD requests accepting/denying them.

4.5.2 Quality on Demand Scenario

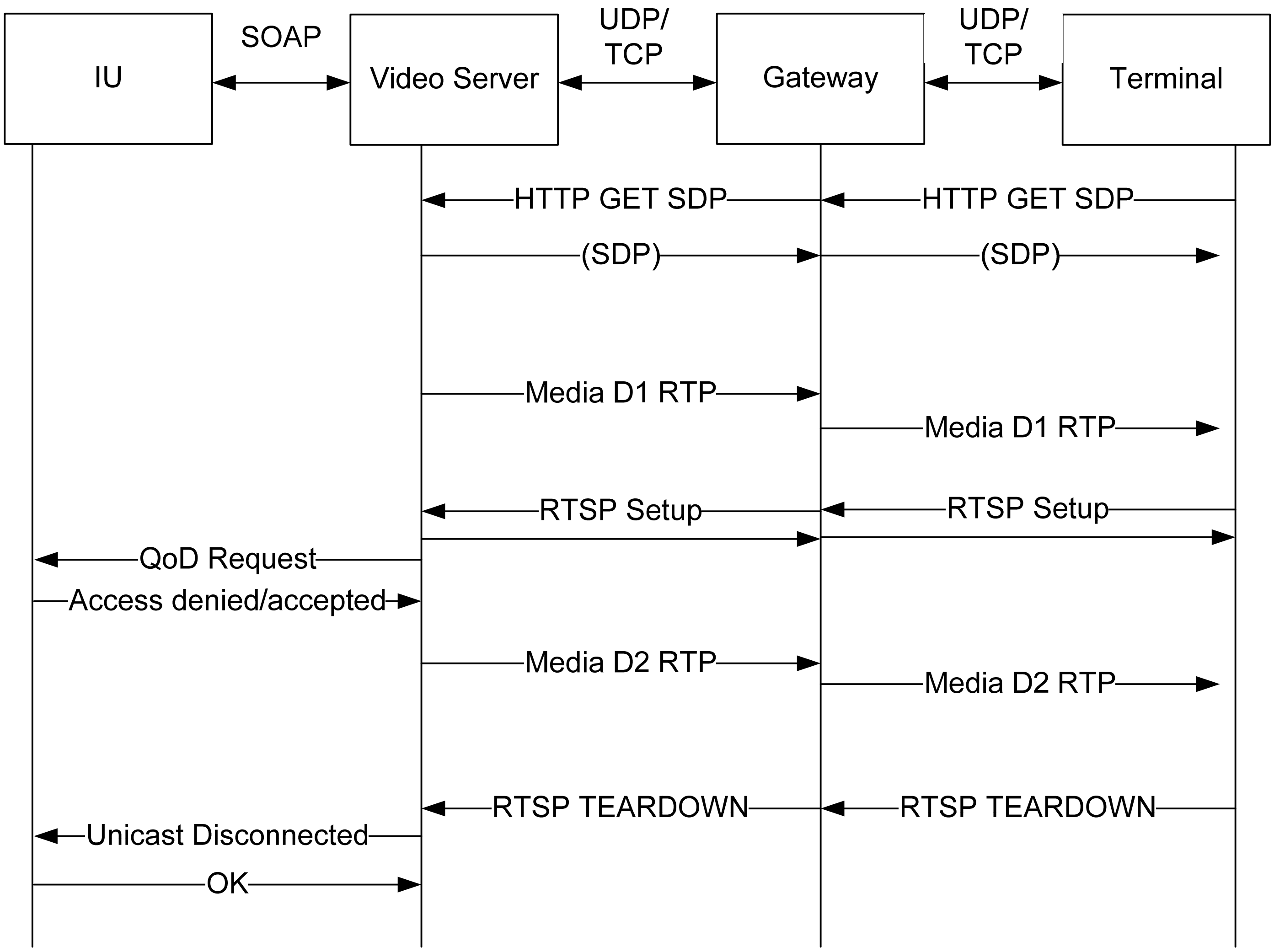

Fig. 25- Quality on Demand Scenario

Fig. 25 shows all communication and protocols used for requesting a higher Video quality in SUIT. Once again, SUIT tries to follow standards. Basically, the terminal requests a 2nd description in order to get a more robust communication system. The local Intelligent Unit manages this type of requests.

5 SUIT Database

5.1 *Configuration*

The SUIT database is based on MySQL 4.1, and its name is suit_db. It allows the playout to store all the information about services, descriptions, video servers, platforms, backups, users, transceivers, terminal capabilities, bitrates, etc.
The Intelligent Unit Web Service, the Web Manager, the SD&S Server and the RCIC are the modules that can access the database.
The Web manager and the IU access the database using a .net dll called MySql.Data.dll.

5.2 MySql Tables

The tables used are MyISAM.

6 Audio Integration

6.1 Introduction

6.1.1 Scope

This section of the report presents the architecture proposed for including audio within the SUIT project.

It includes:

- Description of the audio codec and the proposed MDC scheme that will be used
- System architecture with respect to the existing SUIT system
- RTP encapsulation scheme
- Synchronization issues
- Audio-visual multiplexing
- Implementation issues and plans for the demonstrator

The section is broken up into four further main sections:

- Section 6.2 describes the system architecture
- Section 6.3 delineates the audio codec to be used, the MDC scheme, surveys open source audio software, and deals with some implementation issues
- Section 6.4 discusses audio-visual multiplexing, and RTP encapsulation issues

6.2 System Architecture

6.2.1 Introduction

This section provides an outline of the basic system architecture with respect to the previously presented video-only system. The modified and added blocks are highlighted in the text and the diagrams. Further details of the functionality of these blocks are provided in sections 6.6.3 and 6.6.4.

6.2.2 System Description

6.2.2.1 Transmitter encoding & encapsulation Blocks

Fig. 23 shows a block diagram of the overall transmitter architecture.

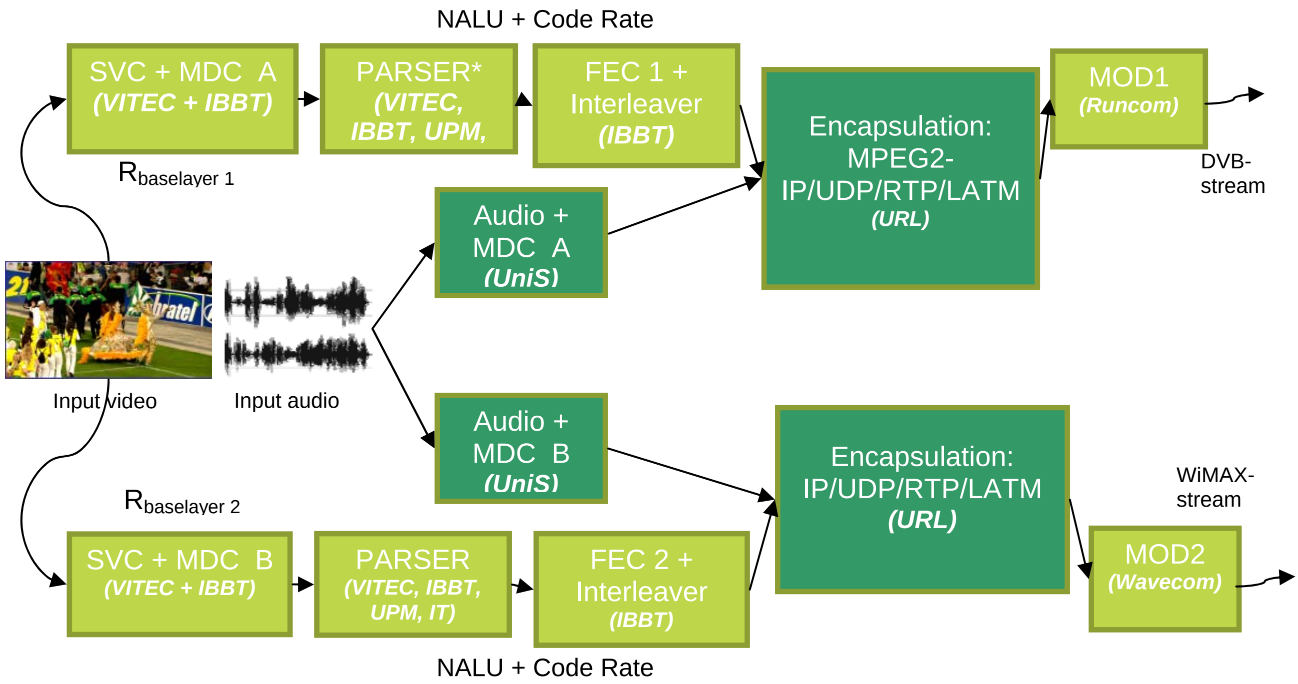

Fig. 26- Architecture for transmitter side, showing integration of audio. Affected blocks are highlighted in a dark green. Note the addition of MPEG-LATM to encapsulate audio.

Fig. 27 shows the audio coding algorithm based on multiple description audio coding and illustrates how the audio data is fed into the encapsulator for both descriptions.

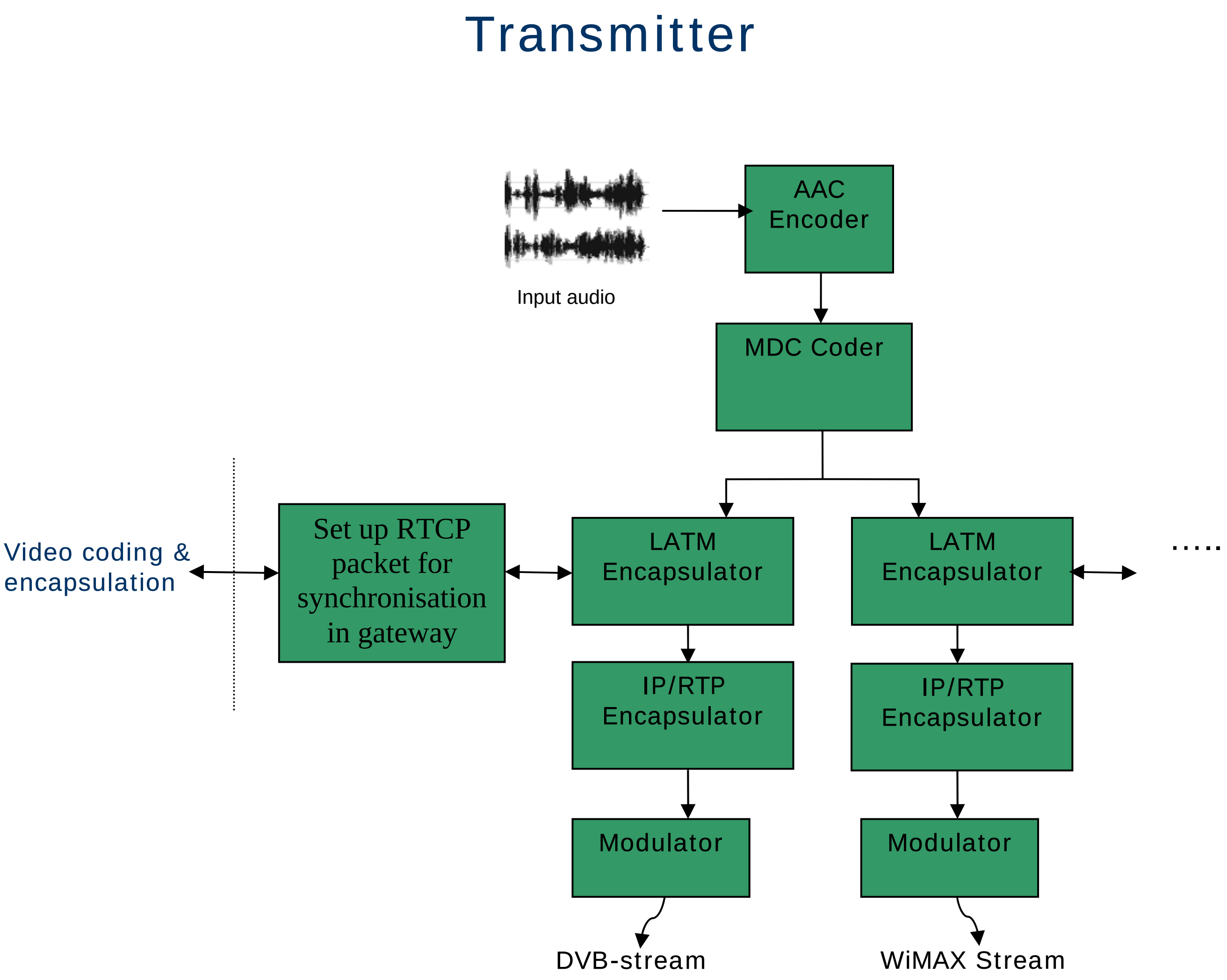

Fig. 27- Architecture for the transmission: Showing audio coding, encapsulation and enabling synchronisation with video

6.2.2.2 Gateway decoding, decapsulation, synchronisation and transmission to terminal Blocks

The existing decapsulation blocks is being modified to take into account the audio, and also pass timestamps to the respective combiners. The timestamps will be used to ensure that the audio and video streams can be synchronised after combining, when they need to be packaged for forwarding. It has been decided to use RTCP for synchronising audio and video.

Synchronisation is going to be performed similarly to the video combiner, so that packets from both descriptions can be handled by the MDC audio combiner algorithm in the gateway. To provide compatibility with the video transmission, LATM multiplexing has been decided to be used for audio encapsulation.

Fig. 28 shows the overall architecture for the gateway. A more detailed block diagram of the gateway and terminal are shown below in Fig. 29.

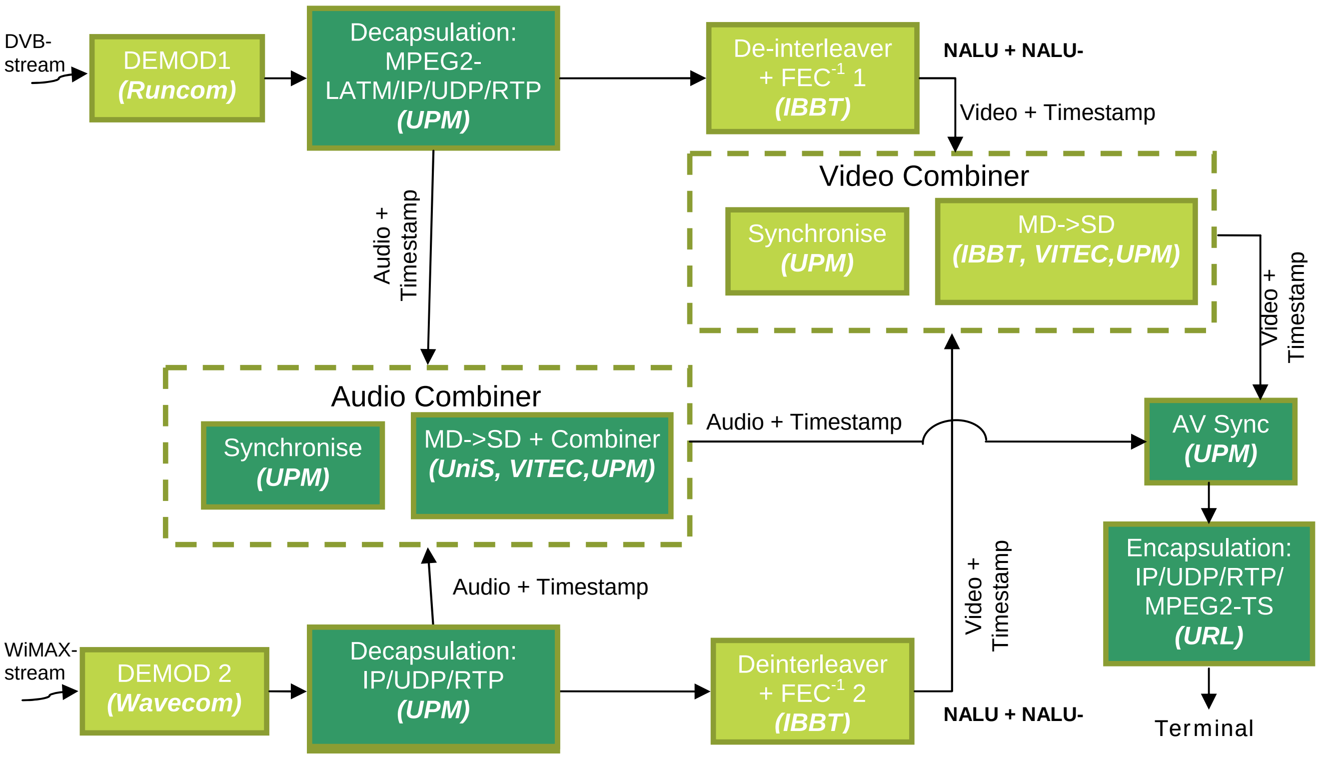

Fig. 28- Proposed receiver/gateway architecture.

Fig. 29 shows the process of audio decoding, when multiple descriptions are used. It also illustrates how the audio data is sent to the terminal.

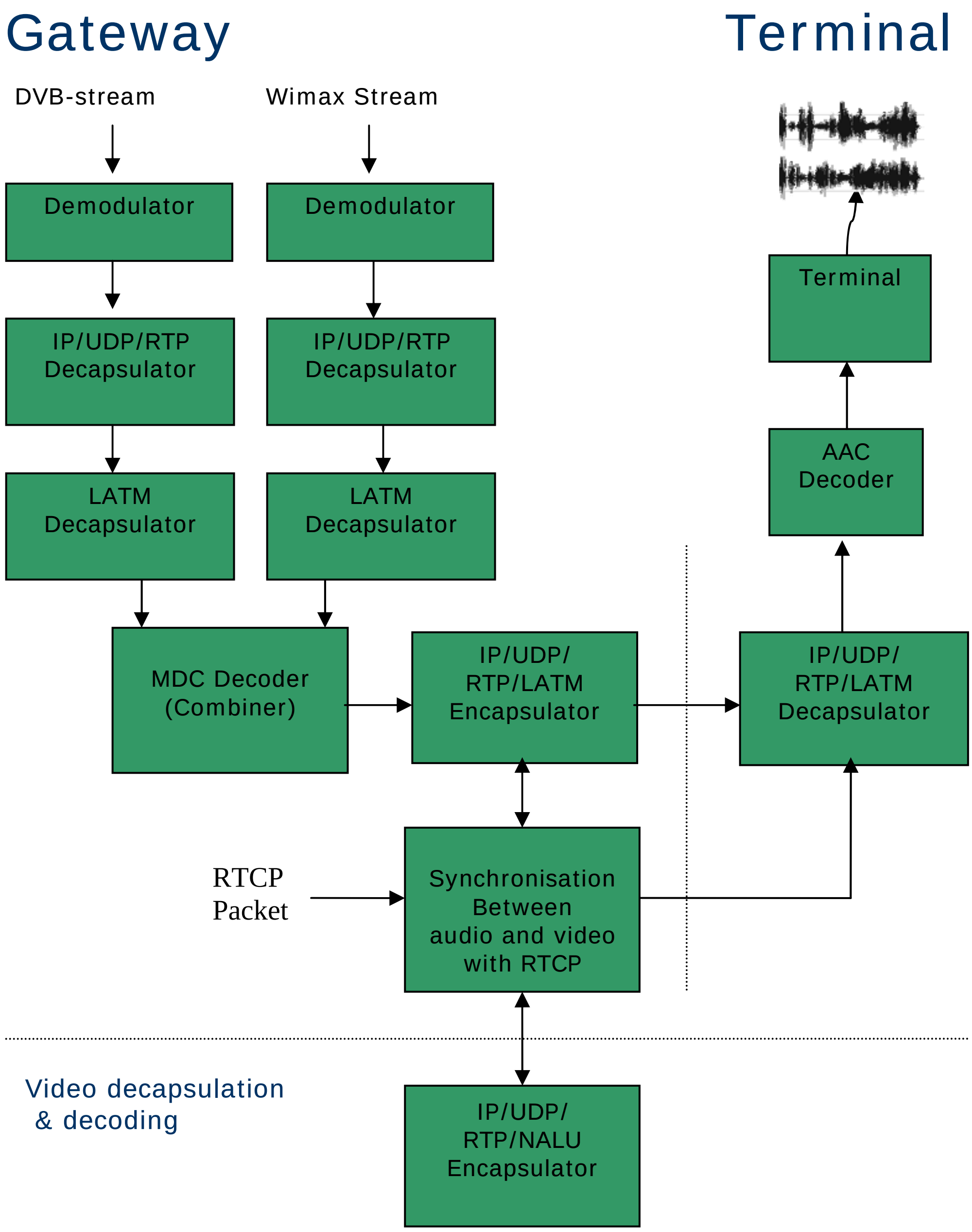

Fig. 29- Architecture for the transmission, showing audio coding, encapsulation and synchronization with video

6.3 Audio Codec Description

6.3.1 Introduction

This section provides an outline of the proposed scalable multiple description audio coding scheme and clarifies some theoretical and implementation issues.

Real time audio transmissions in IP networks suffer performance degradation because of the packet losses. Redundancy-free techniques such as interpolation can not repair perceivable damage in the audio signal if the packet loss rates exceed 1-2%. One solution would be the double transmission of packets. However, this implies a high cost in terms of bandwidth. Multiple Description Coding (MDC) may be incorporated into a scalable audio coder to provide both error-resilience and flexibility in the system.

6.3.2 Codec to be Deployed

Advanced Audio Coding (AAC) [11] is widely accepted as the state of the art in the audio coding. The proposed system is therefore based on AAC technology. Being more specific, High Efficiency (HE)-AAC [1] will be used which provides Spectral Band Replication (SBR). SBR is a compression tool which represents the high frequency part of the audio signal by replicating the low frequency part and using some low bit rate side information.

The scalability tools of MPEG audio, which may be SNR or bandwidth oriented, may be used.

6.3.3 Implementation Issues

- Sampling Rates [kHz]: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 96.
- Window Sizes [no of samples]: 120, 128, 960, 1024.
- Example frame lengths: 960/48kHz=20ms, 1024/32kHz=32ms.
- Real-time implementation of FAAC/FAAD has been evaluated on a 3.2 GHz PC, which reveals that:

  - Encoder uses 9% of CPU

  - Decoder uses 5% of CPU

6.3.3.1 <u>Survey of Encoding/Decoding Software</u>

FAAC (Free Advanced Audio Coder) is an open source software project which consists of all the necessary tools except MPEG Surround will be used. It is designed as library with a set of functions written in C programming language. The software works both in Windows and Linux. FAAC download information can be found at [13].

MPlayer has an audio coding module based on FAAC, called libFAAC (or libFAAD for the decoder). However this version of FAAC lacks scalability tools. To overcome this problem another version of FAAC, which includes scalability tools would need to be used.

MPEG Surround software has not been released yet. However since the rest of the system is independent of spatial coding, its software can be fused into the system when it is released.

6.3.3.2 Proposed Audio Set-up

The combiner block will will ensure that only one description is forwarded to the terminal. Given the early release of MPEG Surround, binaural and multi-channel rendering may be considered. Otherwise only stereo rendering will be considered.

6.3.3.3 Implementation Phases

Phase 1: the transmission of audio with LATM multiplexing (already implemented).

Phase 2: the synchronisation and joint transmission with video.

Phase 3: replicating audio packets in a similar fashion to the redundant slice scheme used for the video.

Phase 4: if time permits, more sophisticated MDC and scalability schemes will be incorporated into the system.

6.4 Encapsulation/Decapsulation

6.4.1 Introduction

This section discusses some of the additional issues that need to be addressed to cope with the integration of audio into the SUIT system. LATM multiplexing is being used for encapsulation of audio data.

6.4.2 Transmitter Audio Encapsulation/Decapsulation

LATM/UDP/IP encapsulation has been implemented for transmitting audio. This is illustrated in Figure 2. Software packages GPAC [14] and Live555 have been explored and used in the implementation of encapsulation and decapsulation. Another software package, VLC, is also being explored since the field trials are likely to be done with it.

6.4.3 Gateway Audio Encapsulation/Decapsulation

Fig. 26. illustrates the gateway and the terminal architecture. The transmitted audio format can be read and played by Mplayer.

6.4.4 Synchronisation

Synchronisation between audio and video will be done with RTCP packets.

6.5 Quality evaluation of audio

The PEAQ (Perceptual Evaluation of Audio Quality) tool will be used to assess the quality of the audio [15]. Methods for evaluating the quality of audio are discussed in more detail in deliverable 6.2.

6.6 Conclusions

- An architecture for the inclusion of audio within the SUIT system is described.

- The system is based on HE-AAC.
- FAAC/FAAD software is being used in the implementation.
- LATM is used for audio encapsulation/decapsulation.
- A combiner will be used in the gateway to combine the multiple description AAC streams into a single description AAC packet.
- Synchronisation between audio and video will be provided by using RTCP packets.

7 Acronyms

AAC	Advanced Audio Coder
API	Application Programming Interface
CBR	Constant Bit Rate
DVB	Digital Video Broadcasting
DVB-RCT	DVB Return Channel Terrestrial
DVB-T/H	DVB Terrestrial/ Handheld
FAAC	Free Advanced Audio Coder
IU	Intelligent Unit
HD	High definition
IP	Internet Protocol
ISP	Internet Service Provider
Kbps	Kilobits per second
LATM	Local Asynchronous Transfer Mode
Mbps	Megabits per second
MDC	Multiple Description Coding
QoD	Quality On Demand
MPEG-21	Motion Picture Experts Groups -21
RCIC	Return Channel Information Collector
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SOAP	Simple Object Access Protocol
TS	MPEG-2 Transport Stream
SVC	Scalable Video Coding
SD	Single Definition
SQL	Structured Query Language
UDP	User Datagram Protocol
VBR	Variable Bit Rate
SD&S	Service Discovery and selection
SDP	Session Description Protocol
VoD	Video On Demand
VLAN	Virtual Local Area Network
WS	Web Service

8 References

[1] IETF RFC 2326 Schulzrinne, H., Rao, A., and Lanphier, R., 1998. Real Time Streaming Protocol (RTSP). IETF, Request for Comments, RFC 2326.

[2] W3C Recommendation (27 April 2007). SOAP Version 1.2 Part 0: Primer (Second Edition).

[3] W3C Recommendation (27 April 2007). SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).

[4] W3C Recommendation (27 April 2007). SOAP Version 1.2 Part 2: Adjuncts (Second Edition).

[5] W3CW3C Recommendation (27 April 2007). SOAP Version 1.2 Specification Assertions and Test Collection (Second Edition).

[6] W3C Recommendation (25 January 2005). SOAP Message Transmission Optimization Mechanism

[7] W3C Recommendation (25 January 2005). Resource Representation SOAP Header Block.

[8] RFC 768 UDP (User Datagram Protocol) J. Postel ISI 28 August 1980

[9] ETSI TS 102034. Transport of MPEG-2 TS based DVB Services over IP based networks (09-2006)

[10] http://www.codingtechnologies.com/products/assets/CT_aacPlus_whitepaper.pdf

[12] http://www.mpegsurround.com/documents/SPACE_AES_119.pdf

[13] http://www.audiocoding.com

[14] http://gpac.sourceforge.net

[15] http://www.peaq.org/

[16] IETF RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video".

[17] IETF RFC 3984: "RTP payload for transport of H.264".

[18] IETF RFC 3550: "RTP, A Transport Protocol for Real Time Applications"

[19] “draft-wenger-avt-rtp-svc-03.txt”, S. Wenger, Y.-K. Wang, T. Schierl, June 2006

[20] IETF RFC 4566. Handley, M., Jacobson, V., and Perkins, C., Jul. 2006. SDP: Session Description Protocol, IETF, Request for Comments, RFC 4566.

[21] Moving Picture Experts Group Mar. 2003. Information technology — Multimedia framework (MPEG-21) — Part 7: Digital Item Adaptation. ISO/IEC JTC1/SC29/WG11 FDIS 21000-7.

[22] IETF RFC 2616. Hypertext Transfer Protocol – HTTP/1.1. R. Fielding UC Irvine, J. Gettys Compaq/W3C, J. Mogul Compaq, H. Frystyk W3C/MIT, L. Masinter Xerox, P. Leach Microsoft, T. Berners-Lee W3C/MIT. June 1999