<DISSEMINATION LEVEL>

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

SUIT Doc Number	SUIT_213
Project Number	IST-4-028042
Project Acronym+Title	SUIT- Scalable, Ultra-fast and Interoperable Interactive Television
Deliverable Nature	Report
Deliverable Number	D4.1
Contractual Delivery Date	30/11/2006
Actual Delivery Date	15/01/2007
Title of Deliverable	Synchronization and Encapsulation
Contributing Workpackage	WP4
Project Starting Date; Duration	01/02/2006; 27 months
Dissemination Level	PU
Author(s)	Antonio Navarro (IT) , Alois Zistler (IRT), Michael Probst (IRT), Francesc Enrich (URL), Victor Domingo (URL), Francesc Pinyol(URL),

Abstract

SUIT project gathers several transmission technologies to deliver contents to the final user. The contents delivered in separate networks have to be encapsulated and synchronized in such a way that the terminal could be able to recover them correctly. This document presents different technologies used and specifies the protocols (some still experimental) to encapsulate correctly all the content that will be delivered to the terminals. Additionally, the document presents the solution adopted to synchronize the contents.

Keyword list: RTP, RTCP, H.264, SVC, MD, Encapsulation, Synchronization, DVB-T/H, IEEE16e

PUBLIC

Synchronization and Encapsulation

SUIT_106

15-01-2007

Table of Contents

1 Introduction

SUIT project gathers several transmission technologies to deliver contents to the final user. The contents delivered in separate networks have to be encapsulated and synchronized in such a way the terminal would be able to recover correctly the contents.
This document presents the different technologies used and specifies the protocols (some still experimental) to encapsulate correctly all the content that will be delivered to the terminals. Moreover, the document presents the solution adopted to synchronize the contents.
This deliverable is organized as follows: Section 2 presents the SUIT architecture and identifies all the protocols used in the play-out; Section 3 identifies different transmission technologies used in SUIT project and specifies the encapsulation protocols that will be used to transmit all the content through the SUIT network regarding to each technology; Section 4 summarizes the current methods to synchronize contents depending on the technology used in SUIT and how this methods will be applied in this project. Finally section 5 presents the main conclusions of this document.

2 Synchronization and Encapsulation related to the SUIT Project

2.1 Suit architecture and end-to-end SUIT protocol stack overview

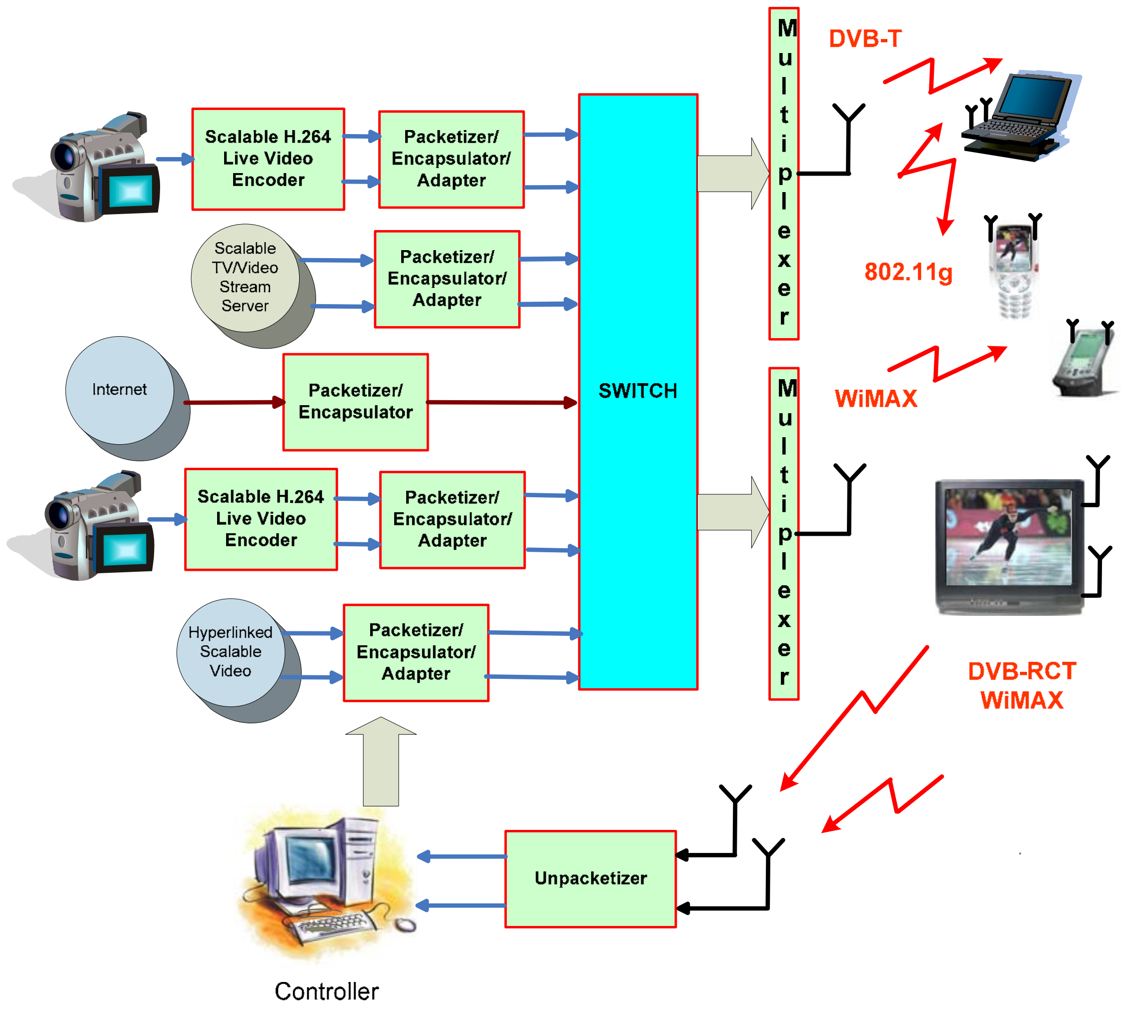

Scalable H.264 Live Video Encoder
Packetizer/ Encapsulator/ Adapter
Scalable TV/Video Stream Server
Packetizer/ Encapsulator/ Adapter
Internet
Packetizer/ Encapsulator
Scalable H.264 Live Video Encoder
Packetizer/ Encapsulator/ Adapter
Hyperlinked Scalable Video
Packetizer/ Encapsulator/ Adapter
SWITCH
Multiplexer
Multiplexer
DVB-T
802.11g
WiMAX
DVB-RCT WiMAX
Unpacketizer
Controller

Figure 1: Suit overall architecture

Figure 1 above outlines the overall SUIT reference architecture. The radio interfaces are fed by live scalable contents, pre-recorded scalable contents and internet data. The playout will dynamically

and optimally manage all those resources and adapt them according to the network conditions. Once the SUIT terminal interfaces to the WLAN, it will deal locally with instantaneous variations of QoS and thus minimizing the effects by reacting as swift as possible. The connection between the playout and the transmitters will be by the core network, possibly radio links that allow us to implement different field trials easily.
In SUIT, there will be three types of end-user terminals to demonstrate the various network and service scenarios:

- WiMAX/DVB-T/H/-RCT terminal – to demonstrate -amongst others- mobility by reception of multiple description scalable video content (including handover)
- WiFi terminal – to receive rate-adapted content via a IEEE 802.11g (WiFi) connection from a deployed WiMAX/DVB-T/H gateway
- MHP-IPTV terminal – to demonstrate MHP applications and hyperlinked content-on-demand remote-interactively retrieved from a playout server and conveyed over WiMAX

Deliverable D1.4 - Architecture and Reference Scenarios identified three network scenarios, Home, Mobile and MHP-IPTV. The first one, Home, requires a gateway as depicted, below, in *Figure 2*. The WiFi end-user terminals will be fed via the gateway interfacing to two different wireless broadband last mile networks, WiMAX and DVB-T/H. The gateway informs the playout about the network and terminal characteristics. The playout selects the right network to deliver the content and may reduce the bit-rate for e.g. video-on-demand (unicast) according to the network conditions. Figure 2 shows two homes and therefore two home gateways. In each home two terminals are connected to each gateway.

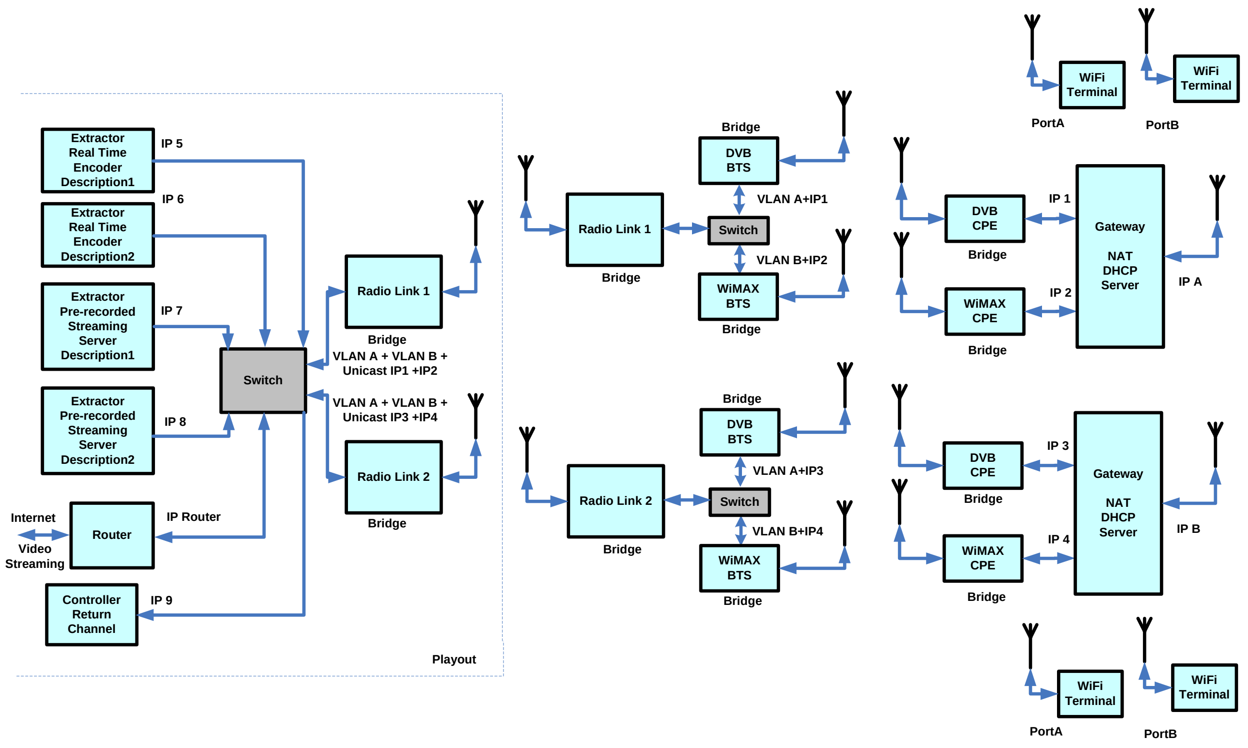


Figure 2: Block diagram of the home network scenario

SUIT will set up four base stations in two cells, where they will be co-sited in pairs. So, each cell will have one DVB-T/H base station and one WiMAX base station. This network scenario allows us to test different types of services and functionalities. However, it requires four experimental frequencies and the associated licenses in order to perform the field trials.
Deliverable D1.4 also mentioned service scenarios. With the application of new techniques in the SUIT project, various innovating service scenarios will be enabled when converging the two broadband mobile networks IEEE 802.16e (WiMAX), ETSI/EN 300 744 (DVB-T) and ETSI/EN 302

304 (DVB-H). SUIT will deliver a layered description of each of those last mile networks. By using two different video descriptions, SUIT will push video scalability into broadcasting and telecom networks in a fruitful way. As a final objective, SUIT intends to demonstrate an end-to-end communication system, from the playout to the terminal, where the terminal can feed an HDTV screen or a small pocket-sized display.
In terms of encapsulation required at the playout, let us simplify Figure 2 as shown in Figure 3, where it is supposed that only one basestation is in operation in each cell.

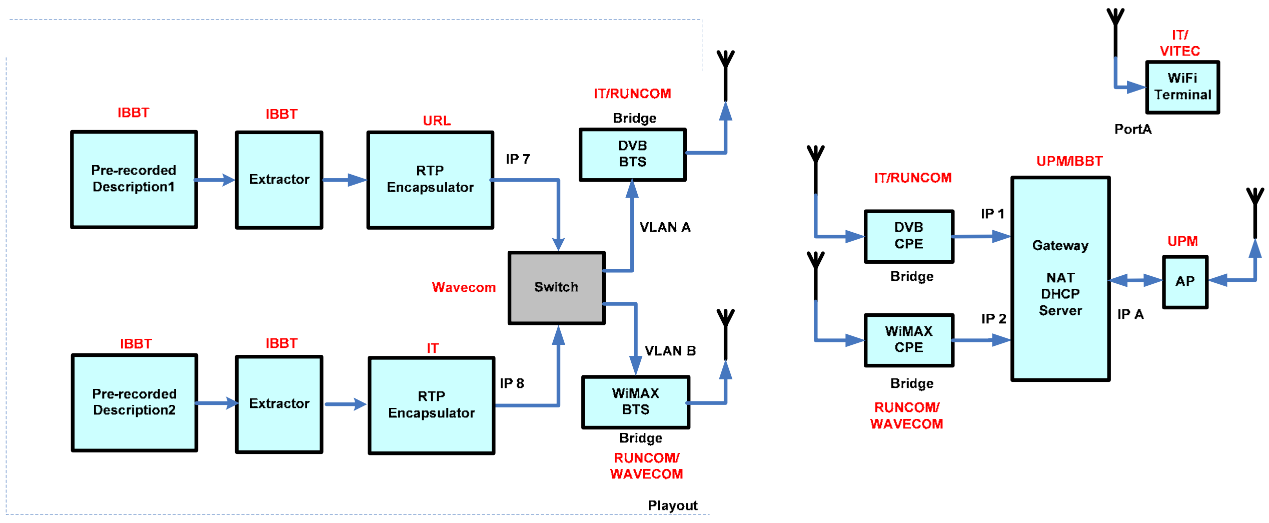


Figure 3: First audit demo scenario

From *Figure 3*, we conclude that the encapsulation procedure before the Switch is identical for both branches, WiMAX and DVB.

3 Encapsulation

3.1 Introduction

As we have seen in section*Figure 2*2, in the SUIT architecture several protocols are used depending on the protocol stack level. This chapter is organized in five sections. Section 3.2 presents an introduction to the generic RTP encapsulation specification and makes an overview of the current RTP specifications and drafts for the H264/SVC payload, and finally presents the encapsulation solution adopted by the SUIT project. The following sections deal with the DVB-T, WIMAX and DVB-IPI specifications.

3.2 RTP encapsulation

3.2.1 Introduction

Real Time Protocol was designed to satisfy the necessity for a robust mechanism to deliver real-time media above an unreliable transport layer.
RTP is a non complete protocol, allowing each application to implement new functionalities depending on data to be delivered and new features to be achieved. This means that exist other documents that complements the RTP protocol specification to be used in a specific application:

- **RTP Profiles:** This document defines a group of payload types (for example video types) and it can define modifications and extensions to fit in the RTP protocol to the application characteristics.
- **RTP Payload format:** This kind of documents defines the way to deliver a particular kind of payload.

RTP consists of two parts: the **RTP Data Transfer Protocol** and an **RTP Control Protocol**.

The **RTP Data Transfer Protocol** manages delivery of multimedia real-time data (audio and video), between end-to-end systems. It already adds a sequence number for loss detection, timestamp to enable timing recovery, payload type and source identifiers, and a marker for significant events within the media stream. Each RTP payload profile defines rules for timestamp, sequence number usage and for multiplexing multiple streams within a session.
The **RTP Control Protocol (RTCP)** provides information about the reception quality feedback, participant identification, and synchronization between media streams. RTCP runs alongside RTP and provides periodic reporting of this information. Although data packets are typically sent every few milliseconds, the control protocol operates on the scale of seconds. The information sent in RTCP is necessary for synchronization between media streams—for example, for lip synchronization between audio and video—and can be useful for adapting the transmission according to reception quality feedback, and for identifying the participants.

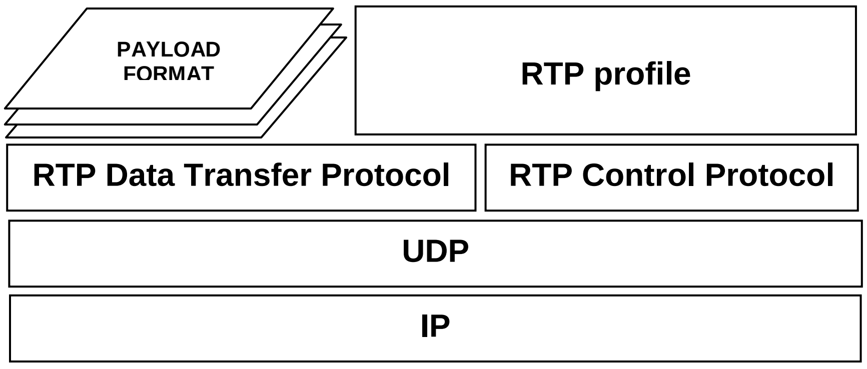


Figure 4: RTP Components

3.2.2 RTP Specifications

On November 22, 1995, RTP was approved by the IESG as an Internet proposed standard. It has been published as

- **RFC 1889**, *RTP: A Transport Protocol for Real-Time Applications*
- **RFC 1890**, *RTP Profile for Audio and Video Conferences with Minimal Control*

Nowadays, RTP and the associated audio-video profile are Internet Standard protocols, designated as STD 64 and STD 65, documented in RFC 3550 and 3551.
RFC 3550 describes RTP, the real-time transport protocol. RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. The protocol supports the use of RTP-level translators and mixers.
RFC 3551 describes the Audio-Video Profile of RTP, "RTP/AVP", providing interpretations of generic fields within the RTP specification suitable for audio and video conferences with minimal control. It defines a set of default mappings from payload type numbers to encodings, pointers to reference implementations and standards, and registrations of media types.
RFC 3984 describes an RTP Payload format for the ITU-T Recommendation H.264 video codec and the technically identical ISO/IEC International Standard 14496-10 video codec. The RTP payload format allows for packetization of one or more Network Abstraction Layer Units (NALUs), produced by an H.264 video encoder, in each RTP payload. The payload format has wide applicability, as it supports applications from simple low bit-rate conversational usage, to Internet video streaming with interleaved transmission, to high bitrate video-on-demand.

RTP Payload Format for SVC Video (October 2006) describes an RTP Payload format for the scalable extension of the ITU-T Recommendation H.264 video codec which is the technically identical to ISO/IEC International Standard 14496-10 video codec. The RTP payload format allows for packetization of one or more Network Abstraction Layer Units (NALUs), produced by the video encoder, in each RTP payload. The payload format has wide applicability, as it supports applications from simple low bit-rate conversational usage, to Internet video streaming with interleaved transmission, to high bit-rate video-on-demand.

3.2.3 RTP Data Transfer Protocol

3.2.3.1 RTP Sessions

We could define as a multimedia session a group of RTP flows that can be accessed by a group of users. This session usually consists of one or more data types. Each data type can be delivered in a different RTP session allowing to each participant discarding data flows depending on the network or terminal capacities. Thus, a multimedia session can finally be defined as a group of RTP sessions.
One RTP session is an association between one group of participant sharing same RTP data type. Each RTP session has its own RTCP control which is usually delivered separately from the RTP data flows. To do this a different pair of ports is assigned (one for RTP and another to the RTCP) to each session. Usually we use an even port for RTP and the immediate superior for RTCP.
A session can be unicast, either directly between two participants (a point-to-point session) or to a central server that redistributes the data. Or it can be multicast to a group of participants. A session also need not be restricted to a single transport address space.

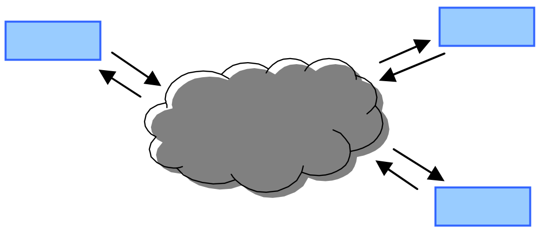


Figure 5: Multicast Example

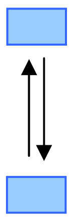


Figure 6: Unicast Example

3.2.3.2 The RTP Data Transfer Packet

The RTP packet consists of four parts:
- o The mandatory RTP header
- o An optional header extensión
- o An optional payload header (depending on the payload format used)
- o The payload data itself

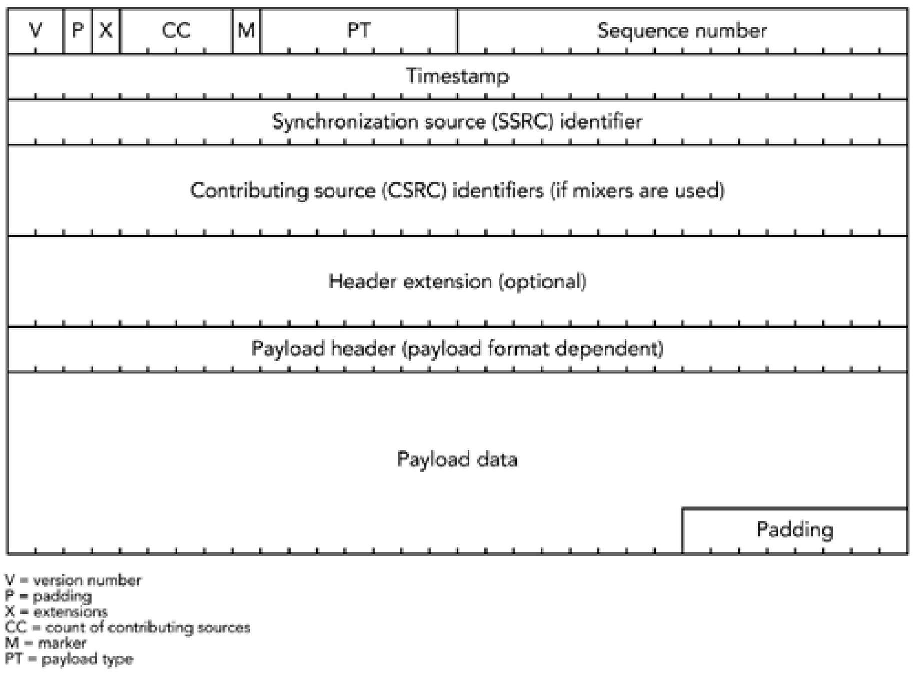

V
P
X
CC
M
PT
Sequence number
Timestamp
Synchronization source (SSRC) identifier
Contributing source (CSRC) identifiers (if mixers are used)
Header extension (optional)
Payload header (payload format dependent)
Payload data
Padding
V = version number
P = padding
X = extensions
CC = count of contributing sources
M = marker
PT = payload type

Figure 7: RTP Transfer Packet

V, Version (2 bits): RTP version number. Always set to 2.
P, Padding. (1 bit): If set, this packet contains one or more additional padding bytes at the end which are not part of the payload. The last byte of the padding contains a count of how many padding bytes should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.
X, Extension. (1 bit): If set, the fixed header is followed by exactly one header extension.
CC, CSRC count. (4 bits): The number of CSRC identifiers that follow the fixed header.
M, Marker. (1 bit): The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field.
PT, Payload Type. (7 bits): Identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats.
Sequence Number. (16 bits): The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable).
Timestamp. (32 bits): The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations. The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means. If RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock. As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

SSRC, Synchronization source. (32 bits): Identifies the synchronization source. The value is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same *SSRC*..
CSRC, Contributing source. (32 bits): An array of 0 to 15 CSRC elements identifying the contributing sources for the payload contained in this packet. The number of identifiers is given by the *CC* field. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources..

3.2.4 RTP Payload format for the H.264/AVC and H264/SVC

The H.264 video codec covers all forms of digital compressed video from, low bit-rate Internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. Compared to the current state of technology, the overall performance of H.264 is such that bit rate savings of 50% or more are reported. Digital Satellite TV quality, for example, was reported to be achievable at 1.5 Mbit/s, compared to the current operation point of MPEG 2 video at around 3.5 Mbit/s.
H.264 codec specification distinguishes conceptually between a video coding layer (VCL) and a network abstraction layer (NAL).

- Video Coding Layer (VCL): it mainly contains the signal processing functionality of the codec, such as: mechanisms such as transform, quantization, and motion compensated prediction. The VCL outputs slices.
- Network Abstraction Layer (NAL): it encapsulates the slice output of the VCL encoder into Network Abstraction Layer Units (NAL units), which are suitable for transmission over packet networks or use in packet oriented multiplex environments.
A NAL unit of H264/AVC consists of a one byte header and the payload byte string. The header indicates the type of the NAL unit, the (potential) presence of bit errors or syntax violations in the NAL unit payload, and information regarding the relative importance of the NAL unit for the decoding process. This RTP payload specification is designed to be unaware of the bit string in the NAL unit payload. One of the main properties of H.264 is the complete decoupling of the transmission time, the decoding time, and the sampling or presentation time of slices and pictures. The decoding process specified in H.264 is unaware of time, and the H.264 syntax does not carry information such as the number of skipped frames. Also, there are NAL units that affect many pictures and that are, therefore, inherently timeless. For this reason, the handling of the RTP timestamp requires some special considerations for NAL units for which the sampling or presentation time is not defined or, at transmission time, unknown.

3.2.4.1 Common structure for the RTP Payload Format

The SVC RTP payload is designed to provide backward compatibility with [7] wherever possible. This means that expect the SVC base layer to be H.264/AVC [1] compatible, we assume the base layer (when transmitted in its own session) to be encapsulated using [7]. Requiring this has the desirable side effect that it can be used by [7] legacy devices. Packet integrity needs to be preserved end-to-end.
The payload format defines three different basic payload structures. The receiver can identify these structures by the first byte of the RTP payload (Figure 8), which co-serves as the RTP payload header. This byte is always structured as a NAL unit header. The NAL unit type field indicates which structure is present.

0	1	2	3	4	5	6	7
F	NRI		TYPE				

Figure 8: NAL Unit Octet

F (forbidden_zero_bit): (1 bit) A value of 0 indicates that the NAL unit type octet and payload should not contain bit errors or other syntax violations. A value of 1 indicates that the NAL unit type octet and payload may contain bit errors or other syntax violations. The H.264 specification

requires that the F bit is equal to 0. When the F bit is set, the decoder is advised that bit errors or any other syntax violations may be present in the payload or in the NAL unit type octet.
NRI (nal_ref_idc): (2 bits) The semantics of value 00 and a non-zero value remain unchanged from the H.264 specification. In other words, a value of 00 indicates that the content of the NAL unit is not used to reconstruct reference pictures for inter picture prediction. Such NAL units can be discarded without risking the integrity of the reference pictures. Values greater than 00 indicate that the decoding of the NAL unit is required to maintain the integrity of the reference pictures. In addition to the specification above, according to this RTP payload specification, values of NRI greater than 00 indicate the relative transport priority, as determined by the encoder.
According [9], for a slice or slice data partitioning NAL unit, a NRI value of 11 indicates that the NAL unit contains data of a key picture.
TYPE (nal_unit_type): (5 bits) The NAL unit type field indicates which structure is present.
According [9], this component specifies the NAL unit payload type as defined in table 7-1 of [9], and later within this memo. For a reference of all currently defined NAL unit types and their semantics, please see table 3.
Previously, NAL unit types 20 and 21 (among others) have been reserved for future extensions. SVC is using these two NAL unit types. They indicate the presence of one more byte that is helpful from a transport viewpoint. The additional byte(s), described below, is called transport priority indicator.

0	1	2	3	4	5	6	7
RR		PRID					

Figure 9: Transport Priority Indicator

RR (reserved_zero_two_bits): (2 bits) Must be zero.
PRID (simple_priority_id): (6 bits) This component specifies a priority identifier for the NAL unit. A lower value of PRID indicates a higher priority. MANEs implementing unequal error protection may use this information to protect NAL units with smaller PRID values better than those with larger PRID values, for example by including only the more important NAL units in a FEC protection mechanism. The desirable transport priority increases as the PRID value increases.

0	1	2	3	4	5	6	7
TL			DID			QL	

Figure 10: Third byte SVC-NALU header

TL (temporal_level): (3 bits) indicates the temporal layer (or frame rate) hierarchy. Informally put, a layer consisted of pictures of a smaller temporal_level value has a smaller frame rate. A given temporal layer typically depends on the lower temporal layers (i.e. the temporal layers with smaller temporal_level values) but never depends on any higher temporal layer.
DID (dependency_id): (3 bits) Denotes the inter-layer coding dependency hierarchy. At any temporal location, a picture of a smaller dependency_id value may be used for inter-layer prediction for coding of a picture of a larger dependency_id value, while a picture of a larger dependency_id value is disallowed to be used for inter-layer prediction for coding of a picture of a smaller dependency_id value.
QL (quality_level): (2 bits) Designates the quality level hierarchy of a progressive refinement (PR) or quality (SNR) enhancement layer slice. At any temporal location and with identical dependency_id value, a picture with quality_level equal to ql uses a picture with quality_level equal to ql-1 for inter-layer prediction.

Values of TL, DID or QL indicate the relative priority in their respective dimension. A higher value of TL, DID or QL indicates a higher priority if the other two components are identical correspondingly. MANEs may use this information to protect more important NAL units better than less important NAL units.

0	1	2	3	4	5	6	7
R	B	U	D	G	L	O	

Figure 11: Fourth byte SVC-NALU header

R (reserved_zero_bit): (1 bit) Reserved bit for future extension. R must be zero.
B (layer_base_flag): (1 bit) Indicates that no inter-layer prediction (of coding mode, motion, sample value, and/or residual prediction) is used for the current slice otherwise inter-layer prediction may be used. A MANE or receiver may use this information in order to identify the [1] conforming base layer NAL units (if marked by a suffix NAL unit) and may determine the temporal layer (by the TL value of the suffix NAL unit) of it. Thus it allows for generating an outgoing RTP stream, with a certain temporal scalability layer that conforms to [7] and [1].
U (use_base_prediction_flag): (1 bit) Indicates that the base representation of the reference pictures (i.e. only NAL units of the reference pictures with QL equal to zero are used for inter prediction) is used during the inter prediction process.
D (discardable_flag): (1 bit) A value of 1 indicates that the content of the NAL unit with dependency_id equal to currDependencyId is not used in the decoding process of NAL units with dependency_id larger than currDependencyId. Such NAL units can be discarded without risking the integrity of higher scalable layers with larger values of dependency_id. discardable_flag equal to 0 indicates that the decoding of the NAL unit is required to maintain the integrity of higher scalable layers with larger values of dependency_id.
MANEs may use this information to protect NAL units with D equal to 0 better than NAL units with D equal to 1. Furthermore a MANE or a receiver may determine whether a given NAL unit is required for successfully decoding a certain operation point of the SVC bitstream.
G (fragmented_flag): (1 bit) Indicates that the current NAL unit is fragmented, which may be the case for partitions of an FGS (progressive refinement) slice.
L (last_fragemented_flag): (1 bit) Indicates, that the NAL unit is the last fragment of a fragmented NAL unit.
O (fragment_order): (2 bits) Indicates the order in which the NAL units with fragmented_flag equal to 1 shall be ordered before the parsing process is started, starting from lower values.
For G, L and O, a MANE or receiver may detect a fragmented PR slice by G, L and O. Using this knowledge may let the MANE do FGS adaptation on the PR slice, by forwarding not all of the fragments in fragement_order (O).
PRID, D, TL, DID, and QL, in combination, provide complete information of the relative priority of a NAL unit compared to any other NAL unit.
Next, the NAL possible structures are described:

- **Single NAL Unit Packet:** Contains only a single NAL unit in the payload. The NAL header type field will be equal to the original NAL unit type.

Type	NAL Unit Type name
0	Unspecified
1	Coded slice
2	Coded slice data partition A
3	Coded slice data partition B
4	Coded slice data partition C
5	Coded slice of an IDR picture
6	Supplemental enhancement information (SEI)
7	Sequence parameter
8	Picture parameter
9	Picture delimiter
10	End of sequence
11	End of stream

12	Filler data

Table 1: Orginal NAL unit types codes

- **Aggregation packet:** Packet type used to aggregate multiple NAL units into a single RTP payload. This packet exists in four versions:
    - Single-Time Aggregation Packet type A (STAP-A).
    - Single-Time Aggregation Packet type B (STAP-B)
    - Multi-Time Aggregation Packet (MTAP) with 16-bit offset (MTAP16)
    - Multi-Time Aggregation Packet (MTAP) with 24-bit offset (MTAP24).
- **Fragmentation unit:** Used to fragment a single NAL unit over multiple RTP packets. There are two versions: FU-A and FU-B.

In the following table there is the relation between NAL unit types and their payload structures:

Type	Packet Type name
0	undefined
1-23	NAL unit Single NAL unit packet per H.264
24	STAP-A Single-time aggregation packet
25	STAP-B Single-time aggregation packet
26	MTAP16 Multi-time aggregation packet
27	MTAP24 Multi-time aggregation packet
28	FU-A Fragmentation unit
29	FU-B Fragmentation unit

Table 2: NAL unit types

As you have seen, the SVC RTP payload will follow the same format as [7]. But besides the NAL Header extension and its particular meanings, the SVC draft adds the following points:
When different layers of a SVC bitstream are transported over more than one RTP session, SSRC multiplexing (The scalable SVC bitstream is distributed in a single RTP session, but that session comprises more than one RTP packet stream, identified by its SSRC), may be applied. When SSRC multiplexing is in use, the same IP address and port number are shared between all RTP streams and all layers, while the relative importance for the decoding process of each RTP stream and/or layer is differentiated by the SSRC values.
A packet with a higher SSRC value contains data belonging to higher layers or layers of lower transport priority. SSRC multiplexing as discussed works only for Single-Sender RTP sessions (an (perhaps multicasted) RTP session in which all RTP packet streams in the session stem from entities that are in close cooperation, and can coordinate SSRC values).
It is also possible to implement session multiplexing, in this case the scalable SVC bitstream is distributed onto different RTP sessions, whereby each RTP session carries one RTP packet stream. Each RTP session requires a separate signaling and has a separate Timestamp, Sequence Number, and SSRC space.

3.2.4.2 Packetization modes

There are three possible cases of packetization modes:
The packetization mode in use is signaled by the value of the optional packetization-mode MIME parameter or by external means.

-**Single NAL unit mode:** The single NAL unit mode is targeted for conversational systems that comply with ITU-T Recommendation H.241. This mode is in use when the value of the optional packetization-mode MIME parameter is equal to 0, the packetization-mode is not present, or no other packetization mode is signaled by external means. All receivers must support this mode. The transmission order of single NAL unit packets must comply with the NAL unit decoding order. The single NAL unit mode shall not be used in [9].

-Non-interleaved mode: The non-interleaved mode is targeted for conversational systems that may not comply with ITU-T Recommendation H.241. NAL units are transmitted in NAL unit decoding order. This mode is in use when the value of the optional packetization-mode MIME parameter is equal to 1 or the mode is turned on by external means. It is primarily intended for low-delay applications. The transmission order of NAL units must comply with the NAL unit decoding order.

-Interleaved mode: The interleaved mode is targeted for systems that do not require very low end-to-end latency. The interleaved mode allows transmission of NAL units out of NAL unit decoding order. This mode is in use when the value of the optional packetization-mode MIME parameter is equal to 2 or the mode is turned on by external means. When different layers of a SVC bitstream are transported in more than one RTP packet stream, the interleaved packetization mode must be used.

The used packetization mode governs which NAL unit types are allowed in RTP payloads. Table 3 summarizes the allowed NAL unit types for each packetization mode.

Type	Packet Type name	Single NAL Unit Mode	Non-Interleaved Mode	Interleaved Mode
0	undefined	Ignore	Ignore	Ignore
1-23	Single NAL unit packet per H.264	Allowed	Allowed	Disallowed
24	STAP-A Single-time aggregation packet	Disallowed	Allowed	Disallowed
25	STAP-B Single-time aggregation packet	Disallowed	Disallowed	Allowed
26	MTAP16 Multi-time aggregation packet	Disallowed	Disallowed	Allowed
27	MTAP24 Multi-time aggregation packet	Disallowed	Disallowed	Allowed
28	FU-A Fragmentation unit	Disallowed	Allowed	Allowed
29	FU-B Fragmentation unit	Disallowed	Disallowed	Allowed

Table 3: Allowed NAL unit types for each packetization mode

- **Decoding Order Number (DON):**

In the interleaved packetization mode, the transmission order of NAL units can differ from the decoding order of the NAL units. Decoding order number (DON) is a field in the payload structure or a derived variable that indicates the NAL unit decoding order.

The DON in [9] applies the following in addition:

When different layers of a SVC bitstream are transported in more than one RTP packet stream (regardless of the use of session or SSRC multiplexing, or a combination thereof), the interleaved packetization mode must be used, and the DON values of all the NAL units must indicate the correct NAL unit decoding order over all the RTP packet streams. If Session multiplexing is used, each session must signal the same value for the (marked as optional, but for this use case mandatory) MIME parameters sprop-interleaving-depth, sprop- max-don-diff, sprop-deint-buf-req, and sprop-init-buf-time. Further these values must be valid for the reception capabilities over all sessions. A receiver must signal the same (marked as optional, but for this use case mandatory) MIME parameter deint-buf-cap for all sessions used for Session multiplexing. (see deliverable 4.2)

- **Single NAL Unit Packet:**

The single NAL unit packet defined here must contain only one NAL unit, of the types defined in Table 1. This means that neither an aggregation packet nor a fragmentation unit can be used within a Single NAL unit Packet. A NAL unit stream composed by decapsulating single NAL unit packets in RTP sequence number order must conform to the NAL unit decoding order. The structure of the single NAL unit packet is shown in figure Figure 12.

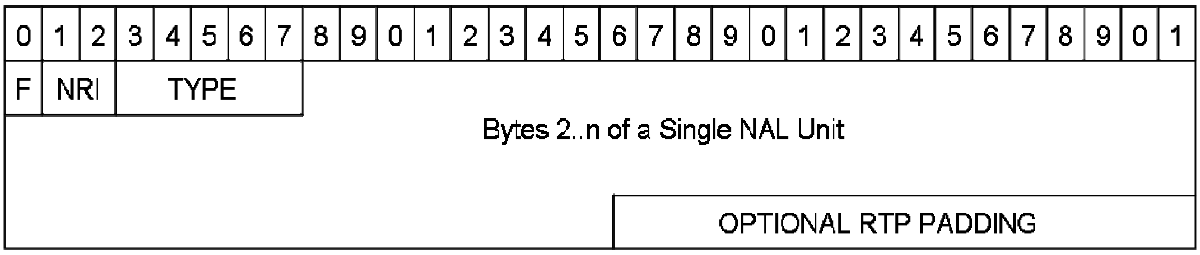


Figure 12: Single NAL Unit payload format

- **Aggregation Packets:**

This packetization mode is introduced to reflect the dramatically different MTU sizes of two key target networks: wireline IP networks (with an MTU size that is often limited by the Ethernet MTU size; roughly 1500 bytes), and IP or non-IP based wireless communication systems with preferred transmission unit sizes of 254 bytes or less. To prevent media transcoding between the two worlds, and to avoid undesirable packetization overhead, a NAL unit aggregation scheme is introduced.
Two types of aggregation packets are defined by this specification:

- **Single-time aggregation packet (STAP):** aggregates NAL units with identical NALU-time. Two types of STAPs are defined, one without DON (STAP-A) and another including DON (STAP-B).
- **Multi-time aggregation packet (MTAP):** aggregates NAL units with potentially differing NALU-time. Two different MTAPs are defined, differing in the length of the NAL unit timestamp offset.

The term *NALU-time* is defined as the value that the RTP timestamp would have if that NAL unit would be transported in its own RTP packet. Each NAL unit to be carried in an aggregation packet is encapsulated in an aggregation unit.
The structure of the RTP payload format for aggregation packets is presented in next figure.

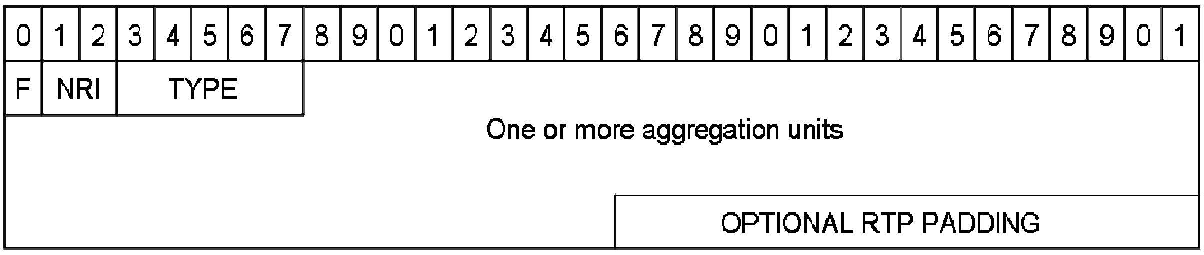


Figure 13: RTP payload format for aggregation packets

Next, only STAP-A aggregation units and their characteristics are shown:

- **Single-Time Aggregation Packet:** Single-time aggregation packet (STAP) should be used whenever NAL units are aggregated that all share the same *NALU-time.*

The payload of an STAP-A does not include DON and consists of at least one single-time aggregation unit, as presented in Figure 14.

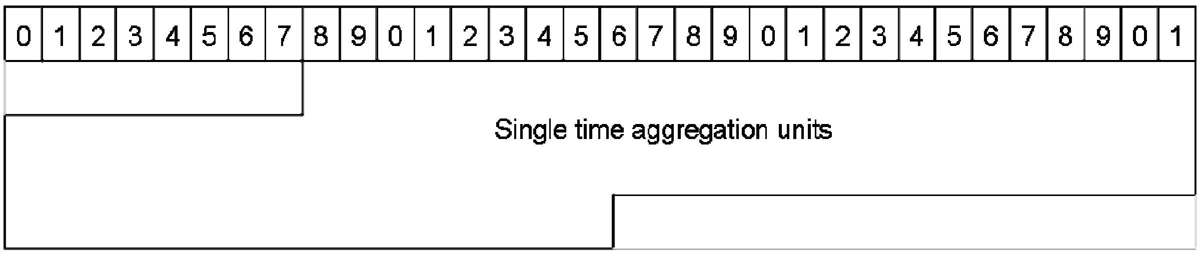


Figure 14: payload format for STAP-A

The payload of an STAP-B consists of a 16-bit unsigned decoding order number (DON) (in network byte order) followed by at least one single-time aggregation unit.

A single-time aggregation unit consists of 16-bit unsigned size information that indicates the size of the following NAL unit in bytes (excluding these two octets, but including the NAL unit type octet of the NAL unit), followed by the NAL unit itself, including its NAL unit type byte. A single-time aggregation unit is byte aligned within the RTP payload, but it may not be aligned on a 32-bit word boundary. Figure 15 presents the structure of the single-time aggregation unit.

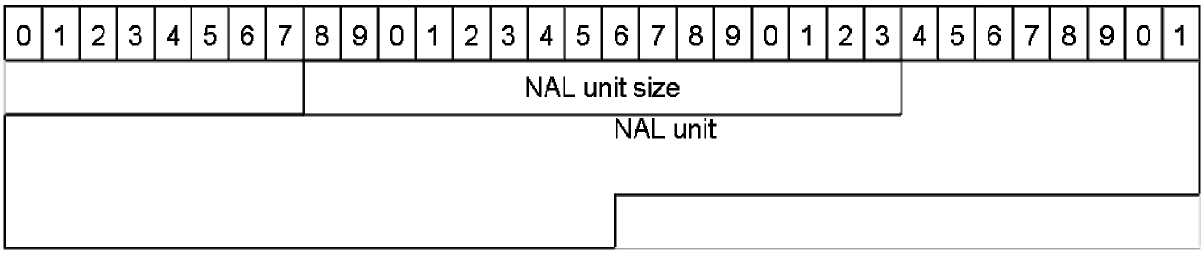


Figure 15: structure for single-time aggregation unit

- **Fragmentation Units (FU):**
  This payload type allows fragmenting a NAL unit into several RTP packets. Doing so on the application layer instead of relying on lower layer fragmentation has the following advantages:
  - The payload format is capable of transporting NAL units bigger than 64 kbytes over an IPv4 network. This is useful to deliver High Definition formats (there is a limit of the number of slices per picture, which results in a limit of NAL units per picture, which may result in big NAL units).
  - The fragmentation mechanism allows fragmenting a single Picture and applying generic forward error correction.

  Fragmentation is defined only for a single NAL unit and not for any aggregation packets. A fragment of a NAL unit consists of an integer number of consecutive octets of that NAL unit. Each octet of the NAL unit must be part of exactly one fragment of that NAL unit. Fragments of the same NAL unit must be sent in consecutive order with ascending RTP sequence numbers (with no other RTP packets within the same RTP packet stream being sent between the first and last fragment).
  Similarly, a NAL unit must be reassembled in RTP sequence number order. When a NAL unit is fragmented and conveyed within fragmentation units (FUs), it is referred to as a fragmented NAL unit. STAPs and MTAPs must not be fragmented. FU must not contain another FU. The RTP timestamp of an RTP packet carrying an FU is set to the NALU time of the fragmented NAL unit.
  An FU-A consists of a fragmentation unit indicator of one octet, a fragmentation unit header of one octet, and a fragmentation unit payload. See Figure 16.

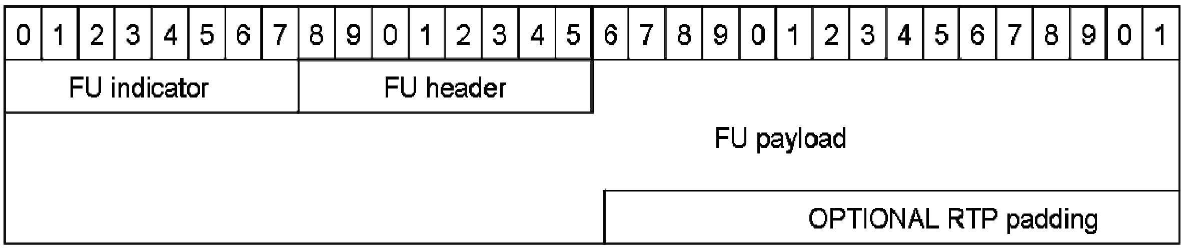


Figure 16: RTP payload format for FU-A

An FU-B consists of a fragmentation unit indicator of one octet, a fragmentation unit header of one octet, a decoding order number (DON) (in network byte order), and a fragmentation unit payload. NAL unit type FU-B must be used in the interleaved packetization mode for the first fragmentation unit of a fragmented NAL unit. NAL unit type FU-B must not be used in any other case.

The FU indicator octet has the following format:

0	1	2	3	4	5	6	7
F	NRI		TYPE				

Figure 17: NAL Unit Octet

Values equal to 28 and 29 in the Type field of the FU indicator octet identify an FU-A and an FU-B, respectively. The value of the NRI field must be set according to the value of the NRI field in the fragmented NAL unit.
The FU header has the following format:

0	1	2	3	4	5	6	7
S	E	R	TYPE				

Figure 18: FU header

S: (1 bit) If set to one, the Start bit indicates the start of a fragmented NAL unit. When the following FU payload is not the start of a fragmented NAL unit payload, the Start bit is set to zero.
E: (1 bit) If set to one, the End bit indicates the end of a fragmented NAL unit, i.e., the last byte of the payload is also the last byte of the fragmented NAL unit. When the following FU payload is not the last fragment of a fragmented NAL unit, the End bit is set to zero.
R: (1 bit) The Reserved bit must be equal to 0 and must be ignored by the receiver.
Type: (5 bits) The NAL unit payload type as defined in table 2.

The FU payload consists of fragments of the payload of the fragmented NAL unit so that if the fragmentation unit payloads of consecutive FUs are sequentially concatenated, the payload of the fragmented NAL unit can be reconstructed. The NAL unit type octet of the fragmented NAL unit is not included as such in the fragmentation unit payload, but rather the information of the NAL unit type octet of the fragmented NAL unit is conveyed in F and NRI fields of the FU indicator octet of the fragmentation unit and in the type field of the FU header.
If a fragmentation unit is lost, the receiver should discard all following fragmentation units in transmission order corresponding to the same fragmented NAL unit. A receiver in an endpoint or in a MANE may aggregate the first n-1 fragments of a NAL unit to an (incomplete) NAL unit, even if fragment n of that NAL unit is not received. In this case, the forbidden_zero_bit of the NAL unit must be set to one to indicate a syntax violation.

- **Payload Content Scalability Information (PACSI) NAL Unit:**

A new NAL unit type is specified in [9], and referred to as payload content scalability information (PACSI) NAL unit. The PACSI NAL unit, if present, must be the first NAL unit in an aggregation packet, and it must not be present in other types of packets. The PACSI NAL unit indicates scalability characteristics that are common for all the remaining NAL units in the payload, thus making it easier for MANEs to decide whether to forward or discard the packet. Senders may create PACSI NAL units and receivers can ignore them.
When the first aggregation unit of an aggregation packet contains a PACSI NAL unit, there must be at least one additional aggregation unit present in the same packet. The RTP header fields are set according to the remaining NAL units in the aggregation packet.
The structure of PACSI NAL unit is exactly the same as the four-byte SVC NAL unit header specified in 3.2.4.1 section of this document. The values of the fields in PACSI NAL unit must be set as details the specification.

3.2.4.3 Scenarios according RTP SVC draft:

The SVC draft presents a subset of scenarios which considers that are clearly in scope of IETF work.
Three main scenarios are considered:

- **Layered multicast**: all layers are individually conveyed in their own RTP packet streams, each carried in its own RTP session using the IP (multicast) address and port number as the single demultiplexing point. Receivers "tune" into the layers by subscribing to the IP multicast (IGMP). DVB-H/T.
  Layered Multicast has the great advantage of simplicity and easy implementation. However, it has also the great disadvantage of utilizing many different transport addresses. So, the receiving client endpoints need to open many ports to IP multicast addresses in their firewalls. This is a practical problem from a firewall/NAT viewpoint.
  When one base and one enhancement layer is in use and are being conveyed separately, that represents one operation point of layered multicast.
- **Streaming of an SVC scalable Stream**: a streaming server has a repository of stored layers for a given content. This generates an scalable stream according client capabilities. Multicast and unicast serving is supported. At the same time, the streaming server may use the same repository of stored layers to compose different streams (with a different set of layers) intended for different audiences.
  In this scenario every endpoint receives only a single SVC RTP session. Thus, only a single firewall pinhole is required, solving the NAT requirements of the Layered multicast scenario.
  When the streaming server learns about congestion, it can reduce sending bitrate by choosing fewer layers when composing the layered stream.
  This payload format is designed to allow for bandwidth flexibility in the mentioned sense, both for CGS and FGS layers. While, in theory, a transcoding step could achieve a similar dynamic range, the computational demands are impractically high and video quality is typically lowered -- therefore, few (if any) streaming servers implement full transcoding.
  Final application oriented to Wimax streaming on demand service type.
- **Multicast to MANE, SVC scalable stream to endpoint**: In this scenario the streaming server multicasts SVC scalable layers, instead of simulcasting different representations of the same content at different bit rates. A gateway receives all multicast layers and creates a single SVC scalable bit stream, according every terminal capabilities.
  The gateway need to understand the signaling and be able to terminate the multicasted layered RTP sessions coming in from the core network side, and create new RTP sessions to the endpoints connected to them. Final application oriented to mobile TV.

3.2.4.4 Considerations of specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols v.1.2.1 [ETSI TS 102 005]

This draft addresses the use of video and audio coding in DVB services delivered over IP protocols. It specifies the use of H.264/AVC [1] video, VC-1, HE AAC v2, AMR-WB+, AC-3 and Enhanced AC-3 for DVB conforming delivery in RTP packets over IP networks.
From this document, some recommendations for the H.264/AVC delivery can be extracted:
For transport over IP, the H.264/AVC data is packetized in RTP packets using [7]. The **Single NAL Unit Mode** or the **Non-Interleaved Mode** of [7] shall be used for the packetization of H.264/AVC data into RTP.
Each frame rate allowed by the applied H.264/AVC Profile and Level may be used. The maximum time distance between two pictures should not exceed 0,7 s.
Where channel change times are important it is recommended that sequence and picture parameter sets are sent together with a random access point at least once every 500 ms. In applications where channel change time is an issue but coding efficiency is critical, it is recommended that random access points are encoded at least once every 2 s. For those applications where channel change time is not an issue, it is recommended that random access

points are encoded at least once every 5 s. In addition It is strongly recommended that the random access point is either an I or an IDR picture. When changing sequence or picture parameter sets, it is recommended to use different values for seq_parameter_set_id or pic_parameter_set_id from the previous active ones. It is strongly recommended that the random access point includes exactly one SPS (that is active), and the PPS that is required for decoding the associated picture
In systems where time-slicing is used, it is recommended that each time-slice begins with a random access point.

3.2.5 RTP Stack Libraries

3.2.5.1 Introduction

Currently, it exists several solutions that supports the RTP specification [6] and [8]. These solutions often are presented in a generic stack library form. Almost of them are intended to deliver VoIP and only a few ones are oriented to deliver video usually being non-commercial solutions.
The aim of this section is to determine what kind of solutions for RTP video encapsulation can be found nowadays, and how SUIT can contribute to the existing solutions, updating and adapting it to the brand new H.264/SVC/MD specification.

3.2.5.2 Libraries comparative

There are a lot of characteristics that can be evaluated to determine the best implementation solution for SUIT project. In this comparative six common characteristics have been taken into account:

- *Programming language:* The library has to be programmed in a language compatible with all the SUIT applications.
- *Operative System:* The library has to be compatible with the maximum types of operative systems.
- *License:* This characteristic evaluates the kind of license of the library. This characteristic would determine the possible utilization of the library in the SUIT project.

    - GPL: The GPL grants the recipients of a computer program the following rights:
        - the right to run the program, for any desired purpose.
        - the right to study how the program works, and modify it. (Access to the source code is a precondition for this)
        - the right to redistribute copies.
        - the right to improve the program, and release the improvements to the public. (Access to the source code is a precondition for this)
    - LGPL: The main difference between the GPL and the LGPL is that the latter can be linked to (in the case of a library, 'used by') a non-(L)GPLed program, which may be free software or proprietary software. This non-(L)GPLed program can then be distributed under any chosen terms, provided the terms allow **"modification for the customer's own use and reverse engineering for debugging such modifications."**

- *Complexity:* This parameter evaluates the use and update complexity of the library.
- *Projects:* This parameter evaluates the popularity of the library and the possible problems and bugs that can be found.
- *Functionalities:* Here the library advantages and disadvantages are revealed, such as protocols or specification supported.

- **RTPlib 3.0 (Lucent/BellLabs)**
    - Programming language: C.
    - Operative System: Unix, Windows, Solaris.

- License: Non-exclusive Source License and Non-commercial License. This license doesn't allow the commercial use and it has some limitations if you want to modify the source code.
- Complexity: This library only provides some functions written in C language. The RTP Library provides a high level interface.
- Projects: Not found.
- Functionalities:
  - Advantages:
    - It supports encryption.
    - API well documented (html).
    - Includes RTCP scalability algorithms.
  - Disadvantages:
    - Header extensions not supported.

- **Vovida RTP Stack (RTP 1.5.0)**
  - Programming language: C++.
  - Operative System: Unix, Windows.
  - License: Vovida Open Communication Application Library (VOCAL).
  - Complexity: Documentation has to be found in forums.
  - Projects: Not found.
  - Functionalities:
    - Advantages:
      - End to end network transport of any audio and video format
      - Unicast service only
      - Sending RTCP status and quality reports
      - Sending RTCP SDES information
      - Resolves lost packets, jitter, and out of sequence packets
      - DTMF in RTP (RFC 2833)
    - Disadvantages:
      - Unicast service only
      - Limited RTCP SDES information

- **ViTooKi(Video Tool Kit)**
  - Programming language: C++.
  - Operative System: Linux-iPAQs and Windows.
  - License: GPL
  - Complexity: The complete package consists of the main library, several small examples that illustrate how to use the library, and some larger multimedia applications like an adaptive media server, an adaptive media proxy, a multi video player, etc. Each of these applications has a separate webpage where more detailed information is available.
  - Projects:
    - SimplePlayer
    - SimpleRTP
    - DvdLeech
  - Functionalities:
    - Advantages:
      - standard compliant video streaming via RTP/UDP
      - standardized RTP extensions to allow intelligent retransmission of lost frames
      - RTSP and HTTP support
      - real-time adaptation according to the client's terminal capabilities,
      - meta-data support: MPEG-21 is used for describing terminal capabilities and user preferences, MPEG-7 for adding semantic information to a video (e.g.: actors in a specific scene of a video)

        - MPEG-1, MPEG-2, MPEG-4 and any other video format supported by the ffmpeg or xvid library
        - OGG Vorbis/Theora support
        - Containers like .mp4, .avi are supported by ffmpeg, .mkv, .ogg or (if available) .mp4 via ISOMP4
        - MP3 audio is decoded via ffmpeg or via libmad (optimized for ipaq)
        - Well documented (Doxygen)
    - Disadvantages:
        - Better lip sync, CPU load

- **Catra Libraries**
    - Programming language: C++.
    - Operative System: Linux, Unix, Solaris and Windows (with some bugs).
    - License: GPL
    - Complexity: It includes an open, extensible multimedia streaming server, a GUI to handle all the servers distributed in sites and some MP4 tools to handle the media files. It provides a final solution for the MP4 streaming. Too many changes needed to adapt it to H264/SVC.
    - Projects:
        - Catra Streaming Platform
    - Functionalities:
        - Advantages:
            - Supports RTP/RTCP, RTSP and SDP.
            - The server supports MP4.
            - ISMA and 3gpp streaming.
            - Well documented (Doxygen)
        - Disadvantages:
            - It has some bugs in the Windows version.
            - MP4 based. (not generic)

- **UCL Multimedia Library**
    - Programming language: C.
    - Operative System: Linux, Irix, FreeBSD, MACOSX and Windows.
    - License: Not known.
    - Complexity: Basic C functions.
    - Projects:
        - Not known.
    - Functionalities:
        - Advantages:
            - Supports RTP/RTCP and SDP.
            - Bad documented (html)
        - Disadvantages:
            - Last code revision was in 2003.
            - The documentation is up to date. (2001)

- **JRTPLIB3.6.0(June 2006)**
    - Programming language: C++.
    - Operative System: Linux, Unix,Windows…
    - License: Totally free.
    - Complexity: The structure of the library is very modular. The source code easy to understand and easily extensible. High Level classes to establish RTP sessions and control all session parameters easily.
    - Projects:
        - Several projects use it.
        - Libraries like EMILIB and JVoIPLIB has been extended from it.
    - Functionalities:

        - Advantages:
            - Supports RTP/RTCP (RFC 3550).
            - RTP Header Extensions.
            - Implements a hash table to improve real-time performance.
            - Very Well documented (Doxygen)
        - Disadvantages:
            - It doesn't support RTSP.
            - It doesn't support SDP.

- **Live555 media streaming**
    - Programming language: C++.
    - Operative System: Linux, Unix,Windows…
    - License: LGPL.
    - Complexity: The structure of the library is very modular. The library currently supports different source codecs. Actually there are people working in the H264 codec, so it seems viable to adapt it to the H264/SVC standard. The library has too many classes and there are lot of inheritances between classes, so it is necessary a hard study of the library before start any implementation.
    - Projects:
        - VLC.
        - LiveCaster
        - VobStreames
    - Functionalities:
        - Advantages:
            - Supports RTP/RTCP (RFC 3550).
            - RTSP compliant
            - SIP/SDP compliant
            - Very well documented (Doxygen)
        - Disadvantages:
            - H264 is not implemented yet.

3.2.5.3 Proposed stack library for SUIT

After reviewing the different libraries, we concluded that **Jrtplib-3.6.0** and **Live555** libraries seem to be the best candidates for our implementation due its kind of licenses and its easy extensibility. Both libraries are very modular and have been successfully tested in several projects, so both are good candidates for the SUIT project.
Jrtplib-3.6.0 is a very intuitive library and it supplies to user a set of high level functions to establish and control sessions and users. This library is easy to adapt to any payload, due it is a generic RTP payload library. On the other hand, its weak point is that it hasn't implemented any control over RTSP protocol and it doesn't support SAP and SDP protocols. This last issue implies the adaptation of new classes for these three protocols, so this disadvantage in principle discards this library for the play-out RTP encapsulation. Although, this library could be a good option to easy implement a client for RTP dencapsulation.
The Live555 media streaming forms a set of C++ libraries for multimedia streaming, using open standard protocols such as RTP/RTCP, RTSP and SDP. The current libraries implemented can stream, receive, and process MPEG, H.263+ or JPEG video, and several audio codecs. This library is very modular too, but it has a lot of classes and inheritances, so any modification implies the adaptation of several high level classes of the library. The main advantage is given by the different signaling protocols supported and the utilization of the library in current working projects such as "VLC" and "MPlayer". Nowadays, there are people working in the H264 adaptation of this library, but it is not functional yet.
In conclusion, the **Live555 library** seems to be the best solution for SUIT project RTP H264/SVC/MD encapsulation. The idea is to implement some classes in order to make the library H264 /SVC compatible and modify the library to receive the information from the extractor module.

3.2.6 Proposed RTP encapsulation solution for SUIT

3.2.6.1 Introduction

After reviewing the more state-of-the-art specifications and recommendations available for the H.264/SVC in previous sections and once decided the stack library we will use, some decisions must be taken.
The SUIT requirements for RTP Encapsulator are detailed in section 3.2.6.2.
The communication between Extractor and Encapsulator is defined in section 3.2.6.4. The extractor will give the information about the two descriptions to the encapsulator.
Owing to there is no H264/SVC specification for Multiple Description, SUIT project will have to extend-adapt the current specifications, in order to encapsulate the H.264/SVC streams and deliver it into different descriptions. Thus, the receiver will be able to dencapsulate the incoming RTP descriptions and combine them to obtain the main stream. The decisions taken are detailed in section 3.2.6.6.

3.2.6.2 RTP Encapsulator features

The RTP Encapsulator has to be developed as a software code , including modified live555 library and the Extractor library which will feed it with the two descriptions payload. As detailed in Figure 19, the Encapsulator and the Extractor are implemented in the same application. We will have one Extractor-Encapsulator module for each video (two descriptions).

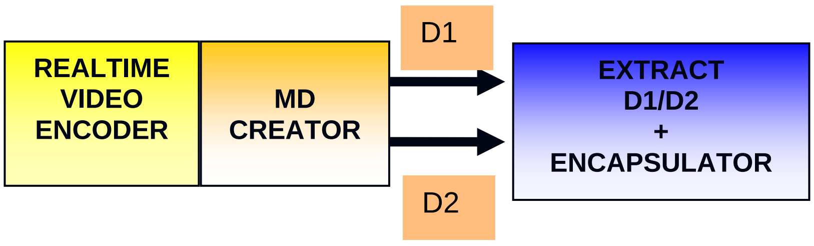


Figure 19: Encapsulator and Extractor modules

The RTP Encapsulator will have to support the following features**:**

- Encapsulate NAL units of D1 and D2 over RTP packets and send it over IP.
- Establish two RTP Sessions, one for each description.
- Generates periodically (depends on available bandwidth) RTCP packets for each session. Sender Reports (SR) should be used in order to transport the relationship of RTP timestamp and clock reference (wallclock).
- Extractor and RTP encapsulator will be the same application. An RTP/RTCP C++ Library will be used in this application.
- The encapsulator module only has to initialize the extractor module and then gets the NAL units and its associated parameters.

3.2.6.3 RTP Encapsulator modules

The RTP Encapsulator can be divided into several logic modules regarding the task performed. Some of these tasks are performed by the RTP library such as RTP/RTCP Session Manager or the RTSP control.
To encapsulate SVC a new class has been developed. This class is called H264SVCMD and includes all the implementations regarding the packetization and encapsulation modules for H264/SVC/MD.

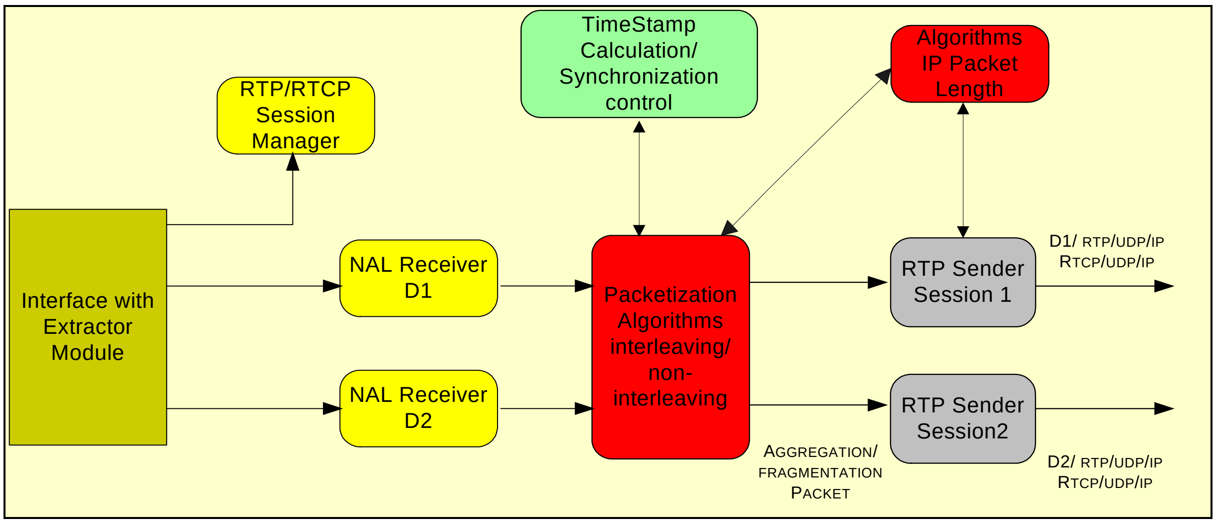

TimeStamp Calculation/ Synchronization control
RTP/RTCP Session Manager
Algorithms IP Packet Length
Interface with Extractor Module
NAL Receiver D1
NAL Receiver D2
Packetization Algorithms interleaving/ non-interleaving
RTP Sender Session 1
RTP Sender Session2
D1/ RTP/UDP/IP RTCP/UDP/IP
D2/ RTP/UDP/IP RTCP/UDP/IP
AGGREGATION/ FRAGMENTATION PACKET

Figure 20: RTP Encapsulator process

3.2.6.4 Interface with Extractor module (NAL receiver module)

The aim of this module is to receive the different types of NALs and all the information related to each NAL from the extractor. The Encapsulator needs some information to encapsulate and deliver correctly all the information supplied by the extractor and the video coder. This information will be extracted from different sources:

- From the extractor:
  - Frame number.
  - NAL Size
  - NAL timestamp
  - Frame-rate

- From NAL Header:
  - Payload type
  - NRI (to agregate NAL into an aggregation packet)
  - DID (dependency_id) ,TL (temporal_level), QL (quality_level)

The information required from the Extractor will be supplied through the methods defined in the ExtractorInterface.h.
The Extractor will deliver each NAL unit accompanied by several extra fields, which are originally delivered by the encoder and aggregated in the AnnotatedNalUnit class:

- The length in bytes of the NAL unit;
- The frame number of the Access Unit this NAL unit belongs to;
- The decoding timestamp of the NAL unit;
- The slice number of the NAL unit.

Frame Rate is returned from a function of this same interface.
For the first demo, one NAL will be equal to one frame. In next versions, this statement could change. The extractor will deliver at the same time all NALs for the same frame.
The Extractor module will deliver different types of NALUs to the RTP encapsulator library: NAL_SLICE (1), NAL_IDR_SLICE (5), NAL_SEI (6), NAL_SPS (7), NAL_PPS (8). Specific SVC NAL types (20 and 21) and Scalability Information SEI message (SEI type 22) to signal the total number of layers, quality information etc.

3.2.6.5 H.264/SVC/MD Packetization Algorithms

In the RTP H.264/SVC specifications several types of packetization modes are contemplated (section 3.2.4.2).

The main aim of this module is to provide algorithms to select intelligently the correct packetization mode. This selection will be based on entry parameters such as: frame size, MTU, etc…
The MTU size information will be delivered from the IP packet length algorithms.
After the optimal packetization mode is selected, functions from the RTP encapsulation module will be called. Please, see next section.

3.2.6.6 RTP/RTCP Encapsulation module

This module will contain several functions to encapsulate the NAL units complaining with the RTP H.264/SVC specification. All the intelligence will be held by the packetizations algorithms module.
According to [9] specification only the *Non-Interleaved Mode* and *Interleaved Mode* can be used.
According to the Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols v.1.2.1 recommendation [4], only the *Single NAL Unit Mode* and *Non-Interleaved Mode* must be used.
Thus, for SUIT project it has been decided to use the *Non-Interleaved Mode*. This mode, at glance, supports all required RTP project scenarios.
Interleaving is a technique for resilience against errors in the network, and it is situated ‘below’ the video coding layer. The choice of interleaved vs. non-interleaved is a trade-off between error resilience and delay+complexity.
For SUIT project we will avoid complexity at the receiver side, then it’s another reason to not use interleaved mode.
Moreover we use the non-interleaving mode because an MD module is already employed in order to provide error resilience.

The Encapsulation module will support the following packetization modes for NAL units:

- Single NAL Unit packet
- Aggregation packets (STAP-A)
- Fragmentation packets (FU-A)

Working at 10 Mbps, 25 fps, an average frame costs 50kB. But an I-Frame is above this average and could break the 65kB limits. Then, fragmentation packets should be used for this argument and also for the network IP packet length constraints.
Extra information of each NAL unit (Frame Number and Slice Number) will be conveyed in the extension header of the RTP header as follows:
Concerning frame_number length, in practice, there is no application that is storing and sending streams longer than 24 hours. If it is the case, they are split like for instance in video surveillance. So you may not encounter more than 24 hours x 3600 seconds x 60 Hz frames in a single stream, the result should stand into 3 bytes.
Concerning slice_number length, in a single picture, you cannot have more than slices than macroblocks without taking in account redundant slices. So if you are dealing with video formats sizing up to 1080i/p, you would not have more than (1920 * 1080) / (16 * 16) = 8100 slices in an HD picture, the result should stand into 2 bytes even it is including the presence of redundant slices.
And with HD pictures we could approach the limit of 65Ko for NAL size, but not much more. So 3 bytes should be sufficient.
Maybe for the final RTP Encapsulator, only the Slice_number will be send into the RTP header extension, because the frame_number could be calculed by the terminal using the timestamp and the frame rate.

Next, the packetization modes are explained:

- **Single NAL Unit packet:**

In this mode, each NAL unit belonging to the same frame is encapsulated in a RTP packet keeping same timestamp in all the RTP packets.
The format for the RTP Payload containing a Single NAL Unit packet will be the following:

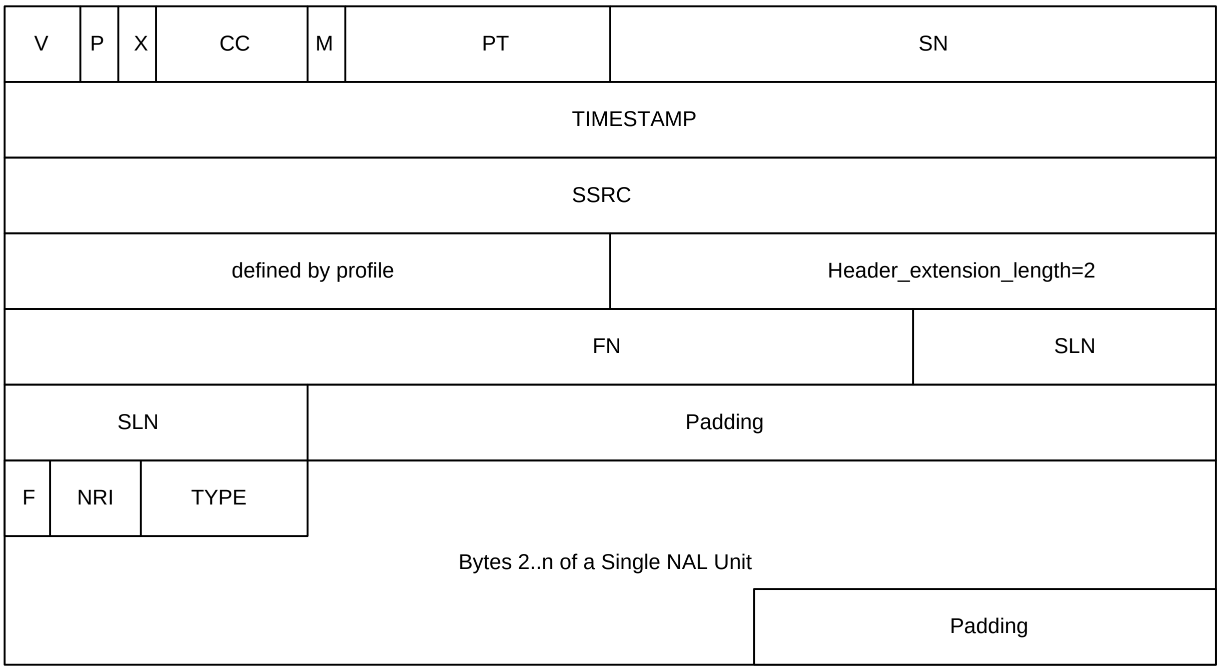


Figure 21: SUIT Single NAL Unit Packet

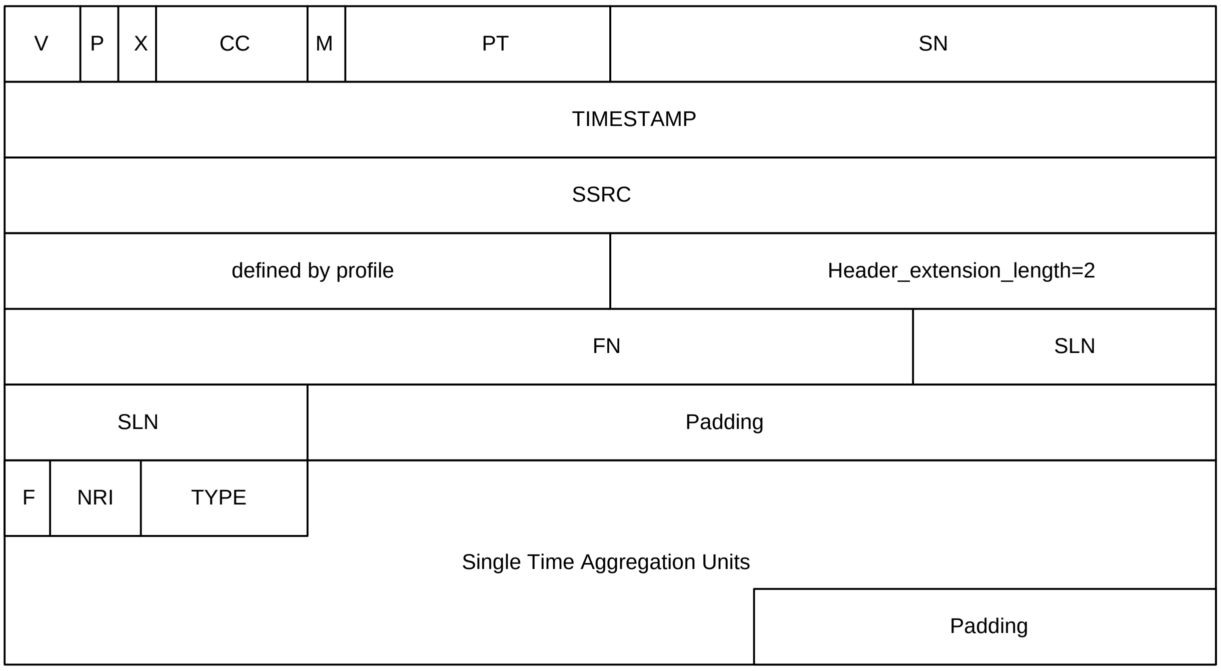
This is the packet format for AVC video. In case of using SVC the Single NAL Unit has extra header bytes according deliverable 5.2.

V = 0 (version)
P = 0
X = it is set if the additional parameters are conveyed in the RTP header extension
CC = 0
M = it is set if the NAL unit conveyed is the last NAL of the frame.
PT = 98 , it is H264/AVC and H264/SVC payload
SN = it is incremented with 1 for each RTP packet
TIMESTAMP = it is the same as the NAL timestamp.
SSRC = it is a random value assigned by the RTP library.
header_extension_lentgh = 2, indicates extension of two bytes.
FN = it is the Frame number from the NAL units belongs to. This value is got for the extractor module.
SLN = it is the Slice number of the slice contained in the NAL unit. This value is got for the extractor module.
F = 0.
NRI = this value is the same of the NAL delivered by the extractor.
TYPE = this value is the same of the NAL delivered by the extractor. Values from 1 to 23.
In case of using SVC instead of AVC, then the NAL unit extension header applies.

- **Single Time Aggregation Packets (STAP-A):**

This mode will be selected by the packetization algorithms if they detect that several NALs from the same frame can be delivered in one RTP packet. This constrain will be limited by the MTU Size for this description. This encapsulation function will receive one NAL unit each time and it will fill the aggregation packet until this will reach the MTU. Then the RTP packet will be sent maintaining the frame timestamp.

The format for the RTP Payload containing a Single Time Aggregation Packets will be the following:



The Single Time Aggregation packets are as follows:

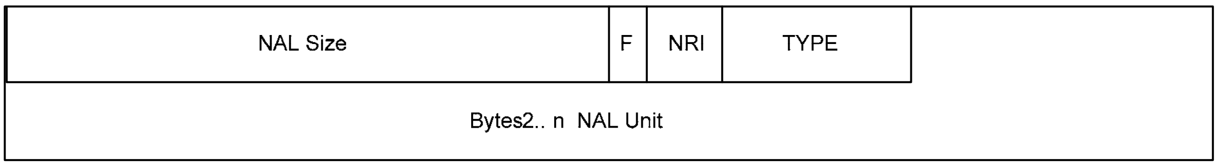


Figure 22: SUIT Aggregation Packets

V = 0 (version)
P = 0
X = it is set if the additional parameters are conveyed in the RTP header extension
CC = 0
M = it is set if one of the NAL units conveyed is the last NAL of the frame.
PT = 98 , it is H264/AVC and H264/SVC payload
SN = it is incremented with 1 for each RTP packet
TIMESTAMP = it is the same as all the aggregated NAL timestamps, because they belongs to the same frame.
SSRC = it is a random value assigned by the RTP library.
header_extension_lentgh = 2, indicates extension of two bytes.
FN = it is the Frame number from the NAL units belongs to. This value is got for the extractor module. This value applies to all the NALs included in the aggregation packet.
SLN = it is the Slice number of the slice contained in the NAL unit. This value is got for the extractor module. This value applies to all the NALs included in the aggregation packet.
F = 0.
NRI = this value is set to the maximum of all the NAL units carried in the aggregation packet.
TYPE = this value is used to identify a STAP-A packet (type = 24).

Then follows one or more Single Time Aggregation Units, with the following syntax:
NAL Size = this value indicates the size of the following NAL unit in bytes (excluding these two octets, but including the NAL unit type octet of the NAL unit)
NAL Header = F,NRI,TYPE (Values from 1 to 23).

In case of using SVC instead of AVC, then the NAL unit extension header applies.
Note: The use of PACSI NAL units to reduce overhead in aggregation packets would be studied to be included in SUIT.

- **Fragmentation Units (FU-A):**

This mode will be selected by the packetization algorithms if they detect one NAL with a greater size than the MTU allows for that description. This encapsulation function will receive the NAL unit and it will fragment it into several RTP packets. All the RTP packets belonging to the same NAL will maintain the same frame timestamp.
The function that performs this encapsulation has the following form:

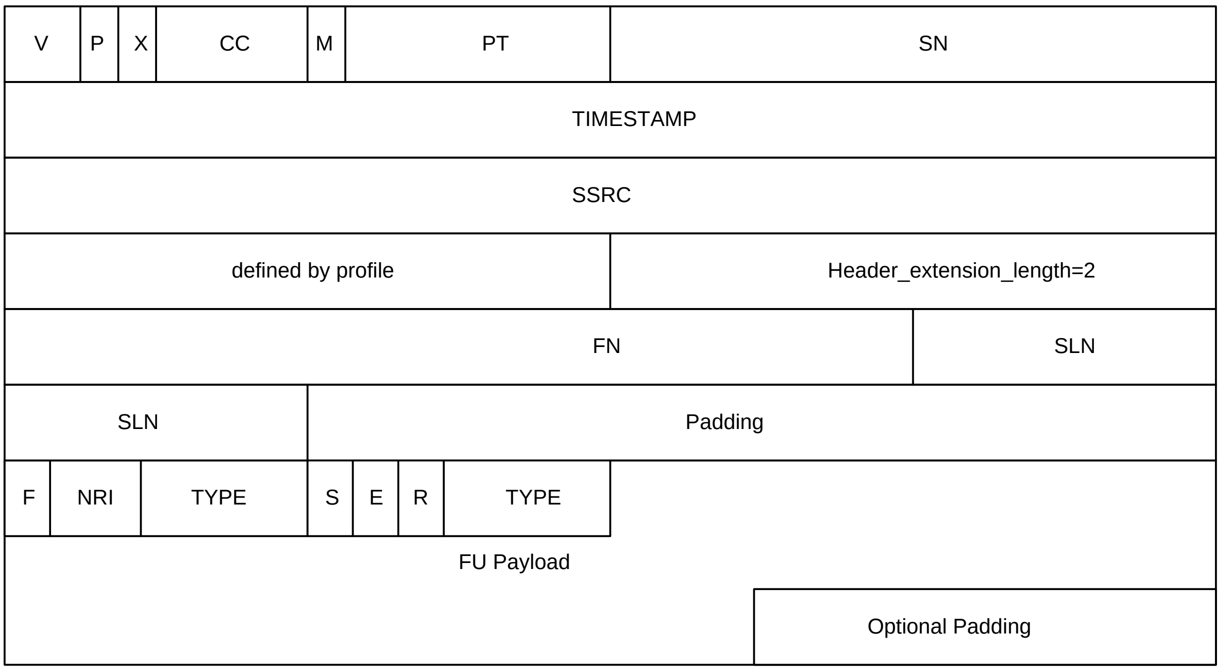


Figure 23: SUIT Fragmentation Packets

V = 0 (version)
P = 0
X = it is set if the additional parameters are conveyed in the RTP header extension.
CC = 0
M = it is set if the NAL unit conveyed is the last NAL of the frame. All RTP packets containing fragments of the NAL will have the marker bit set to 1.
PT = 98 , it is H264/AVC and H264/SVC payload
SN = it is incremented with 1 for each RTP packet
TIMESTAMP = it is the timestamp of the NAL fragmented.
SSRC = it is a random value assigned by the RTP library.
header_extension_lentgh = 2, indicates extension of two bytes.
FN = it is the Frame number from the NAL units belongs to. This value is got for the extractor module.
SLN = it is the Slice number of the slice contained in the NAL unit. This value is got for the extractor module.
F = 0.
NRI = this value is the same of the NAL delivered by the extractor.
TYPE = this value is used to identify a FU-A packet (type = 28).

Then follows the fragmented NAL with the following syntax:
S = it is set if the RTP packet conveys the first NAL fragment.
E = it is set if the RTP packet conveys the last NAL fragment.
R = 0

TYPE = Values from 1 to 23. In case of using SVC instead of AVC, then the NAL unit extension header applies.

In case of using fragmentation packets, the extension header is only present in the RTP Packet that conveys the first part of the Fragmented NAL, (S=1).

3.2.6.7 IP packet length algorithms

The aim of this module is to design algorithms to find out the optimal network IP packet length. These values will be decisive for the packetizations algorithms selection. In this module, results from the simulations in the channel models will be taken account.
Common MTU sizes:

~1500 bytes ethernet packet (wireline IP link)
~256 or less bytes in wireless

TL DID and QL allow to measure and weighten the importance of NALUs and should help to define the appropriate length of RTP packets. A strategy must be defined by testing different schemes from the most simple (fixed-length and no care about layer levels) up to sophisticated ones and then choosing the one providing the best trade-off.

3.2.6.8 Timestamp calculation / Synchronization module

The finality of this module is to control the timings in each session and the RTP timestamp assignment:

TimeStamp: 32 bits. It represents the sampling instant of the first octet of frame data. The RTP clock for H264 ticks at 90kHz.
ex: If we have a 25Hz video the timestamp will increment 3600 in each clock time.

Two possibilities are contemplated:

- NAL unit has no timing properties of its own. Then the RTP timestamp is set to the RTP timestamp of the primary coded picture of the access unit (frame) in which the NAL unit is included.
- Timing information of an Acces Unit delivered in NAL units:
    - The NAL SPS contains an optional section called VUI (Video usability Information).
    - NAL SEI of type “Picture timing”. Indicates whether a picture should be displayed as a frame or one or more fields.

The solution adopted in project SUIT is to use the timestamp supplied by the Extractor for each frame, and all the NALs of one frame will maintain the same timestamp. This timestamp is already assigned to each NAL in the RealTime Encoder.
To synchronize the two descriptions delivered by the Encapsulator it would be not necessary to send RTCP with the SR containing the NTP, due to the two descriptions have been generated in the same application, the receiver only need its own clock at 90kHZ and the timestamp delivered in each RTP packet to synchronize them.
Anyway, RTCP packets would be delivered to maintain compatibility with the standard, and to provide more information to the receiver.

3.3 IP over DVB: DVB-T Encapsulation

The RTP payload has been discussed in Section **Error! Reference source not found.**. In this section, we describe all encapsulation tasks down to the physical layer. Internet packets are transported over DVB network by using the multiprotocol encapsulation (MPE), despite in DVB-S2 another encapsulation procedure has been introduced, GSE-Generic Stream Encapsulation. MPE makes use of sections where sections are also transported on TS. The data broadcast specification profile for MPE supports data broadcast services that requires the transmission of datagrams of communication protocols via DVB compliant broadcast networks. The transmission of datagrams according to the multiprotocol encapsulation specification is done by encapsulating the datagrams in DSM-CC sections, which are compliant with the MPEG-2 private section format. Figure 24 shows the encapsulation procedure in which the MPE section datagram is generated at the DVB BST. It encapsulates an IP/UDP/RTP packet.

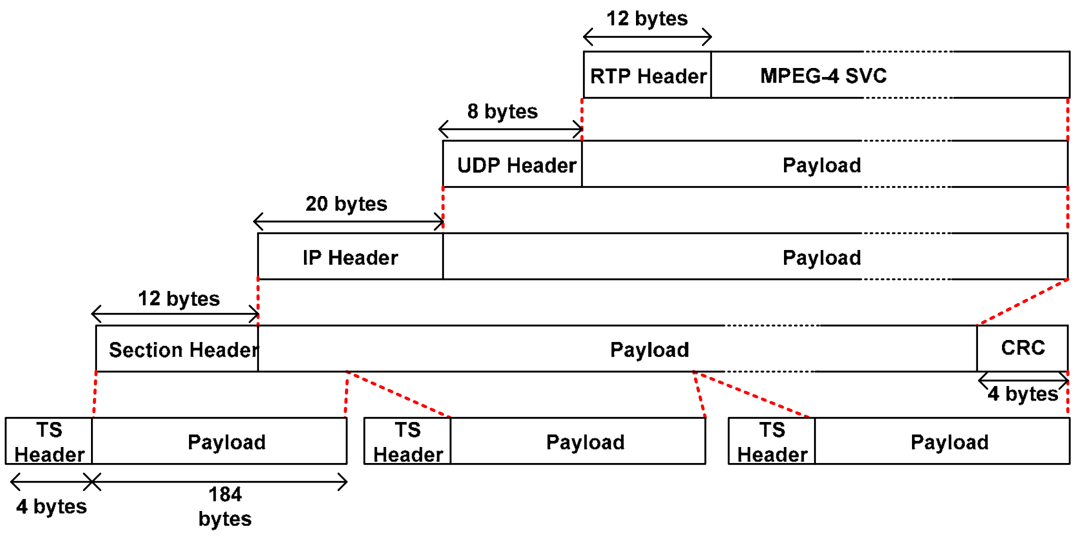


Figure 24: IP/UDP/RTP over Sections/TS.

The presence of a multiprotocol data stream in a service shall be indicated by the Stream Type and Table ID. The STB removes the TS headers and MPE section headers and optionally calculates the CRC. As the TS/MPE encapsulation is done at the DVB BST, the IP/UDP/RTP/MPEG-4 SVC packet is sent to the Switch (see Figure 3), obviously encapsulated on a link layer packet, Ethernet.

The figure below shows the data flow in the Down Link:

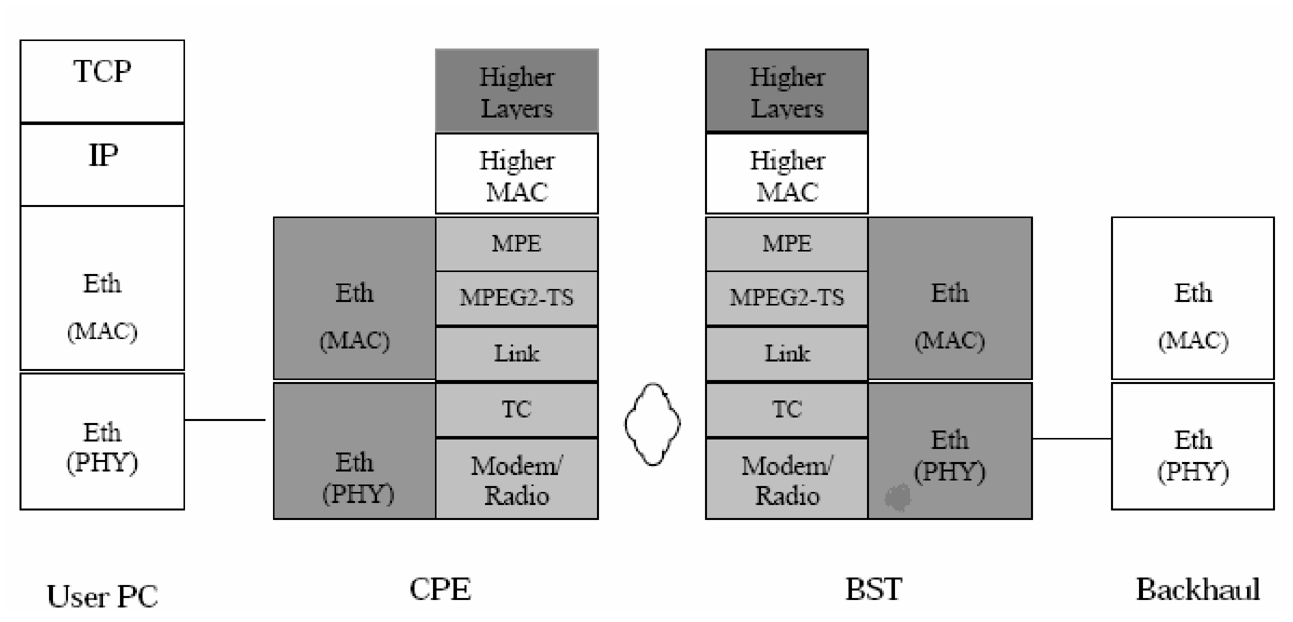
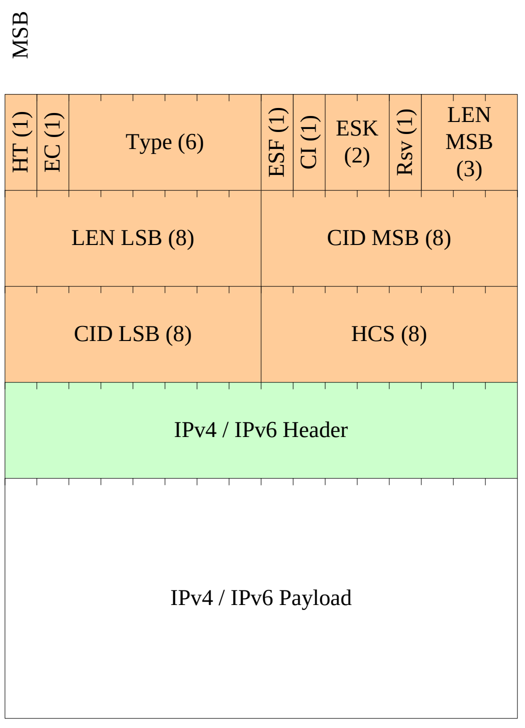


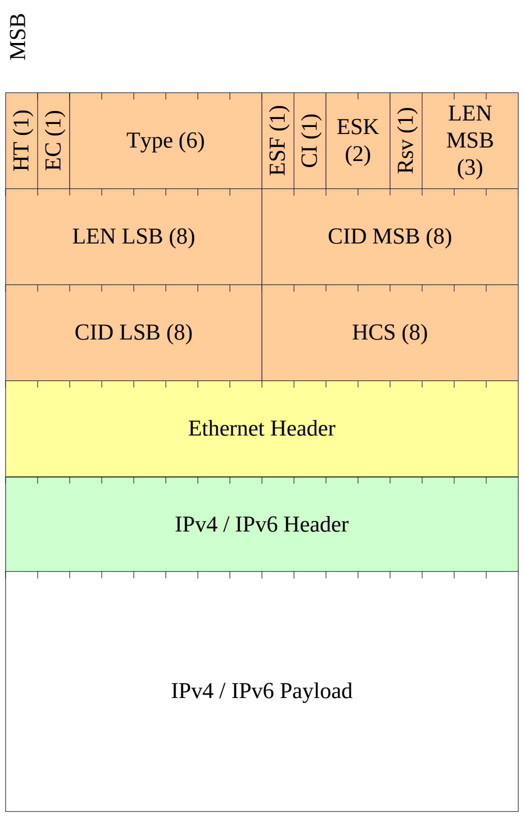
Figure 25: Data Flow in the Down Link

3.4 IP over 802.16: WIMAX Encapsulation

This section specifies the frame format for transmission of IP packets over Wimax network.
IEEE 802.16 0**Error! Reference source not found.** defines a new air interface and medium access control (MAC) protocol for the provisioning of high data rate services over large cell coverage.
The frame format for the MAC PDU (Packet data Unit), in the IEEE 802.16 U interface, consists of a 6-byte MAC header, various optional sub-headers, and an IP data payload as shown in Figure 26 (a). When the Ethernet CS is used (for A interface), an Ethernet header should be inserted between the MAC header and the IP header 00 as shown in Figure 26 (b). The meanings of the Generic MAC header are as follows:
HT: Header Type (1 bit). This should be set to zero indicating that it is a Generic MAC PDU.
EC: Encryption Control (1 bit). 0 = Payload is not encrypted; 1 = Payload is encrypted.
Type (6 bit). This field indicates the sub-headers (such as fragmentation, packing etc.) and special payload types (ARQ feedback) present in the message payload.



(a)



(b)

Figure 26 Frame format for (a) Reference point U (b) Reference point A

ESF: Extended Sub-header Field (1 bit). 0 = The extended sub-header is absent; 1 = The extended sub-header is present. The ESF is applicable both in the downlink and in the uplink.
CI: CRC Indicator (1 bit). 1 = CRC is included in the PDU by appending it to the PDU Payload after encryption, if any. 0 = No CRC is included.
EKS: Encryption Key Sequence (2 bit). The index of the traffic encryption key and initialization vector used to encrypt the payload. This field is only meaningful if the EC field is set to 1.
Rsv: Reserved (1 bit). It should be set to zero.
Length (11 bit). The length in bytes of the MAC PDU including MAC header and the CRC.

CID: Connection IDentifier (16 bit). The CID in the generic MAC header identifies a connection to equivalent peers in the MAC of the BS and SS (Subscriber Station).
HCS: Header Check Sequence (8 bit). An 8-bit field used to detect errors in the header. The transmitter shall calculate the HCS value for the first 40 bits of the IEEE 802.16 MAC header, and insert the result into this field. The HCS value is the remainder of the division by the generator polynomial that is g(x) = X8 + X2 + X + 1.
CRC: Cyclic Redundancy Check (32 bit). A MAC PDU may contain a CRC. Implementation of CRC capability is mandatory for OFDM. CRC = 1 indicates the presence of CRC in the MAC PDU. The generator polynomial is g(x) = X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1. The CRC shall be calculated after encryption; i.e., the CRC protects the generic header and the ciphered payload.

3.5 DVB-IPI Encapsulation

The MHP IPTV scenario in SUIT will be based on the DVB-IPI [11] and DVB-MHP [12] standards. The following subsections summarize the encapsulation of service metadata and media streams as defined by these documents.

3.5.1 Service Announcement and Service Information

The DVB-IP handbook proposes SD&S (Service Discovery and Selection) for announcing services and for providing terminals and users with enough information to access available services. The service provider can choose between two different SD&S delivery models: The *push* model is based on IP multicast while the *pull* model uses http requests.
In either case the terminal, which is called *HNED* (Home Network End Device) in DVB-IPI, has to look up an entry point (IP address) providing the necessary SD&S information. The look up procedure and more details on SD&S for SUIT will be described in a dedicated section of Deliverable 4.2: “Session and Description Protocols”.
The service information (called *SD&S record* in DVB-IPI) is formatted using an XML-based scheme which is defined in the DVB-IP handbook. In case of the multicast delivery method, the SD&S records are encapsulated in DVBSTP (DVB SD&S transport protocol) sections.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|Resrv|Enc|C|              Total_Segment_Size               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload ID    | Segment ID                    |Segment_Version|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Section_Number        | Last Section Number   |Compr|P|HDR_LEN|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 (Conditional) ServiceProviderID               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                 (Optional) Private Header Data                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                             payload                           :
|                                               +-+-+-+-+-+-+-+-+
|                                               |(Optional) CRC |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           (Optional)    CRC (Cont)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 27: Syntax SD&S multicast delivery protocol

The DVBSTP section header contains information about the type (Payload ID), the ID of the record (Segment ID) and the version of the record (Segment Version). This header enables the terminal to decide whether to reconstruct the SD&S record from the transmitted section or to ignore it. This will reduce the costs when monitoring the SD&S records for updates. The size of a section is limited to the maximum payload of a UDP packet, however, the handbook recommends avoiding

fragmentation of sections in the underlying layers. Thus, for Ethernet, this limitation reduces the payload size of a section to 1452 bytes.
If using the pull model, the HTTP [13] protocol is used for the communication between the HNED and the SD&S server. In order to obtain the necessary service information, the HNED sends a HTTP request to the server. The consequent HTTP response then contains the necessary SD&S record.

3.5.2 Encapsulation of Live Broadcast Services

For the time being (end of 2006), the DVB-IP handbook [11] only covers services based on the MPEG2-TS [14]. Version 1.2.1 of the DVB-IP handbook (phase 1.2) has been released in November 2006. The IPTV group of the DVB commercial module is currently discussing commercial requirements for DVB-IP phase 2. For that phase the standard will support the direct encapsulation of audio and video in RTP. The motivation behind this is to avoid the overhead in the MPEG-2 system layer protocol stack. Until phase 1.2, the encapsulation of MPEG2-TS into RTP [6] remains mandatory. For the upcoming phase 1.3, it has been agreed to additionally allow direct encapsulation of the transport-stream in UDP as defined in [16]. Current IPTV service providers have the delivery network under their control and prioritize broadcast services over other data. Other services, such as Internet traffic and Voice over IP, will not interfere with TV transmissions. In such a controlled environment the use of RTP is not advantageous but causes unnecessary overhead.
DVB-IPI adds no restriction on how to encapsulate services in MPEG2-TS. It is possible to use a multi-program transport stream (MPTS) with 38 Mbit/s as it is common for DVB-S. Nevertheless, it is unlikely to see MPTS on IPTV. Most probably, there will be only single-program transport streams (SPTS) which contain only one service.
The MHP-IPTV part of SUIT implements version 1.2.1 of the DVB-IPI standard (Phase 1.2), i.e., video and audio content is encapsulated in the MPEG-2 transport stream in compliance with [17]. The transport-stream packets are transmitted over RTP [6]. For a detailed description, please refer to section 7 of the DVB-IP handbook [11].

3.5.3 Transmission of Interactive Applications (DVB-MHP)

SUIT will implement an interactive application which allows a user to select and watch a streaming video (Video-on-Demand) from within a broadcast TV service. To do so, on user request, an MHP application has to provide and display a list of hyperlinks to video clips available on a streaming server.
Concerning the encapsulation of MHP applications, the signaling and the transmission of application resources have to be discussed and defined. MHP applications are signaled either via the AIT (Application Information Table of the transport stream) or, for IPTV systems, within the SD&S record of the related broadcast service. The application resources (e.g. binaries, graphics, etc.) can be either transferred in an object carousel [15] inside the transport stream or downloaded by the terminal from a HTTP server [13].

4 Synchronization

4.1 Introduction

As explained in chapters before, the extractor module supplies to the RTP Encapsulator two descriptions of the same H.264/SVC video. These two descriptions have to be encapsulated in such a way the Gateway would be able to combine them successfully obtaining the desired video stream. The Encapsulator would have to add some additional information in the RTP packets and use some mechanism to deliver a common reference clock to the same session. These points will be achieved modifying the RTP timestamp field and using the associated control protocol of the RTP protocol.
So, in this chapter the associated control protocol “RTCP” is described.

4.2 H264/AVC SVC timing reference information

A NAL SPS contains an optional section called VUI (Video Usability Information). the optional timing_info_present_flag enables the coding - inside the SPS - of 3 parameters: num_units_in_tick, timescale, fixed_frame_rate_flag. The 2 first are used to determine the timestamp of a frame.
Moreover a SEI of type “Picture timing” contains information about the current time position. This SEI can be sent to refresh the time, for instance at IDR NAL (equivalent to start of GOP).
Timestamp is set to the sampling instant of the picture (Acces Unit) the NAL unit belongs to (or presentation timestamp if sampling instant is not known). Sampling instant is known in case of real time encoding.
The contents of the clock timestamp syntax elements indicate a time of origin, capture, or alternative ideal display.
With the VUI parameters, the SEI Picture timing and the frame_num of the slice_header, the timestamp of the NAL can be computed (Annex D of AVC standard) :

clockTimestamp = ((hH * 60 + mM) * 60 + sS) * time_scale +
nFrames * (num_units_in_tick * (1 + nuit_field_based_flag)) + tOffset,

in units of clock ticks of a clock with clock frequency equal to time_scale Hz, relative to some unspecified point in time for which clockTimestamp is equal to 0. Output order and DPB output timing are not affected by the value of clockTimestamp. When two or more frames with pic_struct equal to 0 are consecutive in output order and have equal values of clockTimestamp, the indication is that the frames represent the same content and that the last such frame in output order is the preferred representation.
frame_num is 0 for each IDR, we have to refresh the base time regularly.
If the NAL unit has no timing properties of its own (e.g.,parameter set and SEI NAL units), the RTP timestamp is set to the RTP timestamp of the primary coded picture of the access unit in which the NAL unit is included, according to section 7.4.1.2 of [1].
Receivers should ignore any picture timing SEI messages included in access units that have only one display timestamp. Instead, receivers should use the RTP timestamp for synchronizing the display process.
RTP senders should transmit picture timing SEI messages for pictures that are not supposed to be displayed as multiple fields.
If one access unit has more than one display timestamp carried in a picture timing SEI message, then the information in the SEI message should be treated as relative to the RTP timestamp, with the earliest event occurring at the time given by the RTP timestamp, and subsequent events later, as given by the difference in SEI message picture timing values. Let tSEI1, tSEI2, ...,tSEIn be the display timestamps carried in the SEI message of an access unit, where tSEI1 is the earliest of all such timestamps.
Let tmadjst() be a function that adjusts the SEI messages time scale to a 90-kHz time scale. Let TS be the RTP timestamp. Then, the display time for the event associated with tSEI1 is TS. The display time for the event with tSEIx, where x is [2..n] is TS + tmadjst (tSEIx - tSEI1).
Displaying coded frames as fields is needed commonly in an operation known as 3:2 pulldown, in which film content that consists of coded frames is displayed on a display
using interlaced scanning. The picture timing SEI message enables carriage of multiple timestamps for the same coded picture, and therefore the 3:2 pulldown process is perfectly controlled. The picture timing SEI message mechanism is necessary because only one timestamp per coded frame can be conveyed in the RTP timestamp.
Because H.264 allows the decoding order to be different from the display order, values of RTP timestamps may not be monotonically non-decreasing as a function of RTP sequence numbers. Furthermore, the value for interarrival jitter reported in the RTCP reports may not be a trustworthy indication of the network performance, as the calculation rules for interarrival jitter assume that the RTP timestamp of a packet is directly proportional to its transmission time.

4.3 RTCP

The RTCP implementation consists of three parts: the packet formats, the timing rules and the participants database.
There are several types of RTCP packets that must be aggregated into compound packets for transmission. These packets are sent periodically, according to a timing rules described in next sections. The interval between packets is known as the reporting interval.
All RTCP activity happens in multiples of the reporting interval. The interval varies according to the media format in use and the size of the session; typically it is on the order of 5 seconds for small sessions, but it can increase to several minutes for very large groups. Senders are given special consideration in the calculation of the reporting interval, so their source description and lip synchronization information is sent frequently; receivers report less often.
Each implementation is expected to maintain a participant database, based on the information collected from the RTCP packets it receives. This database is used to fill out the reception report packets that have to be sent periodically, but also for lip synchronization between received audio and video streams and to maintain source description information.
The RTP control protocol (RTCP) is based on the periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets. The underlying protocol must provide multiplexing of the data and control packets, for example using separate port numbers with UDP. RTCP performs four functions:

1. The primary function is to provide feedback on the quality of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport.
2. RTCP carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME. Since the SSRC identifier may change if a conflict is discovered or a program is restarted, receivers require the CNAME to keep track of each participant. Receivers may also require the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video. Inter-media synchronization also requires the NTP and RTP timestamps included in RTCP packets by data senders.
3. The first two functions require that all participants send RTCP packets, therefore the rate must be controlled in order for RTP to scale up to a large number of participants. By having each participant send its control packets to all the others, each can independently observe the number of participants. This number is used to calculate the rate at which the packets are sent.

A fourth, optional function is to convey minimal session control information, for example participant identification to be displayed in the user interface. This is most likely to be useful in "loosely controlled" sessions where participants enter and leave without membership control or parameter negotiation. RTCP serves as a convenient channel to reach all the participants, but it is not necessarily expected to support all the control communication requirements of an application. A higher-level session control protocol, which is beyond the scope of this document, may be needed.

4.3.1 The RTCP packet

Five types of RTCP packets are defined in the RTP specification: receiver report (RR), sender report (SR), source description (SDES), membership management (BYE), and application-defined (APP). They all follow a common structure, although the format-specific information changes depending on the type of packet.

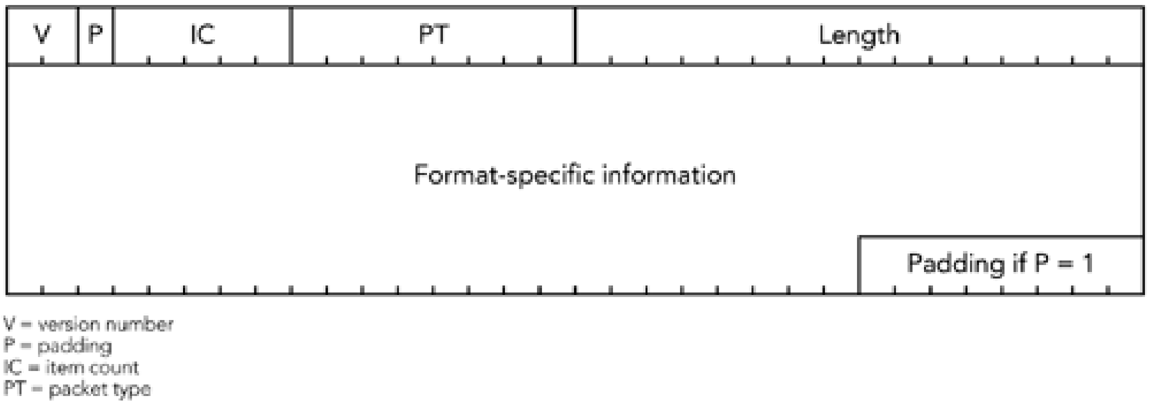


Figure 28: RTCP packet

V (Version number): (2 bits) . The version number is always 2 for the current version of RTP. There are no plans to introduce new versions, and previous versions are not in widespread use.
P (Padding): (1 bit) The padding bit indicates that the packet has been padded out beyond its natural size. If this bit is set, one or more octets of padding have been added to the end of this packet, and the last octet contains a count of the number of padding octets added., RTP Data Transfer Protocol, in the section titled Padding.
IC (Item count): (5 bits) Some packet types contain a list of items, perhaps in addition to some fixed, type-specific information. The item count field is used by these packet types to indicate the number of items included in the packet (the field has different names in different packet types depending on its use). Up to 31 items may be included in each RTCP packet, limited also by the maximum transmission unit of the network. If more than 31 items are needed, the application must generate multiple RTCP packets. An item count of zero indicates that the list of items is empty (this does not necessarily mean that the packet is empty). Packet types that don't need an item count may use this field for other purposes.
PT (Packet type): (8 bits). The packet type identifies the type of information carried in the packet. Five standard packet types are defined in the RTP specification; other types may be defined in the future (for example, to report additional statistics or to convey other source-specific information).
Length: (16 bits) The length field denotes the length of the packet contents following the common header. It is measured in units of 32-bit words because all RTCP packets are multiples of 32 bits in length, so counting octets would only allow the possibility of inconsistency. Zero is a valid length, indicating that the packet consists of only the four-octet header (the IC header field will also be zero in this case).

4.3.1.1 Sender Reports

These packets are sent by the participants of the session which have been sent data recently. The information sent in these packets informs about the media being sent and enables the receiver to synchronize multiple media streams.

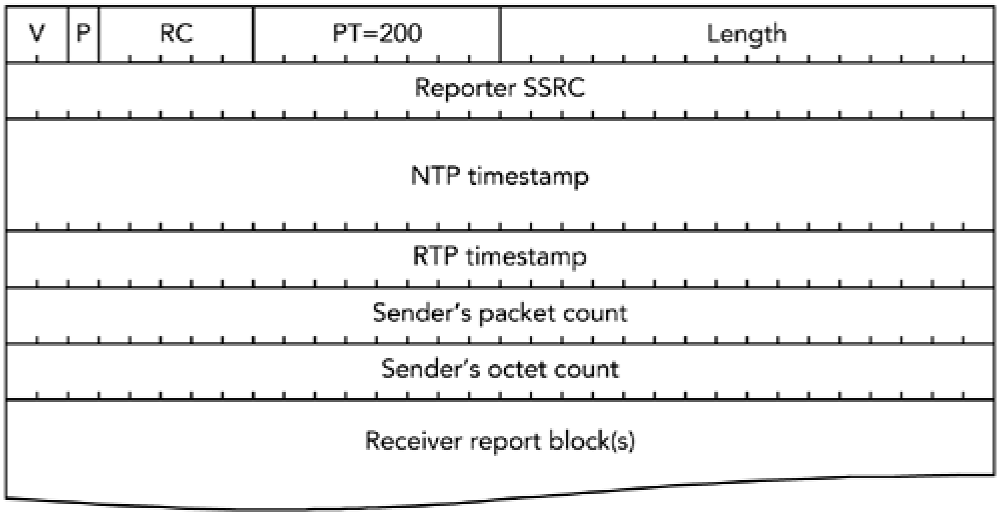


Figure 29: SR Packet format

We have to notice the 64 bit unsigned field NTP timestamp. This value indicates the time at which the actual RTCP SR packet is sent. This field is usually used to synchronize two media streams generated by different systems. The RTP timestamp is expressed in the units of the RTP media clock.
The Sender's packet count is the number of data packets that this synchronization source has generated since the beginning of the session. The sender's octet count is the number of octets contained in the payload of those data packets.
With this information an application can calculate the average payload data rate and the average packet rate over an interval without receiving the data.

4.3.1.2 Receiver Reports (RR)

The aim of this kind of packets is to report information about the quality of the transmission. This information is reported from all the participants in the session.

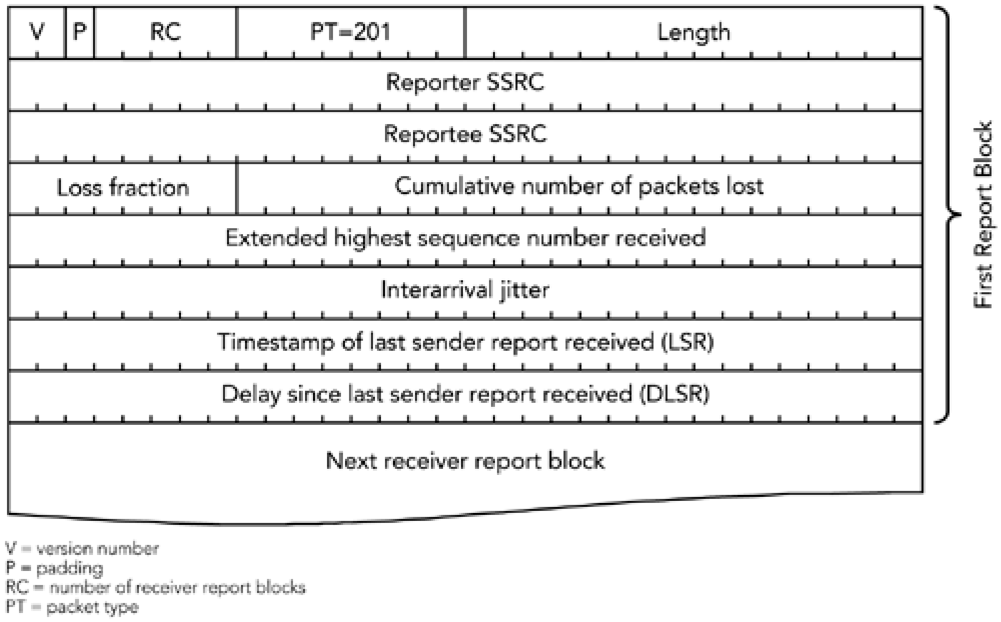

V
P
RC
PT=201
Length
Reporter SSRC
Reportee SSRC
Loss fraction
Cumulative number of packets lost
Extended highest sequence number received
Interarrival jitter
Timestamp of last sender report received (LSR)
Delay since last sender report received (DLSR)
Next receiver report block
First Report Block
V = version number
P = padding
RC = number of receiver report blocks
PT = packet type

Figure 30: RR Packet format

The information within these packets allows calculating values such as round-trip delay, etc…

4.3.1.3 Source Description (SDES)

This kind of packets provides participant identification and supplementary details.
Several types of SDES items are defined in the RTP specification. Standard item types are CNAME, NAME, EMAIL, PHONE, LOC, TOOL, NOTE, and PRIV.

4.3.1.4 Membership Control (BYE)

RTCP provides for loose membership control through RTCP BYE packets, which indicate that some participants have left the session. A BYE packet is generated when a participant leaves the session, or when it changes its SSRC.

4.3.1.5 Application-Defined (APP)

These kinds of packets are intended for application-defined extensions. So, are used for nonstandard extensions to RTCP, and for experimentation with new features.

4.3.1.6 Synchronization of content delivered over IP

RTP also provides tools for synchronization. For that purpose, an RTP time stamp is present in the RTP header; the RTP time stamps are used to determine the presentation time of the audio and video access units. The method to synchronize content transported in RTP packets is described [8].

4.4 Proposed Solution for SUIT

Besides the information provided in 3.2.6.8 about Timestamp calculation/synchronization module, in this chapter more information is detailed.
Concerning the real time encoder output, for a given input frame (YUV frame usually), the encoder will generate many NALU in the same time. It will include the base layer, spatial NALUs, SNR NALUs, each referenced with a frame number. Each NAL of the same frame will be timestamped with the same timestamp of the RTP packet.
The decoder as well will need to receive the NALU of the same timestamp in the same time. Thus they will have to be re-ordered before being given to the decoder.
Specifically to SVC, it is important to note that the base layer and the enhancement layer will be timestamped with the same time (except for temporal scalability layers) , therefore for a given frame-rate (temporal scalability) we have several layers (base layer, spatial layers, SNR layers) which have to be timestamped with the same time.
In Figure 31, it is shown how the different frames for each layer are timestamped.

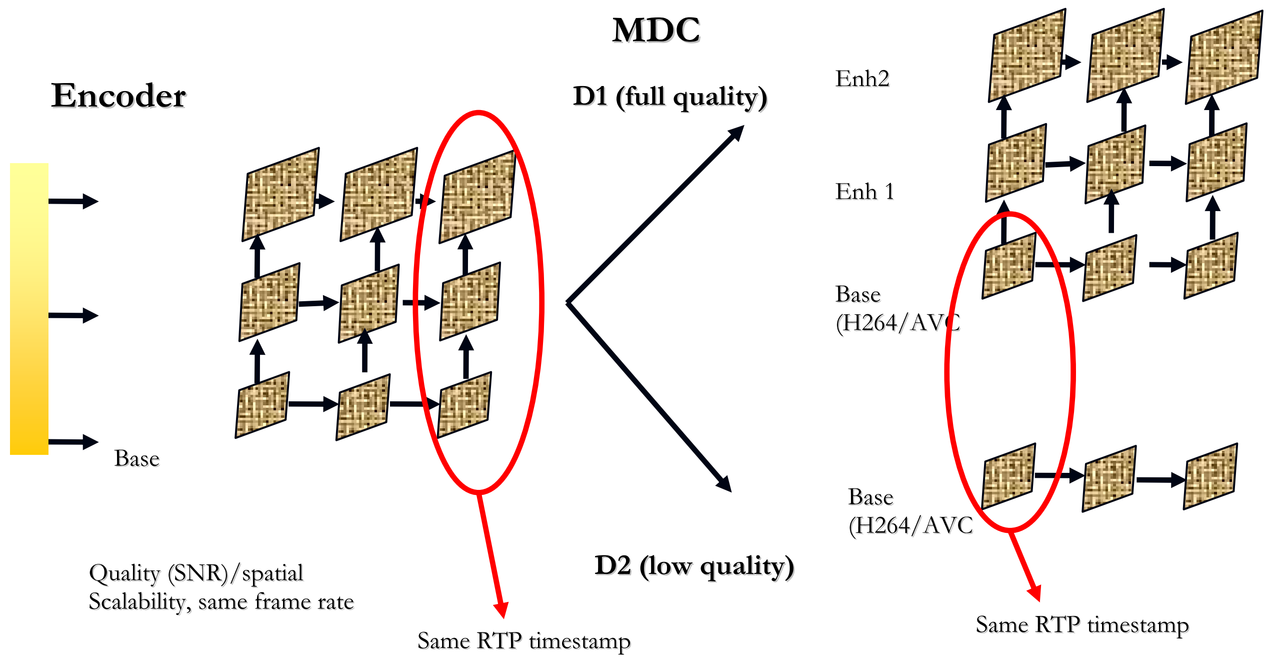


Figure 31: MDC and SVC coding

Concerning RTCP,

- RTCP Sender Reports packets will be used to synchronize descriptions. (maybe it is not necessary because we have the same clock for each description)
- RR not used in broadcast, a lot of bandwidth to receive all users RR.
- Maybe RR in unicast scenario should be used, and processed by the Intelligent Unit of the playout system.

5 Conclusions

This Deliverable is an overview of the encapsulation procedure at the different layers, from IP level down to the physical layer. The RTP Encapsulation for H264/AVC and H264/SVC is well defined for the playout system, however the RTP encapsulation at the gateway side could be a little bit different in the final demonstration of the SUIT project. The synchronization process of all the RTP streams belonging to different MDC descriptions should be improved day by day in order to achieve the best performance at the terminal side.

6 Acronyms

AIT: Application Information Table
AU: Access Unit
DON: Decoding Order Number
DONB: Decoding Order Number Base
DOND: Decoding Order Number Difference
DVB: Digital Video Broadcasting
DVB-IPI: Digital Video Broadcasting Internet Protocol Infrastructure
DVBSTP: DVB Service Detection & Selection Transport Protocol
FEC: Forward Error Correction
FU: Fragmentation Unit
GOP: Group of Pictures
GPL: General Public License
GSE: Generic Stream Encapsulation
HNED: Home Network End Device
HTTP: Hypertext Transfer Protocol
IDR: Instantaneous Decoding Refresh
IEC: International Electro technical Commission
IPTV: Internet Protocol Television
ISO: International Organization for Standardization
ITU-T: International Telecommunication Union, Telecommunication Standardization Sector
LGPL: Lesser General Public License
MANE: Media Aware Network Element
MDC: Multiple Description Coding
MHP: Multimedia Home Platform
MPE: Multiprotocol Encapsulation
MPEG: Motion Picture Experts Group
MPEG2-TS: MPEG2 transport stream
MPEG-21: Moving Picture Experts Group -21
MPTS: Multi-Program Transport Stream
MTAP: Multi-Time Aggregation Packet
MTAP16: MTAP with 16-bit timestamp offset
MTAP24: MTAP with 24-bit timestamp offset
NAL: Network Abstraction Layer
NALU: NAL Unit
PACSI: Payload Content Scalability Information NAL Unit
PPS: Picture Parameter Set
RTP: Real-Time Protocol
RTCP: Real Time Control Protocol
RTSP: Real Time Streaming Protocol
SDP: Session Description Protocol
SD&S: Service Detection & Selection
SEI: Supplemental Enhancement Information
SPS: Sequence Parameter Set
SPTS: Single Program Transport Stream
STAP: Single-Time Aggregation Packet
STAP-A: STAP type A
STAP-B: STAP type B
SVC: Scalable Video Coding
TS: Timestamp
UDP: User Datagram Protocol
VCL: Video Coding Layer
VUI: Video Usability Information

7 References

[1] *ITU-T Recommendation H.264: "Advanced video coding for generic audiovisual services" / ISO/IEC 14496-10 (2005): "Information Technology - Coding of audio-visual objects Part 10: Advanced Video Coding".*

[2] *ISO/IEC International Standard 14496-10:2003..*

[3] *ITU-T Recommendation H.241, "Extended video procedures and control signals for H.300 series terminals", 2004.*

[4] *ETSI TS 102 005: "Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols v.1.2.1"*

[5] *"H264/AVC Over IP" - Stephan Wenger , 1051-8215, IEEE transaction on circuits and systems for video technology, vol.13, No.07, July 2003.*

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

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

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

[9] *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)15–21 July, 2006*

[10] *"draft-wenger-avt-rtp-svc-03.txt", S. Wenger, Y.-K. Wang, T. Schierl, June 2006*

[11] *ETSI TS 102 034 V1.2.1, "Transport of MPEG-2 Based DVB Services over IP Based Networks", Sep 2006.*

[12] *ETSI TS 102 812 V1.2.1, "Multimedia Home Platform (MHP) Specification 1.1.1", Jun 2003.*

[13] *RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1*

[14] *ISO/IEC International Standard 13818-1: 2000.*

[15] *ISO/IEC International Standard 13818-6:1998.*

[16] *ITU-T H.610: "Full service VDSL – System architecture and customer premises equipment", Jul 2003.*

[17] *ETSI TS 101 154 V1.7.1, "Implementation guidelines for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".*

[18] IEEE 802.16,"802.16-2004 Standard for Local and Metropolitan Area Networks Part 16: Air interface to fixed and mobile broadband wireless access systems", 2004

[19] IEEE 802.16e, "Part 16: Air interface to fixed and mobile broadband wireless access systems", 2004

[20] Shin, M and Han, Y "ISP IPv6 Deployment Scenarios in Broadband Access Networks", draft-ietf-v6ops-8012-16-deployment-scenarios-00 (work in progress), May 24, 2006.

[21] Kim, S., Paik, E., Jin, J.; "IP Deployment over IEEE 802.16 Networks", draft-nam-ipv6-802-16e-01.txt (work in progress), June 24, 2006