Universal Serial Bus
Device Class Definition
for
Video Devices:
H.264 Payload

Revision 1.5
August 9, 2012

Contributors

Hans van Antwerpen	Cypress Semiconductor
David Roh	Dolby Laboratories Inc.
Choon Chng	Google Inc.
Ville-Mikko Rautio	Google Inc.
Van Duros	Immedia Semiconductor Inc.
Abdul R. Ismail	Intel Corp.
Bradley Saunders	Intel Corporation
Ygal Blum	Jungo
Yoav Nissim	Jungo
Remy Zimmermann	Logitech Inc.
Chandrashekhar Rao.	Logitech Inc.
Chris Yokum	MCCI Corporation
Stephen Cooper	Microsoft Corp.
Maribel Figuera	Microsoft Corp.
Ming-Chieh Lee	Microsoft Corp.
Mei Lu	Microsoft Corp.
Gary Sullivan	Microsoft Corp.
Chengjie Tu	Microsoft Corp.
Richard Webb	Microsoft Corp.
Andrei Jefremov	Microsoft Corp.
Tim Vlaar	Point Grey Research Inc
Mark Bohm	SMSC
John Sisto	SMSC
Will Harris	Texas Instruments
Grant Ley	Texas Instruments
Paul E. Berg	USB-IF

USB-IF AND THE AUTHORS OF THIS SPECIFICATION EXPRESSLY DISCLAIM ALL LIABILITY FOR INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. USB-IF AND THE AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE THE INTELLECTUAL PROPERTY RIGHTS OF OTHERS.

THIS SPECIFICATION IS PROVIDED "AS IS” AND WITH NO WARRANTIES, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE. ALL WARRANTIES ARE EXPRESSLY DISCLAIMED. NO WARRANTY OF MERCHANTABILITY, NO WARRANTY OF NON-INFRINGEMENT, NO WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE, AND NO WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

IN NO EVENT WILL USB-IF OR USB-IF MEMBERS BE LIABLE TO ANOTHER FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, INDIRECT, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SPECIFICATION, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Please send comments via electronic mail to <video-chair>@usb.org

Revision History

Version	Date	Description
1.5	July 25, 2012	Initial release of this H.264 Payload specification

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

This document defines the H.264 payload format for devices that are compliant with the *USB Device Class Definition for Video Devices* document.

1.2 Scope

The payload format and associated header information are fully specified in this document. This includes:

- USB Video Class stream header
- Payload-specific header

1.3 Related Documents

USB Specification Revision 3.0, November 12, 2008, www.usb.org
USB Specification Revision 2.0, April 27, 2000, www.usb.org
USB Device Class Definition for Video Devices, www.usb.org
ISO/IEC 10918-1 / ITU-T Recommendation T.81 information technology – Digital compression and coding of continuous-tone still images - Requirements and guide-lines.

1.4 Document Conventions

The following typographic conventions are used:

• *Italic*	Documents references
• **Bold**	Request fields
• UPPERCASE	Constants

The following terms are defined:

- Expected
  a keyword used to describe the behavior of the hardware or software in the design models assumed by this specification. Other hardware and software design models may also be implemented
- May
  a keyword that indicates flexibility of choice with no implied preference.
- Shall/Must
  keywords indicating a mandatory requirement. Designers are required to implement all such mandatory requirements.
- Should
  a keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase is recommended.

1.5 Normative References

1. The H.264/MPEG-4 AVC standard (referred to hereafter simply as *H.264*) is specified in the following document:
   a. *ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced video coding for generic audiovisual services*. The standard is available at http://www.itu.int/rec/T-REC-

H.264. Unless otherwise specified, this document refers to the edition approved by ITU-T in December 2011 (posted at the ITU-T web site link above).

b. The Scalable Video Coding (SVC) extensions to the H.264/MPEG-4 AVC standard (referred to hereafter simply as SVC) are specified in Annex G of the above document.

c. The Multiview Video Coding (MVC) extensions to the H.264/MPEG-4 AVC standard (referred to hereafter simply as MVC) are specified in Annex H of the above document.

2. When supported, the use of SVC and simulcast of multiple streams in the context of this specification shall additionally conform to the following specification (hereafter referred to as *UCConfig specification*):
   a. *Unified Communication Specification for H.264/MPEG-4 AVC and SVC Encoder Implementation*. This specification is available at http://technet.microsoft.com/en-us/lync/gg278176.aspx. Unless otherwise specified, this document refers to the edition of version 1.1 published in April 2011 (posted at the Microsoft web site link above).

1.6 Terminology

1.6.1 Abbreviations

For the purposes of this specification, the following abbreviations apply:

AU	Access Unit (A set of sequential NAL units that comprise a single video frame)
BP	Buffering Period
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CGS	Coarse Grained Scalability
EOF	End of Frame
EOS	End of Slice
FID	Frame Identifier
HRD	Hypothetical Reference Decoder
IDR	Instantaneous Decoding Refresh
MB	Macro block, a block of 16x16 pixels
MGS	Medium Grained Scalability
MVC	Multi-view Video Coding
NAL	Network Abstraction Layer
POC	Picture Order Count
PPS	Picture Parameter Set
PT	Picture Timing
PTS	Presentation Time Stamp
QP	Quantization Parameter
SCP	Start Code Prefix
SCR	Source Clock Reference
SEI	Supplemental Enhancement Information

SOF	Start of Frame
SPS	Sequence Parameter Set
SVC	Scalable Video Coding

1.6.2 Definitions

For the purposes of this specification, the following definitions apply:

Bitstream	A sequence of bits that forms a representation of a NAL unit stream.
NAL unit (NALU)	An H.264/MPEG-4 syntax structure containing a one-byte header and the payload byte string.
Reference frame	A frame that may be used for inter prediction in the decoding process of subsequent frame(s) in decoding order.
Simulcast streams	Multiple concurrent, independently coded bit streams from the same source, interleaved according to the *UCConfig specification*.
Frame	For purposes of this specification a *frame* is either an H.264 coded frame or a complementary pair of H.264 coded fields. Non-paired fields are not supported.
IDR Frame	For purposes of this specification, an IDR frame is defined as either a coded frame that is an IDR picture, or a coded pair of fields in which the first coded field is an IDR picture.
Random Access I Frame	For purposes of this specification, a *random access I frame* is a frame that does not use any other frames as references for inter-picture prediction and does not have any frames that follow it in *both* decoding order (i.e. bit stream order) and output order (i.e. display order) that use frames that precede it in decoding order as references for inter-picture prediction. If a random access I frame is coded as a pair of fields, the second field may use the first field as a reference for inter-picture prediction.

2 Video Class Specific Information

2.1 Compression Class

H.264 is a video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. It is the product of a partnership effort known as the Joint Video Team (JVT). The ITU-T H.264 standard and the ISO/IEC MPEG-4 AVC standard (formally, ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding) are jointly maintained so that they have identical technical content.

The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a “network-friendly” video representation addressing “conversational” (video telephony) and “non-conversational” (storage, broadcast, or streaming) applications.

The H.264 standard can be viewed as a "family of standards" representing multiple profiles. This specification the three currently supported H.264 profiles: Advanced Video Coding (AVC) including Annex B, Scalable Video Coding (SVC) as defined in Annex G, and Multiview Video Coding (MVC) as defined in Annex H.

2.2 Stream Header

Every payload transfer containing H.264 video data must start with a payload header. The format of the payload header is defined as follows.

Table 2-1 Header Format for H.264 Streams

HLE	Header Length							
BFH[0]	EOH	ERR	STI	EOS	SCR	PTS	EOF	FID
PTS	PTS[7:0]							
	PTS[15:8]							
	PTS[23:16]							
	PTS[31:24]							
SCR	SCR[7:0]							
	SCR[15:8]							
	SCR[23:16]							
	SCR[31:24]							
	SCR[39:32]							
	SCR[47:40]							
SLI	SLI[7:0]							
	SLI[15:8]							

Table 2-2 Format of the Payload Header

Offset	Field	Size	Value	Description
0	**bHeaderLength**	1	Number	Header Length field (HLE). Specifies the length of the payload header in bytes, including this field.
1	**bmHeaderInfo**	1	Bitmap	Bit Field Header (BFH[0]) field. Provides information on the sample data following

				the header, as well as the availability of optional header fields in this header. D0: **Frame ID** (FID). This bit toggles at each H.264 Access Unit (AU) start boundary and stays constant for the rest of the AU. D1: **End of Frame** (EOF). This bit indicates the end of an H.264 Access Unit and must be set to 1 only in the last payload transfer belonging to an Access Unit. D2: **Presentation Time Stamp** (PTS). This bit must be set to 1 for each payload header that includes **dwPresentationTime** data. D3: **Source Clock Reference** (SCR). This bit must be set to 1 for each payload transfer that includes **dwSourceClock** data. D4: **End of Slice** (EOS). An H.264 frame may consist of several slices. This bit, when set, indicates the end of an H.264 Slice NAL Unit and must be set to 1 only in the last payload transfer belonging to a slice.If **bmSupportedSliceModes** is not zero, this bit must be supported. . D5: **Still Image** (STI). This bit, when set, identifies the payload transfer contains data that belongs to an IDR slice. D6: **Error** (ERR). This bit is set if there was an error in the H.264 byte stream or an error in the transmission for this payload. The Stream Error Code control reflects the cause of the error. D7: **End of header** (EOH). This bit, when set, indicates the end of the BFH fields.
2	**dwPresentationTime**	4	Number	**Presentation Time Stamp** (PTS). The source clock time, in native device clock units, when the raw frame capture begins. This field must be present for every payload transfer. Payload transfers generated from a single capture time must have the same PTS. The PTS is in the same units as specified in the **dwClockFrequency** field of the Video Probe Control response.
6	**scrSourceClock**	6	Number	A two-part **Source Clock Reference**

				(SCR) value. This field must be present for each payload transfer and must be the same for all payload transfers within the same video frame. The use of SCR is redefined in this specification, putting constraints on SCR that are compatible with the UVC 1.1 specification: • SCR must be captured for SOF when the first video data of a video frame is put on the USB bus. • SCR must remain constant for all payload transfers within a single AU. D31..D0: Source Time Clock in native device clock units. D42..D32: 1KHz SOF token counter. D47..D43: Reserved. Set to zero. The least-significant 32 bits (D31..D0) contain clock values sampled from the System Time Clock (STC) at the source. The clock resolution shall be specified by the **dwClockFrequency** field of the Probe and Commit response of the device. This value shall comply with the associated stream payload specification. The times at which the STC is sampled must be correlated with the USB Bus Clock. To that end, the next most-significant 11 bits of the SCR (D42..D32) contain a 1-KHz SOF counter, representing the frame number at the time the STC was sampled. The STC is sampled when the first video data of a video frame is put on the USB bus. The SOF counter is the same size and frequency as the frame number associated with USB SOF tokens; it is required to match the current frame number. The most-significant 5 bits (D47..D43) are reserved, and must be set to zero.
12	**wLayerOrViewID**	2	Number	Stream Layer ID (SLI) SLI is required for VS_FORMAT_H.264_SIMULCAST payload and is not present for

				VS_FORMAT_H.264 payload. These two bytes contain the **wLayerOrViewID** associated with the payload data in this transfer. **wLayerOrViewID** is defined in section 3.3.3.3.

As a special case of the ERR bit in the payload header, the VC_REQUEST_ERROR_CODE_CONTROL may indicate the cause of the error as 'Buffer Overflow'. In this case, the entire current picture should be considered invalid by the both the encoder and decoder.

2.3 H.264 Payload Data

H.264 payload data consists of video encoded using the H.264 Annex B byte-stream format and is byte-oriented. The payload transfer size is variable, and the total payload transfer length (the combined payload header and payload data) for each payload transfer must not exceed the maximum payload transfer size, as specified by the **dwMaxPayloadTransferSize** field in the video Probe and Commit Control.

A raw H.264 bitstream, in the Annex B byte-stream format, is a sequence of Start Code Prefix (SCP) plus NALU pairs, possibly with zero-byte padding after the NALU data. The first SCP for a picture is 4 bytes long. Each subsequent SCP for the same picture may be either 3 or 4 bytes long. A NALU has variable size. Each NALU starts with a NALU type indicator. The compressed bits for each slice are contained in a single NALU. A video frame may be represented using multiple NALUs, because a video frame can have multiple slices.

Zero-valued bytes that appear at the end of an H.264 Annex B byte stream NALU are referred to as "trailing_zero_8bits" in the H.264 specification. For purposes of this specification, such zero-valued bytes are considered part of the NALU.

A NALU can span multiple payload transfers. If a payload transfer contains the last byte of the last Annex B byte stream NALU of a slice, the EOS flag is set in the payload header. No additional bytes may be contained in the payload transfer beyond the NAL Unit containing this last slice. A new slice must start in a different payload transfer. The slice data will be preceded by an SCP, and may be preceded by other NALUs, for example SPS/PPS and/or SEI messages.

When data from a new capture time begins being transferred, the FID is toggled between 0 and 1, and the PTS/SCR must be set in the payload header. Buffering period (BP) and picture timing (PT) supplemental enhancement information (SEI) NALUs can be used to carry additional timing information in the elementary bitstream. When present, a NALU containing a BP or PT SEI message must contain only one SEI message. A NALU containing a BP SEI message must be the first SEI NALU of the AU. A NALU containing a PT SEI message must be the first SEI NALU of the AU other than a NALU containing a BP SEI message, if present.

3 Payload-Specific Information

3.1 Descriptors

This section provides detailed information about the following Descriptors:

- H.264 Video Format Descriptor
- H.264 Frame Descriptor

3.1.1 H.264 Video Format Descriptor

The H.264 Video Format Descriptor defines the characteristics of a specific video stream. It is used for formats that carry H.264 video encoded in the H.264 Annex B byte-stream format.

A Terminal corresponding to a USB IN or OUT endpoint, and the interface it belongs to, supports one or more format definitions. To select a particular format, host software sends control requests to the corresponding interface.

The **bFormatIndex** field contains the one-based index of this format Descriptor, and is used by requests from the host to set and get the current video format.

The **bDescriptorSubtype** field uniquely identifies the video data format that should be used when communicating with this interface at the corresponding format index. For a video source function, the host software will deploy the corresponding video format decoder (if necessary) based on the format specified in this field.

The **bMaxCodecConfigDelay** indicates the maximum delay, in number of frames, the device incurs to commit a change to the encoder once the request is received.

The fields **bmSupportedSliceModes**, **bmSupportedSynchFrameTypes**, and **bmSupportRateControlModes** are used to list possible configuration settings supported by the device.

The **bDynamicResolution** field indicates if the device supports changing video resolutions while continuing to stream, and if so, with what restrictions.

The twenty **wMaxMBperSecXXX** fields provide the host with an accurate understanding of the encoder throughput for different configurations of SVC and Simulcast/Multicast features. These fields are intended to provide the host with sufficient information to predict a successful multi-stream negotiation.

An H.264 Video Format Descriptor is followed by one or more H.264 Video Frame Descriptor(s); each Video Frame Descriptor conveys information specific to a frame size supported for the format.

An H.264 Video Format Descriptor identifies the following.

Table 3-1 H.264 Payload Video Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor in bytes. The value must be 52.
1	**bDescriptorType**	1	Constant	CS_INTERFACE descriptor type.
2	**bDescriptorSubtype**	1	Constant	VS_FORMAT_H264 or VS_FORMAT_H264_SIMULCAST descriptor subtype (defined as 0x13 or 0x15). For devices that support simulcast transport, the device should create a video format descriptor with this field set to VS_FORMAT_H264_SIMULCAST. All the video frame descriptors that support simulcast shall be under this

				format. Video frame descriptors that do not support simulcast transport must be under a video format descriptor with this field set to VS_FORMAT_H264.
3	**bFormatIndex**	1	Number	Index of this format descriptor. The index must be unique from other format descriptors in the same video interface.
4	**bNumFrameDescriptors**	1	Number	Number of Frame Descriptors following that correspond to this format
5	**bDefaultFrameIndex**	1	Number	Default frame index.
6	**bMaxCodecConfigDelay**	1	Number	Maximum number of frames the encoder takes to respond to a command.
7	**bmSupportedSliceModes**	1	Bitmap	Slice mode: D0: Maximum number of MBs per slice mode D1: Target compressed size per slice mode D2: Number of slices per frame mode D3: Number of Macroblock rows per slice mode D7-D4: Reserved, set to 0 Set everything to 0 if only one slice per frame is supported.
8	**bmSupportedSyncFrameTypes**	1	Bitmap	D0: Reset D1: IDR frame with SPS and PPS headers. D2: IDR frame (with SPS and PPS headers) that is a long term reference frame. D3: Non-IDR random-access I frame (with SPS and PPS headers). D4: Generate a random-access I frame (with SPS and PPS headers) that is not an IDR frame and it is a long-term reference frame. D5: P frame that is a long term reference frame.

				D6: Gradual Decoder Refresh frames D7: Reserved, set to 0
9	**bResolutionScaling**	1	Number	Specifies the support for resolution downsizing. 0: Not supported. 1: Limited to 1.5 or 2.0 scaling in both directions, while maintaining the aspect ratio. 2: Limited to 1.0, 1.5 or 2.0 scaling in either direction. 3: Limited to resolutions reported by the associated Frame Descriptors 4: Arbitrary scaling. 5 to 255: Reserved Resolution scaling is implemented using the Video Resolution Encoding Unit, and cannot set the resolution above that specified in the currently selected frame descriptor
10	**Reserved1**	1	Number	Reserved. Set to zero.
11	**bmSupportedRateControlModes**	1	Bitmap	Supported rate-control modes. D0: Variable bit rate (VBR) with underflow allowed (H.264 low_delay_hrd_flag = 1) D1: Constant bit rate (CBR) (H.264 low_delay_hrd_flag = 0) D2: Constant QP D3: Global VBR with underflow allowed (H.264 low_delay_hrd_flag = 1) D4: VBR without underflow (H.264 low_delay_hrd_flag = 0) D5: Global VBR without underflow (H.264 low_delay_hrd_flag = 0) D7-D6: Reserved, set to 0.
12	**wMaxMBperSecOneResolutionNoScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for a single AVC stream. See Section 3.3 for details.
14	**wMaxMBperSecTwoResolutionsNoScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed

				for two AVC streams of different resolution. See Section 3.3 for details. Zero for devices that do not support simulcast.
16	**wMaxMBperSecThreeResolutionsNoScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for three AVC streams of different resolution. See Section 3.3 for details. Zero for devices that do not support simulcast.
18	**wMaxMBperSecFourResolutionsNoScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for four AVC streams of different resolution.. See Section 3.3 for details. Zero for devices that do not support simulcast.
20	**wMaxMBperSecOneResolutionTemporalScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal scalable SVC, summing up across all layers when all layers have the same resolution. See Section 3.3 for details.
22	**wMaxMBperSecTwoResolutionsTemporalScalablility**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal scalable SVC, summing up across all layers when all layers consist of two different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
24	**wMaxMBperSecThreeResolutionsTemporalScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal scalable streams, summing up across all layers when all layers consist of three different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
26	**wMaxMBperSecFourResolutionsTemporalScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal scalable streams, summing up across all layers when all layers consist of four different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.

28	**wMaxMBperSecOneResolutionTemporalQualityScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and quality scalable SVC streams, summing up across all layers when all layers have the same resolution. See Section 3.3 for details.
30	**wMaxMBperSecTwoResolutionsTemporalQualityScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and quality scalable SVC streams, summing up across all layers when all layers consist of two different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
32	**wMaxMBperSecThreeResolutionsTemporalQualityScalablity**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and quality scalable SVC streams, summing up across all layers when all layers consist of three different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
34	**wMaxMBperSecFourResolutionsTemporalQualityScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and quality scalable SVC streams, summing up across all layers when all layers consist of four different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
36	**wMaxMBperSecOneResolutionsTemporalSpatialScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and spatial scalable SVC streams, summing up across all layers when all layers have the same resolutions. See Section 3.3 for details.
38	**wMaxMBperSecTwoResolutionsTemporalSpatialScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and spatial scalable SVC streams, summing up across all layers when all layers consist of two different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.

40	**wMaxMBperSecThreeResolutionsTemporalSpatialScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and spatial scalable SVC streams, summing up across all layers when all layers consist of three different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
42	**wMaxMBperSecFourResolutionsTemporalSpatialScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for temporal and spatial scalable SVC streams, summing up across all layers when all layers consist of four different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.
44	**wMaxMBperSecOneResolutionFullScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for fully scalable streams, summing up across all layers when all layers have the same resolutions. See Section 3.3 for details.
46	**wMaxMBperSecTwoResolutionsFullScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for fully scalable streams, summing up across all layers when all layers consist of two different resolutions. Section 3.3 for details. Zero for devices that do not support simulcast.
48	**wMaxMBperSecThreeResolutionsFullScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for fully scalable streams, summing up across all layers when all layers consist of three different resolutions. Section 3.3 for details. Zero for devices that do not support simulcast.
50	**wMaxMBperSecFourResolutionsFullScalability**	2	Number	Maximum macroblock processing rate, in units of 1000 MB/s, allowed for fully scalable streams, summing up across all layers when all layers consist of four different resolutions. See Section 3.3 for details. Zero for devices that do not support simulcast.

3.1.2 H.264 Video Frame Descriptors

H.264 Video Frame Descriptors (or simply Frame Descriptors) are used to describe the decoded video frame dimensions, H.264 profile and level, and other frame-specific characteristics supported by a particular stream. One or more Frame Descriptors follow the H.264 Video Format Descriptor they correspond to. The Frame Descriptor is also used to determine the range of frame intervals that are supported for the specified frame size and H.264 profile.
Each Video Frame Descriptor describes a unique video resolution/H.264 profile combination. Codec usages, codec capabilities, video frame rates, and so forth are then enumerated for that combination.

The H.264 Video Frame Descriptor is used only for video formats for which the H.264 Video Format Descriptor applies (see section 3.1.1, "H.264 Video Format Descriptor").
The **bFrameIndex** field contains the one-based index of this Frame Descriptor, and is used by requests from the host to set and get the current frame index for the format in use. The index value ranges from one to **bNumFrameDescriptors**, and must be unique within a Video Format.
The range of frame intervals supported is a discrete set of values where the **dwFrameInterval(x)** fields indicate the range of frame intervals (and therefore frame rates) supported at this frame size. The frame interval is the average display time of a single decoded video frame in 100ns Units.
Each Video Frame Descriptor must support at least one **dwFrameInterval**.

Table 3-2 H.264 Payload Video Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor in bytes. The value must be 44 + (bNumFrameIntervals * 4).
1	**bDescriptorType**	1	Constant	CS_INTERFACE descriptor type.
2	**bDescriptorSubtype**	1	Constant	VS_FRAME_H264 descriptor subtype (defined as 0x14)
3	**bFrameIndex**	1	Number	Index of this Frame Descriptor
4	**wWidth**	2	Number	The width, in pixels, of pictures output from the decoding process. Must be a multiple of 2. Does not need to be an integer multiple of 16, and can be specified using a frame cropping rectangle in the active SPS.
6	**wHeight**	2	Number	The height, in pixels, of pictures output from the decoding process. Must be a multiple of 2. When field coding or frame/field adaptive coding is used, shall be a multiple of 4. Does not need to be an integer multiple of 16, and can be specified using a frame

				cropping rectangle in the active SPS.
8	**wSARwidth**	2	Number	Sample aspect ratio width (as defined in H.264 Annex E); shall be relatively prime with respect to **bSARheight**.
10	**wSARheight**	2	Number	Sample aspect ratio height (as defined in H.264 Annex E); shall be relatively prime with respect to **bSARwidth**.
12	**wProfile**	2	Number	The first two bytes of the sequence parameter set, specified by profile_idc and constraint flags in the H.264 specification, to indicate the profile and applicable constraints to be used. For example: 0x4240: Constrained Baseline Profile 0x4200: Baseline Profile 0x4D00: Main Profile 0x640C: Constrained High Profile 0x6400: High Profile 0x5304: Scalable Constrained Baseline Profile 0x5300: Scalable Baseline Profile 0x5604: Scalable Constrained High Profile 0x5600: Scalable High Profile 0x7600: Multiview High Profile 0x8000: Stereo High Profile
14	**bLevelIDC**	1	Number	The level, as specified by the level_idc flag (9, 10, 11, 12, 13, 20, 21, 22, 30, 31, 32, 40, 41, 42, etc). For example: 0x1F: Level 3.1. 0x28: Level 4.0. Note that this should ordinarily indicate the minimum level that supports the resolution and maximum bit rate for this frame descriptor.
15	**wConstrainedToolset**	2	Number	Reserved, set to zero
17	**bmSupportedUsages**	4	Bitmap	D0: Real-time/UCConfig mode 0. D1: Real-time/UCConfig mode 1.

				D2: Real-time/UCConfig mode 2Q. D3: Real-time/UCConfig mode 2S. D4: Real-time/UCConfig mode 3. D7-D5: Reserved; set to 0. D15-D8: Broadcast modes. D16: File Storage mode with I and P slices (e.g. IPPP). Must be set to 1. D17: File Storage mode with I, P, and B slices (e.g. IB…BP). D18: File storage all-I-frame mode. D23-D19: Reserved; set to 0. D24: MVC Stereo High Mode. D25: MVC Multiview Mode. D31-D26: Reserved; set to 0. Devices must support bmSupportedUsages(D16) “File Storage I, P, P”
21	**bmCapabilities**	2	Bitmap	D0: CAVLC only. D1: CABAC only. D2: Constant frame rate. D3: Separate QP for luma/chroma. D4: Separate QP for Cb/Cr. D5: No picture reordering. D6: Long Term Reference frame D15-D7: Reserved; set to 0. Note when D4 is 1, then D3 must be 1.
23	**bmSVCCapabilities**	4	Bitmap	D2-D0: Maximum number of temporal layers minus 1. D3: Rewrite support. D6-D4: Maximum number of CGS layers minus 1. D9-D7: Maximum number of MGS sublayers. D10: Additional SNR scalability support in spatial enhancement layers. D13-D11: Maximum number of

				spatial layers minus 1. D31-D14: Reserved. Set to zero. See Section 3.3.2 for details.
27	**bmMVCCapabilities**	4	Bitmap	D2-D0: Maximum number of temporal layers minus 1. D10-D3: Maximum number of view components minus 1. D31-D11: Reserved. Set to zero. See Section 3.4 for details.
31	**dwMinBitRate**	4	Number	Specifies the minimum bit rate, at maximum compression and longest frame interval, in units of bps, at which the data can be transmitted.
35	**dwMaxBitRate**	4	Number	Specifies the maximum bit rate, at minimum compression and shortest frame interval, in units of bps, at which the data can be transmitted.
39	**dwDefaultFrameInterval**	4	Number	Specifies the frame interval the device indicates for use as a default, in 100-ns units
43	**bNumFrameIntervals**	1	Number	Specifies the number of frame intervals supported
44	**dwFrameInterval(1)**	4	Number	Shortest frame interval supported (at the highest frame rate), in 100-ns units.
	…			
44 + (**bNumFrameIntervals** * 4) – 4	**dwFrameInterval (bNumFrameIntervals)**	4	Number	Longest frame interval supported (at lowest frame rate), in 100-ns units.

3.1.3 Encoding Unit Controls

The following Encoding Unit controls have specialized behavior when applied to an H.264 payload.

3.1.3.1 Select Layer Control

For H.264, multi-layer equates with SVC (Annex G), multi-view equates with MVC (Annex H) and for single stream AVC **wLayerOrViewID** = 0.

For H.264, The exact matching between each SVC layer and these values is specified in the UCConfig specification and the H.264 standard. Section 3.3.1 describes how stream_id is determined for AVC and SVC streams. Section 3.4.1 describes how stream_id is determined for MVC streams.

Wildcard masks for SVC streams are defined in section 3.3.3.3.1, and wildcard masks for MVC streams are defined in section 3.4.3.1.

3.1.3.2 Profile Toolset Control

The Profile Toolset control is directly mapped to H.264 profiles. The values of **bmSettings** for H.264 are shown below as well.

Table 3-3 Updates to the Profile Toolset Control for H.264 Payloads

Field	Value
wProfile	wProfile represents the first two bytes of the sequence parameter set, specified by profile_idc and constraint flags in H.264 spec to indicate the profile and applicable constraints to be used. 0x4240: Constrained Baseline Profile 0x4200: Baseline Profile 0x4D00: Main Profile 0x640C: Constrained High Profile 0x6400: High Profile 0x5304: Scalable Constrained Baseline Profile 0x5300: Scalable Baseline Profile 0x5604: Scalable Constrained High Profile 0x5600: Scalable High Profile 0x7600: Multiview High Profile 0x8000: Stereo High Profile
bmSettings	D0: CAVLC only. D1: CABAC only. Bits D1-D0 have the following meaning: 00: Let the device choose CAVLC/CABAC. 01: CAVLC only. 10: CABAC only. 11: Reserved. D2: Constant frame rate D3: Separate QP for luma/chroma D4: Separate QP for Cb/Cr D5: No picture reordering D15-D8: Reserved; set to 0 Note that if D4 is 1 then D3 must be 1.

3.1.3.3 Video Resolution Control

If the value of **bResolutionScaling** in the negotiated video format descriptor is not 0, this control can be used to change the decoded video width and height of one or more layers before or during streaming. The **wWidth** and **wHeight** fields must each be a multiple of two. If the value of **bResolutionScaling** imposes limits that are stricter than those imposed by GET_RES() those stricter limits must be respected, otherwise the limits imposed by GET_RES() must be respected.

3.1.3.4 Rate Control Mode Control

Note that the conformance of the leaky bucket model to the H.264 HRD model is specified in terms of the decoder perspective, although the encoder perspective tends to be easier to describe when expressing the intent.

3.1.3.4.1 Variable Bit Rate (VBR)

When bits are flowing, the H.264 HRD operates at the specified peak bit rate

R_P = **dwPeakBitRate** x 64 bps

rather than the "average" bit rate **dwAverageBitRate** bps.

3.1.3.4.2 Constant Bit Rate (CBR)

The application must specify the rate-control parameters in the Average Bitrate, CPB Size, and Peak Bitrate EUs in an HRD-conformant manner with respect to the profile and level combination

3.1.3.4.3 Low Delay and Non-Low Delay Modes

Within the VBR and Global VBR modes of operation, there are two variants that correspond to values of the H.264 low_delay_hrd_flag syntax element. These variants concern the timely availability of the coded bits for decoding purposes. If the decoding time of a picture arrives but not all of the bits that represent that picture have yet drained out of the encoders leaky bucket model (and therefore those bits are not yet available in the decoder's input buffer when the decoding time of the picture arrives), the leaky bucket model is said to "underflow"[1].

- When low_delay_hrd_flag is equal to 1, it is allowed for the leaky bucket model to underflow. Operating in this manner can help reduce the average end-to-end delay through the system (although it may sometimes cause the decoder to not exactly reproduce the correct timing of the pictures for its output/display purposes).
- When operating with the H.264 low_delay_hrd_flag equal to 0, the HRD shall not underflow when operating at the peak bit rate – i.e., although it is allowed for the leaky bucket to sometimes "run dry", it is not allowed for the decoding time of a picture to arrive before all of the bits for that picture have yet departed from the encoder's leaky bucket (which corresponds to arrival into the decoder's corresponding input buffer).

The low delay and non-low delay variants apply to the VBR and Global VBR modes. This is reflected in the following four rate control modes specified in the Rate Control Mode encoding unit in the Video Class specification:

1: Variable Bit Rate low delay (VBR)
4: Global VBR low delay (GVBR)
5: Variable bit rate non-low delay (VBRN)

[1] Note that this condition is described from the perspective of a hypothetical reference decoder (HRD), and the time to move the bits from the encoder's leaky bucket to the decoder's input buffer is not accounted for. Also, the means by which the decoding time for a picture is determined is not covered here, although it may be indicated by picture timing SEI messages.

6: Global VBR non-low delay (GVBRN)

If no VUI information (Annex E) is included in the SPS then fixed_frame_rate_flag is assumed to be 0 and thereby low_delay_hrd_flag is assumed to be 1. These settings correspond to a low delay, variable frame rate stream.

3.1.3.5 Quantization Parameter Control

Table 3-4 Updates to the Quantization Parameter Control for H.264 Payloads

Field	Value
wQpPrime_I, wQpPrime_P, and wQpPrime_B	Up to three values can be passed to configure the picture quantization parameters. D7-D0: QP'$_{Y}$ D11-D8: chroma_qp_index_offset as a signed 4-bit number in two’s complement representation. The value shall be in the range −8 to +7, inclusive. D15-D12: second_chroma_qp_index_offset as a signed 4-bit number in two’s complement representation. The value shall be in the range −8 to +7.

Note: The H.264 standard specifies that QP'Y = QPY + 6 * bit_depth_luma_minus8, where bit_depth_luma_minus8 corresponds to bBitDepthLuma – 8, negotiated in Probe/Commit.

3.1.3.5.1 Quantization Weighting Matrices

This specification does not support control of quantization weighting matrices. For profiles that support the use of quantization weighting matrices, when a QP value control is specified, the quantization weighting matrix entries must either be flat matrices with all entries equal to 16, or must have rate-distortion behavior that is approximately similar to the use of flat matrices with entries equal to 16.

3.1.3.6 Syncronization and Long-Term Reference Frame

When **bSyncFrame** is 1 or 2 the resulting reference frame must be preceded by new SPS and PPS NAL units. When **bSyncFrame** is 3 or 4 the resulting reference frame must be preceded by the active SPS and PPS NAL units.

3.1.3.7 Long-Term Buffer

Note that the encoder is responsible for signaling appropriate decoder picture buffer parameters in the SPS. The encoder shall make sure that reference buffer count stays within the limits given the assigned level_idc. The encoder shall generate an IDR if the total number of buffers changes.

3.1.3.8 Long-Term Buffer Size

The index value referred to in this control is the long_term_frame_idx from the H.264 specification.

3.1.3.9 Long-Term Reference Picture

The index value referred to by **bPutAtPositionInLTRBuffer** is the long_term_frame_idx from the H.264 specification.

3.1.3.10 Priority Control

For H.264 encoding, the Priority control is used to set SVC syntax element prority_id for all of the NALUs in a specific layer. If the selected layer supports prefix NALUs with nal_unit_type equal to 14 or NALUs with nal_unit_type equal to 20, then **bPriority** shall be used to set the priority_id for these NALUs.
On a GET_MIN request, the device shall return 0. On a GET_MAX request, the device shall return 63.

When **bUsage** is between 1 and 5 (Real-time/UCConfig modes), the device shall return defaults according to the rules specified in the *UCConfig specification.*When **bUsage** is outside the 1 to 5 range, GET_DEF shall return **bPriorityID** = 0.

3.1.4 Probe and Commit

The following fields have special behavior negotiating an H.264 stream.

Table 3-5 Updates to the Probe and Commit Control for H.264 Payloads

Field	Value
bUsage	For real time modes 1-5, use the following values: 1: Real-time/UCConfig mode 0 2: Real-time/UCConfig mode 1 3: Real-time/UCConfig mode 2Q 4: Real-time/UCConfig mode 2S 5: Real-time/UCConfig mode 3 For file storage modes 17-19, use the following values 17: File Storage mode with I and P slices (e.g., IPPP) 18: File Storage mode with I, P, and B slices (e.g., IB…BP) 19: File storage with all-I-frame mode For multiview modes 25 – 26, use the following: 25: MVC Stereo High mode 26: MVC Multivew mode The Real-time/UCConfig mode selected must be the highest of all the UCConfig modes that will be used among all simulcast streams. The specific configuration for each stream shall be specified in the **bmLayoutPerStream** field. The MVC mode selected must be the highest of all the MVC modes that will be used among all simulcast streams. The specific configuration for each stream shall be specified in the **bmLayoutPerStream** field.

bmSettings	D0: CAVLC only. D1: CABAC only. Bits D1-D0 have the following meaning: 00: Let the device choose CAVLC/CABAC. 01: CAVLC only. 10: CABAC only. 11: Reserved. D2: Constant frame rate. D3: Separate QP for luma/chroma. D4: Separate QP for Cb/Cr. D5: No picture reordering. D7-D6: Reserved; set to 0.
bMaxNumberOfRefFramesPlus1	When non-zero, the max_num_ref_frames syntax element in H.264 shall be less than or equal to this value minus 1.
bmLayoutPerStream	This field is used to describe the layering structure for multiple streams when leveraging simulcast transport. When **bUsage** indicates an H.264 SVC profile: D15-0: layering structure for simulcast stream with stream_id 0. D31-16: layering structure for simulcast stream with stream_id 1. D47-32: layering structure for simulcast stream with stream_id 2. D63-48: layering structure for simulcast stream with stream_id 3. It is recommended to associate streams with lower resolution/lower bit rate with smaller stream_id. When **bmUsage** indicates an H.264 MVC profile: D15-0: MVC view structure for simulcast stream with stream_id 0. D31-16: MVC view structure for simulcast stream with stream_id 1. D47-32: MVC view structure for simulcast stream with stream_id 2.

3.2 Video Samples

3.3 SVC and Simulcast Support

This section provides technical background, detailed descriptions, and examples to illustrate how to support SVC and/or simulcast in this specification. Developers may skip this section if the encoder does not support the generation of SVC bit streams. In that case, both **bmSVCCapabilities** and **bmLayoutPerStream** shall be set to 0.

3.3.1 SVC Overview

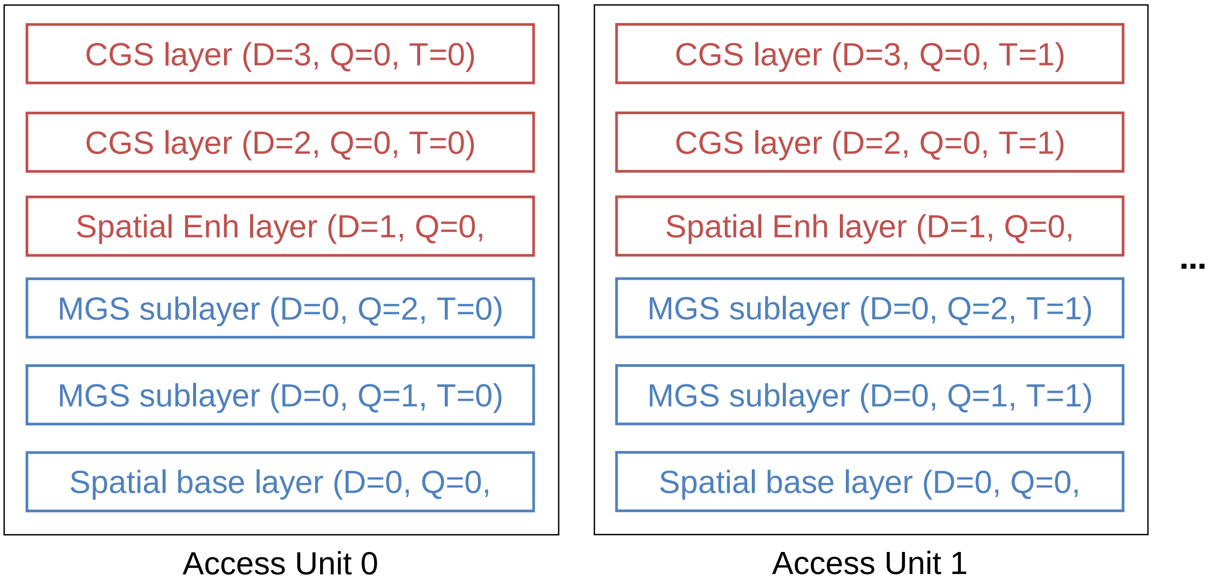
Scalable Video Coding (SVC) is primarily specified in Annex G of the H.264/MPEG-4 Advanced Video Coding (AVC) standard. Within an access unit (AU), there is one "base layer" that is formatted as an H.264/AVC coded picture, and one or more additional scalable layer representations, each of which represents an additional "enhancement layer" of an SVC-encoded bitstream for the same instant in time. SVC supports three main types of classes of scalability: temporal, quality (or SNR), and spatial scalability. Quality scalability can be further classified into Coarse Grained Scalability (CGS) and Medium Grained Scalability (MGS). An SVC bitstream may contain arbitrary combinations of these three classes of scalability. To simplify the design, this specification only considers the most commonly used layering structures as defined in the UCConfig Specification, summarized as follows:

- Temporal scalability, if it is used, is applied first in layering a SVC bitstream. A temporal layer is identified by the syntax element temporal_id for an H.264 NALU. The value of temporal_id must start from 0 and increase continuously.
- Quality scalability, if it is used, is applied next in layering a SVC bitstream. A quality layer is identified by the syntax element dependency_id in CGS mode and quality_id in MGS mode for an H.264 NALU. The values of quality_id and dependency_id must start from 0 and increase continuously. When MGS is used, an MGS layer is split into multiple sublayers by means of transformed coefficient partitioning. CGS is effectively a special case of spatial scalability when two successive spatial layers have identical spatial resolutions.
- Spatial scalability, if it is used, is applied next in layering a SVC bitstream. A spatial layer is identified by the syntax element dependency_id in an H.264 NALU.
    - Additional quality scalable layers may be applied in a spatial enhancement layer.

With these constraints, for a particular layering structure the values of temporal_id, dependency_id, and quality_id associated with a layer can be determined without ambiguity and used as a unique identifier for that layer within its associated stream.

The following figure shows an example SVC layering layout supported in this specification. The bitstream contains two temporal layers, two MGS sub-layers, two spatial layers, and two additional CGS layers in the first spatial enhancement layer. The values of temporal_id, dependency_id, and quality_id of each layer are also shown in the figure:

Figure 9-1 UCConfig Compliant SVC Example


CGS layer (D=3, Q=0, T=0)
CGS layer (D=2, Q=0, T=0)
Spatial Enh layer (D=1, Q=0,
MGS sublayer (D=0, Q=2, T=0)
MGS sublayer (D=0, Q=1, T=0)
Spatial base layer (D=0, Q=0,
Access Unit 0
CGS layer (D=3, Q=0, T=1)
CGS layer (D=2, Q=0, T=1)
Spatial Enh layer (D=1, Q=0,
MGS sublayer (D=0, Q=2, T=1)
MGS sublayer (D=0, Q=1, T=1)
Spatial base layer (D=0, Q=0,
Access Unit 1
...

Section 3.3.2 defines the format of **bmSVCCapabilities** field in the Video Frame Descriptor. Section 3.3.3.1 defines the format of **bmLayoutPerStream** field in the Probe and Commit Control. The **bmLayoutPerStream** field describes the SVC layering structure associated with each simulcast stream, each of which is limited by the maximum SVC capabilities defined by **bmSVCCapabilities**.

3.3.2 SVC Capability Advertisement

The encoder notifies the SVC capabilities using **bmSVCCapabilities** in the Video Frame Descriptor. The **bmSVCCapabilities** field describes the maximum SVC capability supported by the hardware. The following table shows how the field is formatted:

Table 9-1 Bit Values of bmSVCCapabilities Field

Bitfields	Name
[2-0]	MaxNumOfTemporalLayersMinus1
3	RewriteSupport
[6-4]	MaxNumOfCGSLayersMinus1
[9-7]	MaxNumOfMGSSublayers
10	AdditionalSNRScalabilitySupport
[13-11]	MaxNumOfSpatialLayersMinus1
[31-14]	Reserved, set to 0

MaxNumOfTemporalLayersMinus1: Indicates the maximum number of temporal layers in a bitstream. A non-zero value indicates that the encoder supports the creation of temporal scalable bitstreams. This specification only allows and supports values between 0 and 3.

RewriteSupport: Indicates whether the encoder supports the creation of quality scalable bitstreams that can be converted into bitstreams that conform to one of the non-scalable H.264/AVC profiles, by using a low-complexity rewriting process.

MaxNumOfCGSLayersMinus1: Indicates the maximum number of CGS quality layers in a bitsteam. A non-zero value indicates that the encoder supports the creation of CGS quality scalable bitstreams. This specification only allows and supports values between 0 and 2, which corresponds to a maximum of three sublayers.

MaxNumOfMGSSublayers: Indicates the maximum number of MGS sub-layers allowed in an MGS layer in a bitsteam. This specification requires that if supported, only two MGS layers (one base layer and one MGS enhancement layer with multiple sublayers) are present in a spatial layer. A non-zero value indicates that the encoder supports the creation of MGS quality scalable bitstreams. When supported, this specification only allows a value between 2 and 4, which corresponds to a minimum of two and a maximum of four sublayers. Key frame generation shall be supported in MGS.

AdditionalSNRScalabilitySupport: Indicates whether additional quality (or SNR) layers are allowed to be present in a spatial enhancement layer. When this field is 1, additional SNR scalability may be introduced based on the capability of quality scalability specified for the base spatial layer. That is, the introduction of quality layers in a spatial enhancement layer is constrained by RewriteSupport, MaxNumOfCGSLayersMinus1, KeyFrameSupport, and MaxNumOfMGSSublayers.

MaxNumOfSpatialLayersMinus1: Indicates the maximum number of spatial layers in a bitsteam. A non-zero field indicates that the encoder supports the creation of spatial scalable bitstreams. This specification only allows and supports values between 0 and 2, which corresponds to a maximum of three spatial layers.

For encoders that only support the generation of AVC single-layer streams, **bmSVCCapabilities** shall be set to 0.

3.3.3 SVC Stream/Layer Configuration

3.3.3.1 Initialization

The encoder indicates the number of simulcast streams and the layering structure associated with each stream using the **bmLayoutPerStream** field in the Probe/Commit Control. These simulcast streams may be AVC single-layer streams, SVC multi-layer streams, or a combination of both. In this specification, at most four simulcast streams are allowed. They are indexed with stream_id 0, 1, 2 and 3. The following tables define the details of the **bmLayoutPerStream** field.

Table 9-2 Byte Layout of bmLayoutPerStream Field for SVC

SVC_STR3[63:48]	SVC_STR2[47:32]	SVC_STR1[31:16]	SVC_STR0[15:0]

bmLayoutPerStream consists of four 16-bit subfields. Each subfield describes the layering structure of one of the SVC streams. To identify an individual SVC stream, this specification uses the terminology of SVC_STR*x* (*x* = 0, 1, 2 and 3) where x is the stream_id of the stream.The subfields are interpreted as shown in the table below.

Table 3-6: Bit Layout of bmLayoutPerStream subfields for SVC

Bitfields	Name
[2-0]	NumOfTemporalLayers
3	SNRModeBase
4	SNRModeAttributeBase

[6-5]	NumberOfSNRLayersMinus1Base
7	SNRMode1st
8	SNRModeAttribute1st
[11-9]	NumberOfSNRLayers1st
12	SNRMode2nd
13	SNRAttribute2nd
[15-14]	NumberOfSNRLayers2nd

NumOfTemporalLayers:

Indicates the number of temporal layers in the bitstream. This value is effectively the maximum value of the syntax element temporal_id in the H.264 SVC specification plus one. For example, if this field is 3, three temporal layers are present in the bitstream, corresponding to temporal_id 0, 1, and 2. The value 0 indicates that the corresponding stream is not present. The value of this field must not exceed the maximum number of temporal layers specified in **bmSVCCapabilities**.

SNRModeBase:

Indicates whether CGS or MGS is used to generate quality layers in the base spatial layer. The value 0 means CGS is used, and 1 means MGS is used. When CGS is used, **SNRModeAttributeBase** indicates whether the rewriting process is enabled. The value 0 means rewriting is disabled, and 1 means rewriting is enabled. When MGS is used, **SNRModeAttributeBase** indicates whether key frame generation is enabled. The value 0 means key frame generation is disabled, and 1 means it is enabled. The use of CGS or MGS mode must follow what was advised in **bmSVCCapabilities**.

NumberOfSNRLayersMinus1Base:

Indicates the number of CGS quality layers or MGS sublayers, depending on the value of SNRModeBase, in the base spatial layer. When CGS is used (SNRModeBase is 0), this field effectively corresponds to the values of the syntax element dependency_id in the H.264 SVC specification. For example, if this field is 2, three CGS layers are present in the base spatial layer in the bitstream, corresponding to dependency_id 0, 1, and 2. When MGS is used (SNRMode is 1), this field effectively corresponds to the values of the syntax element quality_id in the H.264 SVC specification. For example, if this field is 2, three MGS sublayers are present in the base spatial layer in the bitstream, corresponding to quality_id 1, 2, and 3. The value 0 implies that no SNR scalability is introduced in the base spatial layer. The value of this field must not exceed the maximum number of quality layers specified in **bmSVCCapabilities**.

SNRMode1st:

Indicates whether CGS or MGS is used to generate additional quality layers in the first spatial enhancement layer (if present). The value 0 means CGS is used, and 1 means MGS is used. When CGS is used, SNRModeAttribute1st indicates whether the rewriting process is enabled. The value 0 means rewriting is disabled, and 1 means rewriting is enabled. When MGS is used, SNRModeAttribute1st indicates whether key frame generation is enabled. The value 0 means key frame generation is disabled, and 1 means it is enabled. The use of CGS or MGS mode must be constrained by **bmSVCCapabilities**.

NumberOfSNRLayers1st:

Indicates whether spatial scalability is introduced in the bitstream and how additional SNR scalability is used. If the value is 0, spatial scalability is not introduced in the bitstream. If the value is 1, spatial scalability is used but no additional SNR scalability is introduced in the first spatial enhancement layer. If the value is 2 or larger, spatial scalability is used and additional SNR scalability is introduced in the first spatial enhancement layer. In that case, this field indicates the number of CGS quality layers or MGS sublayers, depending on the value of SNRMode1st, used in the first spatial enhancement layer. When this field is non-zero, the maximum number of spatial layers advised in bmSVCCapabilities must be at least 2. When this field is larger than 1, the use of additional SNR scalability must not exceed the capability of quality scalability specified in **bmSVCCapabilities**.

When CGS is used (SNRMode1st is 0), the value of this field effectively corresponds to the values of the syntax element dependency_id in the H.264 SVC specification. For example, if this field is 3, three CGS layers are present in the first spatial enhancement layer in the bitstream, corresponding to dependency_id *K*+1, *K*+2, and *K*+3, where *K* is 0 if SNRModeBase is 1 and *K* is NumberOfSNRLayersMinus1Base if SNRModeBase is 0.

When MGS is used (SNRMode1st is 1), this value effectively corresponds to the values of the syntax element quality_id in the H.264 SVC specification. For example, if this field is 3, three MGS sublayers are present in the first spatial enhancement layer in the bitstream, corresponding to quality_id 1, 2, and 3. The value of this field must not exceed the maximum number of quality layers specified in **bmSVCCapabilities**.

SNRMode2nd:

Indicates whether CGS or MGS is used to generate additional quality layers in the second spatial enhancement layer (if present). The value 0 means CGS is used, and 1 means MGS is used. When CGS is used, SNRModeAttribute2nd indicates whether the rewriting process is enabled. The value 0 means rewriting is not used, and 1 means rewriting is used. When MGS is used, SNRModeAttribute2nd indicates whether key frame generation is enabled. The value 0 means key frame generation is disabled, and 1 means key frame generation is enabled. The use of CGS or MGS mode must be constrained by **bmSVCCapabilities**.

NumberOfSNRLayers2nd:

Indicates whether the second spatial enhancement layer is introduced in the bitstream and if so, whether additional SNR scalability is used. If the value is 0, the second spatial enhancement layer is not present in the bitstream. If the value is 1, the second spatial enhancement layer exists but no additional SNR scalability is introduced. If the value is 2 or larger, additional SNR scalability is introduced in the second spatial enhancement layer. In that case, the value of this field indicates the number of CGS quality layers or MGS sublayers, depending on the value of SNRMode2nd, used in the second spatial enhancement layer. When this field is non-zero, NumberOfSNRLayers1st must also be non-zero, and the maximum number of spatial layers advised in bmSVCCapabilities must be at least 3. When this field is larger than 1, the use of additional SNR scalability must not exceed the capability of quality scalability specified in **bmSVCCapabilities**.

When CGS is used (SNRMode2nd is 0), the value of this field effectively corresponds to the values of the syntax element dependency_id in the H.264 SVC specification. For example, if this field is 3, three CGS layers are present in the second spatial enhancement layer in the bitstream, corresponding to dependency_id *K*+1, *K*+2, and *K*+3, where *K* equals 1 if both SNRModeBase and SNRMode1st are 1; *K* equals (NumberOfSNRLayersMinus1Base + 1) if

SNRModeBase is 0 but SNRMode1st is 1, and *K* equals NumberOfSNRLayers1st if SNRModeBase is 1 but SNRMode1st is 0.

When MGS is used (SNRMode2nd is 1), this value effectively corresponds to the values of the syntax element quality_id in the H.264 SVC specification. For example, if this field is 3, three MGS sublayers are present in the second spatial enhancement layer in the bitstream, corresponding to quality_id 1, 2, and 3. The value of this field must not exceed the maximum number of quality layers specified in **bmSVCCapabilities**.

For each 16-bit subfield, a non-zero value indicates the presence of the corresponding simulcast stream. For encoders that support only one AVC single-layer stream, at most one of the subfields may be set, and it must be set to 1. For encoders that supports SVC but do not support simulcast, at most one of the subfields may be set, and it may be set to a value greater than or equal to 1.

The host should configure simulcast streams starting from the subfield corresponding to the lowest stream_id. However, the device must be able to handle configurations that result in a gap in stream_id.

Configuration of SVC and simulcast streams involve a single Probe/Commit Control followed by multiple encoding unit (EU) controls, as specified in the rest of this section.

3.3.3.2 Negotiating Total Throughput of the System

In the Probe/Commit Control, the encoder specifies the number of active simulcast streams and the layering layout of each simulcast stream in **bmLayoutPerStream**. The following fields in the Video Frame Descriptor are employed as an indicator for available devices resources given the degree of SVC scalability and the number of resolution rescalings:

For non-simulcast streams (VS_FORMAT_H264), only values that indicate “OneResolution” apply and all others must be zero. For AVC streams, only values that indicate “NoScalability” apply and all others must be zero.

- **dwMaxMBperSecOneResolutionNoScalability** represents the maximum macroblock processing rate when only AVC non-scalable streams are used and when only one resolution is requested (no spatial rescaling).
- **dwMaxMBperSecTwoResolutionsNoScalability**, **dwMaxMBperSecThreeResolutionsNoScalability**, and **dwMaxMBperSecFourResolutionsNoScalability** represent the maximum macroblock processing rate when only AVC non-scalable streams are used and when two (one spatial rescaling), three (two spatial rescaling) and four resolutions (three spatial rescaling) across all layers in all streams are requested.
- **dwMaxMBperSecOneResolutionTemporalScalability**, **dwMaxMBperSecTworResolutionsTemporalScalability**, **dwMaxMBperSecThreeResolutionsTemporalScalability** and **dwMaxMBperSecFourResolutionsTemporalScalability** are used when temporal scalability is employed in streams across which all layers consist of one, two, three, and four resolutions, respectively.
- **dwMaxMBperSecOneResolutionTemporalQualityScalability**, **dwMaxMBperSecTwoResolutionsTemporalQualityScalability**, **dwMaxMBperSecThreeResolutionsTemporalQualityScalability**, and **dwMaxMBperSecFourResolutionsTemporalQualityScalability** are used when both

temporal and quality scalability are employed in streams across which all layers consist of one, two, three, and four resolutions.

- **dwMaxMBperSecOneResolutionFullScalability**, **dwMaxMBperSecTwoResolutionsFullScalability**, **dwMaxMBperSecThreeResolutionsFullScalability**, and **dwMaxMBperSecFourResolutionsFullScalability** are used when full SVC scalability is employed in streams across which all layers consist of one, two, three, and four resolutions..

3.3.3.3 Initialization and Run-Time Encoding Control

A layer in a stream is uniquely identified by **wLayerOrViewID**, a combination of four subfields: dependency_id (bits 0-2), quality_id (bits 3-6), temporal_id (bits 7-9) as defined in the H.264 specification, and stream_id (bits 10-12) as defined in Section 3.3.3.1. **wLayerOrViewID** defines the scope of EU controls, which allows different configurations for different layers in different streams. When a control is issued for a specific **wLayerOrViewID**, the control needs to be applied to all layers that have dependency_id, quality_id, temporal_id and stream_id as given by the subfields in **wLayerOrViewID**. After streaming starts, the application can also use EU controls to perform run-time configuration changes.

3.3.3.3.1 Wildcard Masks

To reduce the number of calls, applications may use wildcard masks. A wildcard mask is a **wLayerOrViewID** where one or more of the subfields have all bits set to 1. Table 3-7 below shows the bit values required to enable a Wildcard mask for each of the four subfields in **wLayerOrViewID**.

Table 3-7 Bit Layout of wLayerOrViewID for SVC Wildcard Masks

wLayerOrViewID bits	**reserved**			**stream_id**			**temporal_id**			**quality_id**				**dependency_id**		
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
All Dependency Layers														1	1	1
All Quality Layers										1	1	1	1			
All Temporal Layers							1	1	1							
All Streams				1	1	1										

Using Wildcard masks, **wLayerOrViewID** can be set to 0x007, 0x0078, 0x0380, 0x1FFF to indicate the scope of configuration applies to all dependency layers, all quality layers, all temporal layers, and all layers across all simulcast streams, respectively.

3.3.3.4 Sub-bitstream Definition

Several Encoding Units apply to sub-bitstreams instead of individual SVC layers. The sub-bitstream is determined as follows. Let **wLayerOrViewID** be the result from a GET_CUR request issued to the EU_SELECT_LAYER_CONTROL control. Then, the sub-bitstream is given by all the NAL units that meet all the following conditions:

- stream_id is equal to the stream_id indicated by **wLayerOrViewID**.
- temporal_id is less or equal than temporal_id indicated by **wLayerOrViewID**.
- dependency_id is less than dependency_id indicated by **wLayerOrViewID**, or dependency_id is equal to dependency_id indicated by **wLayerOrViewID** and quality_id is less or equal than quality_id indicated by **wLayerOrViewID**.

3.4 MVC and Simulcast Support

This section provides technical background, detailed descriptions, and examples to illustrate how to support MVC and/or simulcast in this specification. Developers may skip this section if the encoder does not support the generation of MVC bitstreams In that case both bmMVCCapabilities and bmLayoutPerStream shall be set to 0.

3.4.1 MVC Overview

The MVC design bears certain similarity as SVC. Within an access unit, there is one "base view component" that is formatted as an ordinary H.264/AVC coded picture. Within the same access unit, there are one or more additional view components that each represents an additional "non-base view" of a multiview encoding for the same instant in time. Within a stream, each view is uniquely identified by the syntax_element view_id, starting from 0 (corresponding to the base view) to the number of views minus 1. Similar to the constrained posed in SVC streams, the value of temporal_id (when temporal scalability is supported) must be assigned starting from 0 and increased continuously. Thus for a particular layering structure, the values of view_id and temporal_id associated with a view can be determined without ambiguity and used as an unique identifier for that view.

Aside from minor differences in high-level syntax, the encoding format for a non-base view is basically the same as that of the base view. To leverage correlation between adjacent view components, the view components of other views within the same access unit and the preceding view components for the same non-base view (in decoding order) can be used as reference pictures. Thus, both temporal prediction within a single view (across different access units) and inter-view prediction across different views (within the same access unit) are supported in MVC.

3.4.2 MVC Capability Advertisement

The encoder notifies the MVC capabilities in **bmMVCCapabilities** in Video Frame Descriptor. The following is the format description:

Table 3-8 Byte Layout of bmMVCCapabilities

Bitfields	Name
[2-0]	MaxNumOfTemporalLayersMinus1
[10-3]	MaxNumOfViewsMinus1
[31-11]	Reserved

MaxNumOfTemporalLayersMinus1: indicates the maximum number of temporal layers in a bitsteam. A non-zero value of this field indicates the encoder supports the creation of temporal scalable bitstreams. This specification only allows and supports values between 0 and 3.

MaxNumOfViewsMinus1: indicates the maximum number of view components in a bitstream. A non-zero value of this field indicates the encoder supports the generation of MVC bitstreams. This specification only allows and supports values between 0 and 127. For encoders that only support the Stereo High profile, this field is equal to 1 (i.e. the number of supported views is limited to two).

Remark: For encoders that only support the generation of ordinary AVC single-layer streams, **bmMVCCapabilities** shall be set to 0.

3.4.3 MVC Stream/View Configuration

3.4.3.1 Initialization

The encoder indicates the number of simulcast streams and the structure associated with each stream using the **bmLayoutPerStream** field in the Probe/Commit Control. These simulcast streams are MVC multi-view streams. In this specification, at most four MVC simulcast streams are allowed and supported, and they are indexed with stream_id 0, 1, 2, and 3. The following tables define the details of the **bmLayoutPerStream** field when used to negotiate an MVC usage.

Table 3-9 Byte Layout of bmLayoutPerStream for MVC

MVC_STR3[63:48]	MVC_STR2[47:32]	MVC_STR1[31:16]	MVC_STR0[15:0]

bmLayoutPerStream consists of four 16-bit subfields. Each subfield describes the layering structure of one simulcast stream in a simulcast transport. To identify an individual stream, this specification uses the terminology of MVC_STR*x* (*x* = 0, 1, 2 and 3) where x is the stream_id of the stream.The subfields are interpreted as shown in the table below.

Table 3-10 Bit Layout of bmLayoutPerStream subfields for MVC

Bitfields	Name
[0-2]	NumOfTemporalLayers
[3-10]	NumberOfViewsMinus1
[11-15]	Reserved

NumOfTemporalLayers: indicates the number of temporal layers in the bitstream. This value effectively corresponds to the values of syntax element temporal_id in the H.264 MVC specification. For example, if this field is 3, three temporal layers, corresponding to temporal_id 0, 1, and 2 are present in the bitstream. The value 0 indicates the corresponding stream is not present. The value of this field must not exceed the maximum number of temporal layers specified in **bmMVCCapabilities**.

NumOfViewsMinus1: indicates the number of views in the bitstream. This value effectively corresponds to the values of syntax element view_id in the H.264 MVC specification. For example, if this field is 1, two view components, corresponding to view_id 0 and 1 are present in the bitstream. The value of this field must not exceed the maximum number of view components specified in **bmMVCCapabilities**.

For each 16-bit subfield, a non-zero value indicates the presence of the corresponding simulcast stream. For encoders that only support the generation of one single ordinary AVC single-layer streams, one and only one of the subfields may be set to 1. For encoders that support MVC but no simulcast capability, one and only one of the subfields may be greater than or equal to 1.

3.4.3.2 Configuration Constraint

In the Probe/Commit Control, the encoder specifies the number of active simulcast streams and the MVC configuration corresponding to each simulcast stream in **bmLayoutPerStream**. The valid configurations are constrained by the maximum number of macroblocks per second similar to the SVC case specified in Section 3.3.3.2. When the encoder is constrained to the MVC Stereo High profile or the MVC Multiview profile, the maxmimum number of macroblocks per second for temporal scalability is considered.

3.4.3.3 Initialization and Run-Time Encoding Control

After the probe is done, the caller uses encoding unit control to deliver detailed per-layer or per-view configuration before issuing the Commit control. A layer/view in a stream is uniquely identified by wLayerOrViewId, a combination of three subfields: view_id (bits 0-6) and temporal_id (bits 7-9) as defined in the H.264 MVC specification, and stream_id (bits 10-12) as defined in Section 3.3.3.1. **wLayerOrViewID** defines the scope of EU controls, which allows different configurations for different layers/views in different streams. When a control is issued for a specific wLayerOrViewID, the control needs to be applied to all layers/views that have view_id, temporal_id and stream_id as given by the subfields in **wLayerOrViewID**. After streaming starts, the application can also use EU controls to perform run-time configuration changes.

3.4.3.3.1 Wildcard Masks

To reduce the number of calls, applications may use wildcard masks. A wildcard mask is a **wLayerOrViewID** where one or more of the subfields have all bits set to 1. Table 3-11 below shows the bit values of **wLayerOrViewID** when using wildcard masks for the three supported MVC subfields.

Table 3-11 Bit Layout of wLayerOrViewID for MVC Wildcard Masks

wLayerOrViewID bits	**reserved**			**stream_id**			**temporal_id**			**view_id**						
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
All Views										1	1	1	1	1	1	1
All Temporal Layers							1	1	1							
All Streams				1	1	1										

As an example, if the host wishes to select all views in all streams, then bits 0-6 and bits 10-12 will be set to 1. This results in a hexadecimal word value of 0x1C7F

4 Examples

Each H.264 frame, as defined in this specification, is considered a single video sample. A video sample is made up of one or more *payload transfers* (as defined in the USB Device Class Specification for Video Devices). In this payload specification, only the Isochronous Transfer IN and Bulk Transfer IN cases will be shown. The layout of the video samples in Isochronous OUT and Bulk OUT transfers parallel these examples.

4.1 Isochronous IN: IDR Frame Followed by Non-IDR Frame

For an isochronous pipe, each (micro) frame will contain a single payload transfer. Each payload transfer will consist of a payload header immediately followed by payload data in one or more data transactions (up to 3 data transactions for high speed high bandwidth endpoints).

The example shown in Figure 4-1 below demonstrates the relationship between H.264 NAL Units, USB Payload Transfers and the token and data packets when receiving isochronous transfers from the device. This example is based on the common occurrence of an IDR frame followed by a non-IDR frame. The IDR frame contains a Slice NALU that spans three transfers. This Slice NALU represents a single capture time so the FID, PTS, and SCR should be identical in the payload header for the first three transfers. Because it contains the end of the video sample, the payload header on the third transfer must set the EOF bit. Also, in this example the device does not support the Slice Modes control so EOS is optional. The fourth transfer contains a new Access Unit (AU) associated with a new capture time. For this transfer, the FID bit must be toggled and new values for PTS and SCR are expected. Also, since the video sample is wholly contained in one transfer, the EOF bit must also be set.

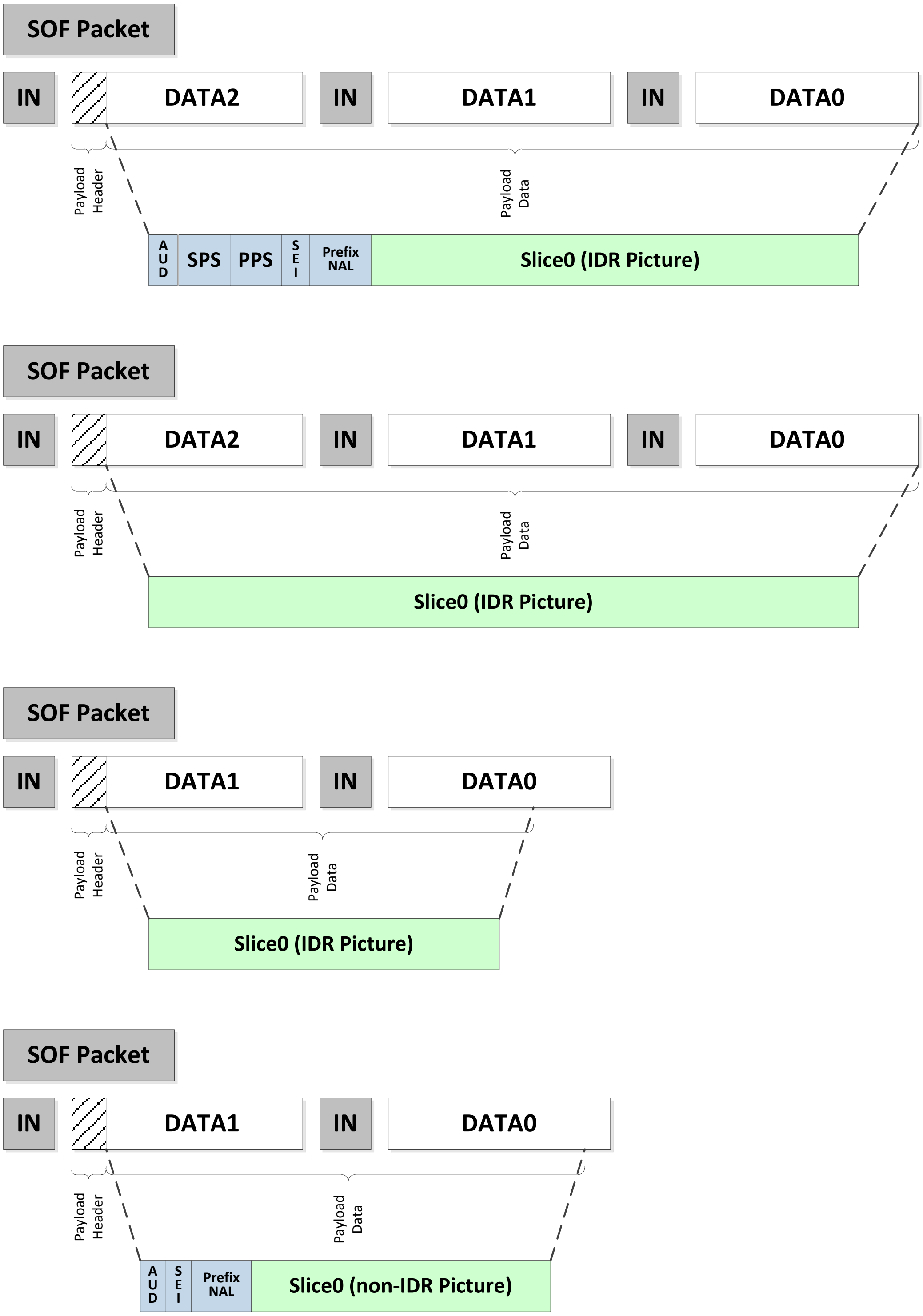


Figure 4-1 Isochronous Example Multiple Transfers per Slice NAL

4.2 Bulk IN: IDR Frame

The example shown in Figure 4-1 below demonstrates the relationship between H.264 NAL Units, USB Payload Transfers and the token and data packets when receiving bulk transfers from the device. Handshake packets are not shown for the sake of clarity.

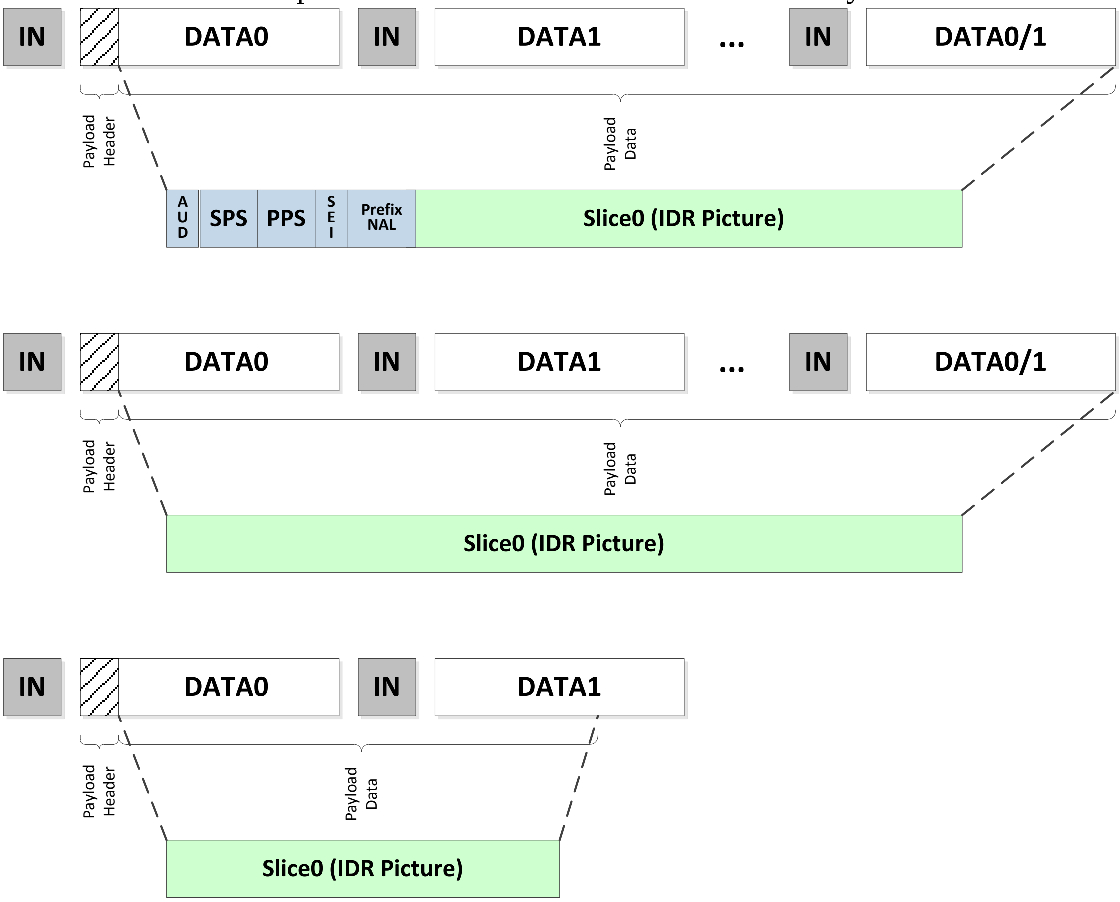


Figure 4-2 Bulk Example Multiple Transfers per Slice NAL

4.3 Isochronous IN: Multiple Slices per Video Sample

The example below is based on a solution that delivers a video sample in three separate Slice NALs. Figure 4-3 demonstrates the proper way to transfer this video sample; using three separate transfers, one Slice NAL per transfer. Figure 4-4 demonstrates the improper way to transfer multiple Slice NALs by placing more than one Slice NAL in the same transfer.

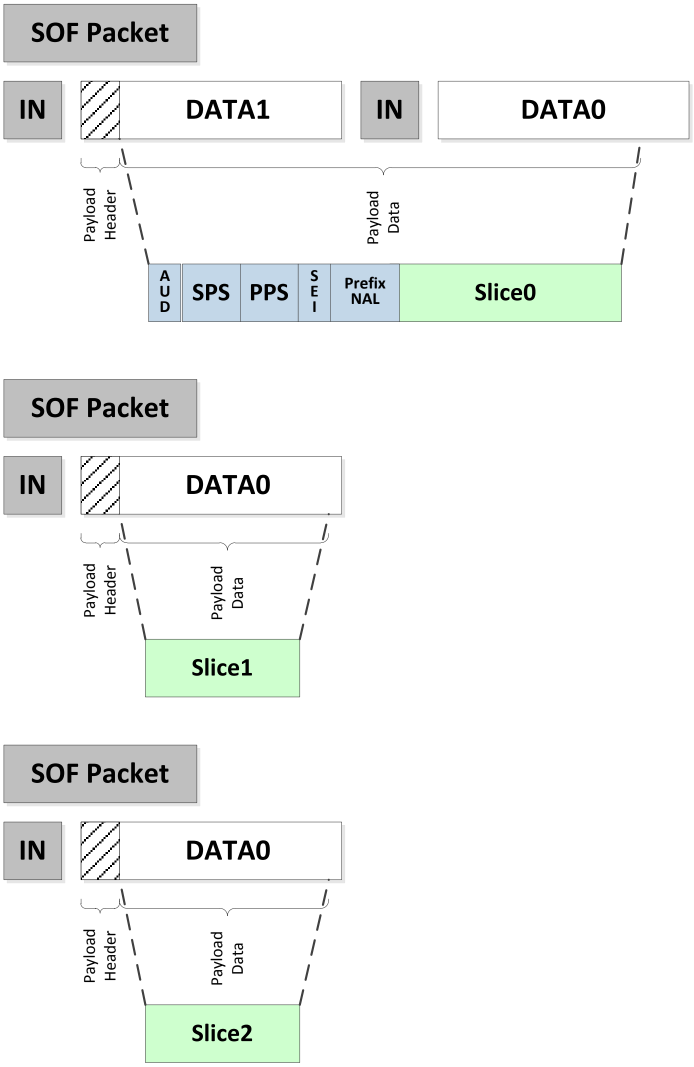


Figure 4-3 Example Multiple Slice NALs per Video Sample

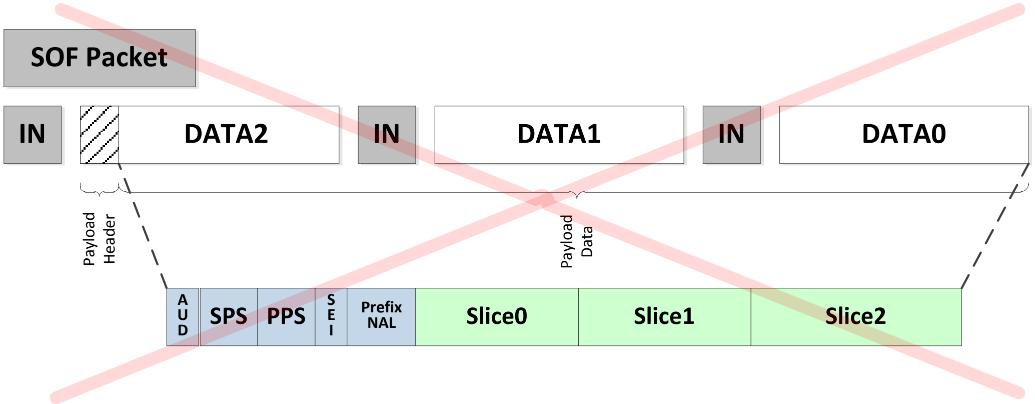


Figure 4-4 Example Incorrect Transfer of Multiple Slice NALs per Video Sample