Universal Serial Bus Device Class Definition for Video Devices: Version 1.5 Examples

Revision 1.5
August 9, 2012

Contributors

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
Chandrashekhar Rao.	Logitech Inc.
Chris Yokum	MCCI Corporation
Stephen Cooper	Microsoft Corp.
Maribel Figuera	Microsoft Corp.
Richard Webb	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

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

Revision History

Version	Date	Description
1.0	July 25, 2012	Initial version, released as part of UVC 1.5

Table of Contents

List of Tables

List of Figures

1 H.264 Simulcast Example

1.1 Introduction

1.1.1 Purpose

This document describes how to configure a device that supports one video control interface with two video streaming interfaces, an uncompressed video format and an H.264 simulcast video format that comply with the USB Video Class specification.

1.1.2 Scope

The scope of this document is to illustrate the additional features that were added to the USB Video Class specification to support H.264 simulcast streaming and encoder control.

1.1.3 Device Description

The device described in this section is a high-speed enabled USB webcam with H.264 encoding capability. In this example, the device is able to support concurrent streaming of uncompressed video format and H.264 simulcast. Both Video Streaming Interfaces are under the same Video Control Interface. The purpose of this document is illustrating how to configure an H.264 Simulcast payload. Configuring the uncompressed payload is covered in a previous example.

The specific features the H.264 Simulcast format supports in this example are:

- Up to two single H.264 streams multiplexed into a single H.264 simulcast stream
- A maximum macro block (MB) processing rate of 244,800 MB/s for H.264 single stream. For example, a maximum resolution of 1080p ((1920*1088)*30/(256) = 244,800).
- A maximum macro block (MB) processing rate of 169,200 MB/s for H.264 simulcasts. For example, the simulcast stream may consist of 720p and 540p at 30fps ((960*544+1280*720)*30/(256) = 169,200).
- Three rate control methods: Constant QP, VBR and GVBR with underflow allowed.
- Eight H.264 frame descriptors (e.g. 4 resolutions x 2 H.264 profiles per resolution):
    - 4 different resolutions: 1080p (1920x1080), 720p (1280x720), 540p (960x540), and 360p (640x360)
    - For each resolution, the device supports 2 profiles: constrained baseline and constrained high
    - Each frame descriptor supports 2 frame intervals: dwFrameInterval = 333,333 (30 Hz) and dwFrameInterval = 666,666 (15 Hz)
    - Each frame descriptor supports 2 bUsage values: UC Config mode 0 and UC Config mode 1

Figure 1-1 represents the internal topology of the camera.

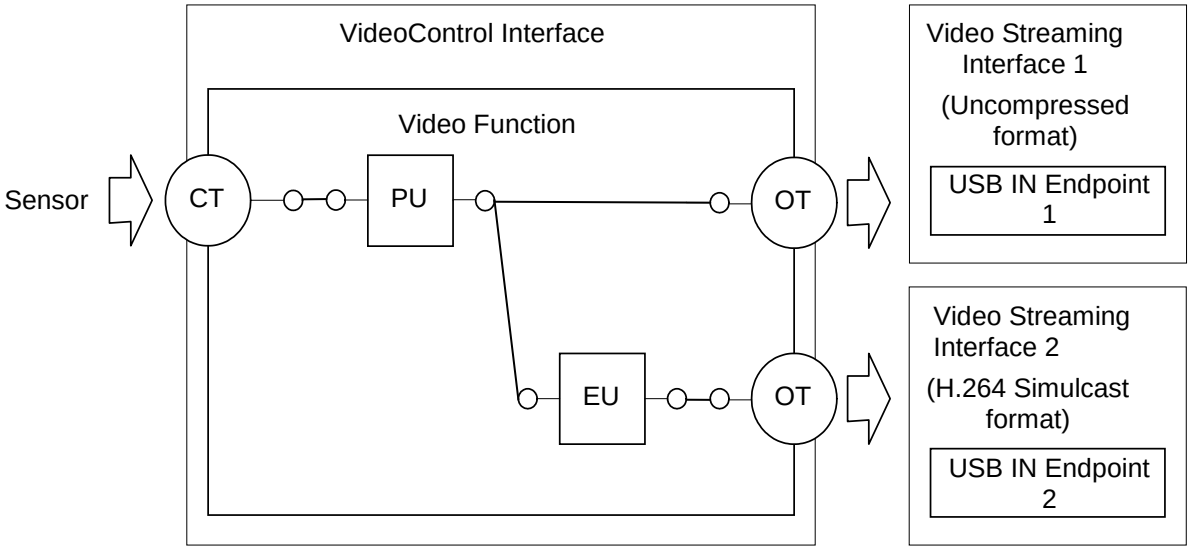

Figure 1-1 USB Video Camera Topology

1.1.4 Encoding Unit Controls

The device supports the following Encoding Unit controls before and after streaming has started:

- Select Layer
- Video Resolution
- Minimum Frame Interval
- Average Bit Rate
- CPB Size
- Quantization Parameter
- Synchronization and Long-Term Reference Frame

1.2 Descriptors

Descriptors are used by USB devices to report their attributes. In this section we illustrate the following Descriptors provided by the device:

- Standard VC Interface Descriptor
- Video Class-Specific VS Interface Input Header Descriptor
- Encoding Unit Descriptor
- H.264 Video Format Descriptor
- H.264 Video Frame Descriptors

Table 1-1 Standard VC Interface Descriptor

Field	Value	Description
bLength	0x09	
bDescriptorType	0x04	
bInterfaceNumber	**0x02**	
bAlternateSetting	0x00	
bNumEndpoints	0x00	
bInterfaceClass	0x0E	Video Interface Class
bInterfaceSubClass	0x02	Video Streaming Interface SubClass
bInterfaceProtocol	0x00	
iInterface	0x00	

Table 1-2 Class-Specific VS Interface Input Header Descriptor

Field	Value	Description
bLength	0x0E	
bDescriptorType	0x24	
bDescriptorSubtype	0x01	
bNumFormats	0x01	One video payload format (simulcast H.264) supported by this interface
wTotalLength	0x04EE	
bEndpointAddress	0x83	Direction: IN - EndpointID: 3
bmInfo	0x00	Dynamic format change not supported
bTerminalLink	0x09	Connected to Output Terminal ID 9
bStillCaptureMethod	0x01	Still image capture method 1
bTriggerSupport	0x00	No hardware triggering support
bTriggerUsage	0x00	
bControlSize	0x01	
bmaControls(1)	0x00	None supported

Table 1-3 Encoding Unit Descriptor

Field	Value	Description
bLength	0x0B	
bDescriptorType	0x24	CS_INTERFACE
bDescriptorSubtype	0x07	VC_ENCODING_UNIT
bUnitID	0x05	
bSourceID	0x04	
iEncoding	0x00	
bControlSize	0x02	2x2 bytes of controls follows
bmControls	0x06CD	D00 = 1 yes - Select Layer D01 = 0 no - Profile and Toolset D02 = 1 yes - Video Resolution D03 = 1 yes - Minimum Frame Interval D04 = 0 no - Slice Mode D05 = 0 no - Rate Control Mode D06 = 1 yes - Average Bit Rate D07 = 1 yes - CPB Size D08 = 0 no - Peak Bit Rate D09 = 1 yes - Quantization Parameter D10 = 1 yes - Synchronization and Long-Term Reference Frame D11 = 0 no - Long-Term Buffer Size D12 = 0 no - Picture Long-Term Reference D13 = 0 no - Valid LTR D14 = 0 no - Level IDC D15 = 0 no - SEI Message
bmControlsRuntime	0x06CD	D00 = 1 yes - Select Layer D01 = 0 no - Profile and Toolset D02 = 1 yes - Video Resolution D03 = 1 yes - Minimum Frame Interval D04 = 0 no - Slice Mode D05 = 0 no - Rate Control Mode D06 = 1 yes - Average Bit Rate D07 = 1 yes - CPB Size D08 = 0 no - Peak Bit Rate D09 = 1 yes - Quantization Parameter D10 = 1 yes - Synchronization and Long-Term Reference Frame D11 = 0 no - Long-Term Buffer Size D12 = 0 no - Picture Long-Term Reference D13 = 0 no - Valid LTR D14 = 0 no - Level IDC D15 = 0 no - SEI Message

Table 1-4 Video Streaming H.264 Format Descriptor

Field	Value	Description
bLength	0x34	
bDescriptorType	0x24	
bDescriptorSubtype	**0x15**	**VS_FORMAT_H264_SIMULCAST**
bFormatIndex	0x01	
bNumFrameDescriptors	0x08	Four resolutions * two H.264 profiles per resolution
bDefaultFrameIndex	0x04	
bMaxCodecConfigDelay	0x01	1 frame
bmSupportedSliceModes	0x00	1 slice per frame only
bmSupportedSyncFrameTypes	0x03	Reset, IDR frame with SPS and PPS
bResolutionScaling	0x03	Limited to resolutions reported by the associated Frame Descriptors
Reserved1	0x00	
bmSupportedRateControlModes	0x0D	VBR with underflow, Constant QP, GVBR with underflow
wMaxMBperSecOneResolutionNoScalability	0x00F4	244,800 MacroBlocks/sec
wMaxMBperSecTwoResolutionsNoScalability	0x00A9	169,200 MacroBlocks/sec
wMaxMBperSecThreeResolutionsNoScalability	0x0000	
wMaxMBperSecFourResolutionsNoScalability	0x0000	
wMaxMBperSecOneResolutionTemporalScalability	0x00F4	244,800 MacroBlocks/sec
wMaxMBperSecTwoResolutionsTemporalScalablility	0x00A9	169,200 MacroBlocks/sec
wMaxMBperSecThreeResolutionsTemporalScalability	0x0000	
wMaxMBperSecFourResolutionsTemporalScalability	0x0000	
wMaxMBperSecOneResolutionTemporalQualityScalability	0x0000	
wMaxMBperSecTwoResolutionsTemporalQualityScalability	0x0000	
wMaxMBperSecThreeResolutionsTemporalQualityScalablity	0x0000	
wMaxMBperSecFourResolutionsTemporalQualityScalability	0x0000	
wMaxMBperSecOneResolutionsTemporalSpatialScalability	0x0000	

wMaxMBperSecTwoResolutionsTemporalSpatialScalability	0x0000	
wMaxMBperSecThreeResolutionsTemporalSpatialScalability	0x0000	
wMaxMBperSecFourResolutionsTemporalSpatialScalability	0x0000	
wMaxMBperSecOneResolutionFullScalability	0x0000	
wMaxMBperSecTwoResolutionsFullScalability	0x0000	
wMaxMBperSecThreeResolutionsFullScalability	0x0000	
wMaxMBperSecFourResolutionsFullScalability	0x0000	

Table 1-5 Video Streaming H.264 Frame Descriptors

Field	Constr. Baseline 1080p	Constr. High 1080p	Constr. Baseline 720p	Constr. High 720p (default)	Constr. Baseline 540p	Constr. High 540p	Constr. Baseline 360p	Constr. High 360p
bFrameIndex	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
wProfile	0x4240	0x640C	0x4240	0x640C	0x4240	0x640C	0x4240	0x640C
bmCapabilities	0x0021	0x002B	0x0021	0x002B	0x0021	0x002B	0x0021	0x002B
wWidth	0x0780 (1920)		0x0500 (1280)		0x03C0 (960)		0x0280 (640)	
wHeight	0x0438 (1080)		0x02D0 (720)		0x021C (540)		0x0168 (360)	
bLevelIDC	0x28 (4.0)		0x20 (3.2)		0x1F (3.1)		0x1E (3.0)	
dwMinBitRate	0x0007A120 (500,000 bps)		0x00061A80 (400,000 bps)		0x000493E0 (300,000 bps)		0x000186A0 (100,000 bps)	
dwMaxBitRate	0x01312D00 (20,000,000 bps)		0x01312D00 (20,000,000 bps)		0x00D59F80 (14,000,000 bps)		0x00989680 (10,000,000 bps)	
bLength	0x34							
bDescriptorType	0x24							
bDescriptorSubtype	0x14 (VS_FRAME_H264)							
wSARwidth	0x0001							
wSARheight	0x0001							
wConstrainedToolset	0x0000 (Reserved)							
bmSupportedUsages	0x00000003 (UC Config modes 0 & 1)							
bmSVCCapabilities	0x00000001 (max of two temporal layers supported)							
bmMVCCapabilities	0x00000000							
dwDefaultFrameInterval	0x00051615 (30 Hz)							
bNumFrameIntervals	0x02							
dwFrameInterval[0]	0x00051615 (30 Hz)							
dwFrameInterval[1]	0x000A2C2A (15 Hz)							

1.3 Scenario

Simulcast of 720p 30fps and 360p 30fps streams, both using UC Config mode 1 with two temporal layers.

1.4 Negotiation

This section shows how the host negotiates a simulcast transport stream that consists of two multiplexed H.264 streams that have different resolutions. The fields that start with "wMaxMBperSec" in the Video Format Descriptor indicate that a simulcast stream generated by this device can support up to two different resolutions. This is given by the non-zero value of wMaxMBperSecTwoResolutionsNoScalability for a simulcast payload composed of multiplexed AVC streams and of wMaxMBperSecTwoResolutionsTemporalScalability for a simulcast payload composed of temporal scalable streams.

Note that the value of wMaxMBperSecTwoResolutions (169,200 MB/s) indicates that 1080p at 30 fps (1920x1088x30/256 = 244800 MB/s) is not supported. A different way to discover this restriction is to leverage GET_MAX, and is illustrated in step 3 in the sequences given below.

Initially, the host selects a simulcast payload composed of two UC Config mode 1 H.264 streams with two temporal layers each where the highest resolution is 1080p and the second resolution is 360p. Once it discovers that at 1080p the device does not support simulcast of two H.264 streams, the host instead selects a simulcast payload with two H.264 streams where the highest resolution is 720p. The 720p stream corresponds to the stream with stream_id = 0 and is set to use VBR low delay rate control mode. The second stream corresponds to the stream with stream_id = 1 and is set to Constant QP rate control mode. The resolution of the second stream is configured to 360p once the device has an active state. This is illustrated in section 0.

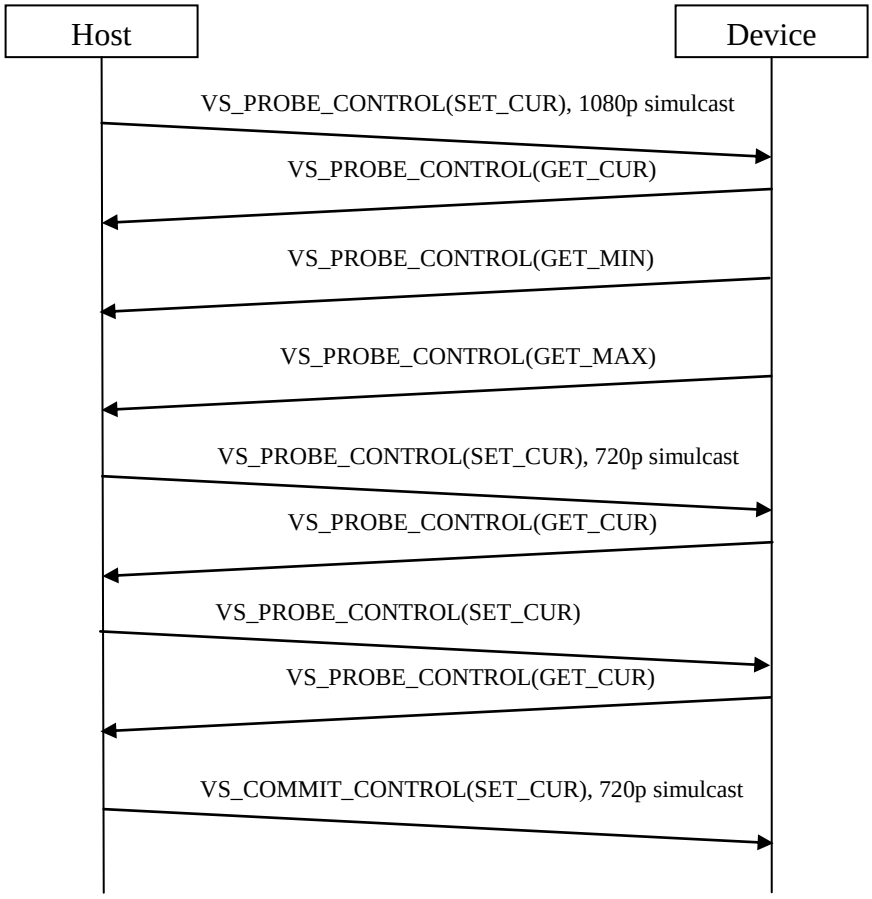

Figure 1-2 Sequence for Negotiating a Simulcast Stream

Figure 1-2 illustrates the communication between host and device during the Probe and Commit stage. The individual steps are:

1) The host sets the streaming interface Probe state by issuing a SET_CUR request to the VS_PROBE_CONTROL with all the fields set to 0 except the following:
    a. bFormatIndex = 0x01
    b. bFrameIndex = 0x02
    c. dwFrameInterval = 0x00051615
    d. bUsage = 0x02
    e. bmLayoutPerStream = 0x0000000000020002 (two H.264 streams, each with 2 temporal layers)
2) Given that at 1080p the device cannot support simulcast of two H.264 streams, upon a GET_CUR request to the VS_PROBE_CONTROL, the device returns a GET_CUR state with bmLayoutPerStream changed to 0x0000000000000002.
3) Next, the host issues a GET_MIN and a GET_MAX request to the VS_PROBE_CONTROL and the device returns bmLayoutPerStream = 0x0000000000000001 and bmLayoutPerStream = 0x0000000000000002, respectively, indicating that the device can support simulcast of one UC Config mode 1 1080p stream with one or two temporal layers.

4) The host now issues a SET_CUR request to the VS_PROBE_CONTROL with all the fields set to 0 except the following:
    a. bFormatIndex = 0x01
    b. bFrameIndex = 0x04
    c. dwFrameInterval = 0x00051615
    d. bUsage = 0x02
    e. bmLayoutPerStream = 0x0000000000020002 (two H.264 streams, each with 2 temporal layers)
5) Upon a GET_CUR request to the VS_PROBE_CONTROL, the device returns the following state:

Table 1-6 GET_CUR Probe state

Control Selector		VS_PROBE_CONTROL		
USB Request		GET_CUR		
Offset	Field	Size	Value	Description
0	**bmHint**	2	0x0000	
2	**bFormatIndex**	1	0x01	
3	**bFrameIndex**	1	0x04	720p Constrained High
4	**dwFrameInterval**	4	0x00051615	30 Hz
8	**wKeyFrameRate**	2	0x0000	Unsupported by payload
10	**wPFrameRate**	2	0x0000	Unsupported by payload
12	**wCompQuality**	2	0x0000	Unsupported by payload
14	**wCompWindowSize**	2	0x0000	Unsupported by payload
16	**wDelay**	2	0x0000	
18	**dwMaxVideoFrameSize**	4	0x000E1000	
22	**dwMaxPayloadTransferSize**	4	0x0400	
26	**dwClockFrequency**	4	0x08F0D180	
30	**bmFramingInfo**	1	0x03	FID and EOF are present in the payload header
31	**bPreferedVersion**	1	0x00	
32	**bMinVersion**	1	0x00	
33	**bMaxVersion**	1	0x00	
34	**bUsage**	1	0x02	UC Config mode 1
35	**bBitDepthLuma**	1	0x08	
36	**bmSetting**	1	0x2A	CABAC, separate QP for luma/chroma, no picture reordering
37	**bMaxNumberOfRefFramesPlus1**	1	0x02	
38	**bmRateControlModes**	2	0x0011	Both H.264 streams in the H.264 simulcast payload are set to VBR low delay

40	**bmLayoutPerStream**	8	0x0000000000020002	2 temporal layers per stream

6) Host changes the rate control mode of the second stream to Constant QP mode by issuing a SET_CUR request to the VS_PROBE_CONTROL with bmLayoutPerStream = 0x0031 and with the remaining fields set to those values the device returned in the GET_CUR state of step 5.
7) Upon a GET_CUR request to the VS_PROBE_CONTROL, the device returns the same Probe data structure as the one set by the host in step 6.
8) The host sets the active device state by issuing a SET_CUR request to the VS_COMMIT_CONTROL where all the field values match the GET_CUR state of step 7. Table 1-7 shows the field values of the Commit data structure.

Table 1-7 SET_CUR Commit data structure

Control Selector		VS_COMMIT_CONTROL		
USB Request		SET_CUR		
Offset	Field	Size	Value	Description
0	**bmHint**	2	0x0000	
2	**bFormatIndex**	1	0x01	
3	**bFrameIndex**	1	0x04	720p Constrained High
4	**dwFrameInterval**	4	0x00051615	30 Hz
8	**wKeyFrameRate**	2	0x0000	
10	**wPFrameRate**	2	0x0000	
12	**wCompQuality**	2	0x0000	
14	**wCompWindowSize**	2	0x0000	
16	**wDelay**	2	0x0000	
18	**dwMaxVideoFrameSize**	4	0x000E1000	
22	**dwMaxPayloadTransferSize**	4	0x0400	
26	**dwClockFrequency**	4	0x08F0D180	
30	**bmFramingInfo**	1	0x03	FID and EOF are present in the payload header
31	**bPreferedVersion**	1	0x00	
32	**bMinVersion**	1	0x00	
33	**bMaxVersion**	1	0x00	
34	**bUsage**	1	0x02	UC Config mode 1
35	**bBitDepthLuma**	1	0x08	
36	**bmSetting**	1	0x2A	CABAC, separate QP for luma/chroma, no picture reordering
37	**bMaxNumberOfRefFramesPlus1**	1	0x02	
38	**bmRateControlModes**	2	0x0031	The first stream (i.e. the

				stream with stream_id = 0) in the simulcast payload is set to VBR low delay. The second stream (i.e. the stream with stream_id = 1) is set to Constant QP.
40	**bmLayoutPerStream**	8	0x0000000000020002	2 temporal layers per stream

Note that the value of **bmSettings** is set to 0x2A, establishing CABAC as the entropy encoding method. The host could have selected CAVLC by setting **bmSettings** to 0x29.

1.5 Configuration using the Encoding Units Prior to Streaming

This section shows how the host configures the resolution for the second stream and the rate control parameters for each stream.

After the SET_CUR request to the VS_COMMIT_CONTROL, the device establishes the bit rate of the 720p stream as follows:

- Base Layer: 2.5 Mbps
- Stream (base layer + enhancement layer): 4 Mbps

The host reduces the base layer to 800 Kbps and the overall stream bit rate to 1.2Mbps. Note that since the current bit rate for the base layer exceeds 1.2Mbps, the host needs to first reduce the base layer bit rate and later the stream bit rate. In addition, the host also reduces the CPB size for each sub-bitstream using 500ms as the leaky bucket period.

After the SET_CUR request to the VS_COMMIT_CONTROL, the host configures the resolution of the second stream in the simulcast payload to 360p. It then sets the QP values for the base and enhancement layer to 34. Once the streams are configured, the host issues a SET_INTERFACE request to start streaming.

Figure 1-3 illustrates the sequence of USB requests to the Encoding Unit controls. For simplicity, note that GET_MIN and GET_MAX requests have not been included in the Figure. However, if an Encoding Unit control supports the GET_MIN and GET_MAX requests, then prior to the initial SET_CUR request to that control, the host should issue a GET_MIN and GET_MAX request to find out the supported range and therefore minimize the probability of getting a protocol stall Out of Range response from the device because the host attempted to set an unsupported value.

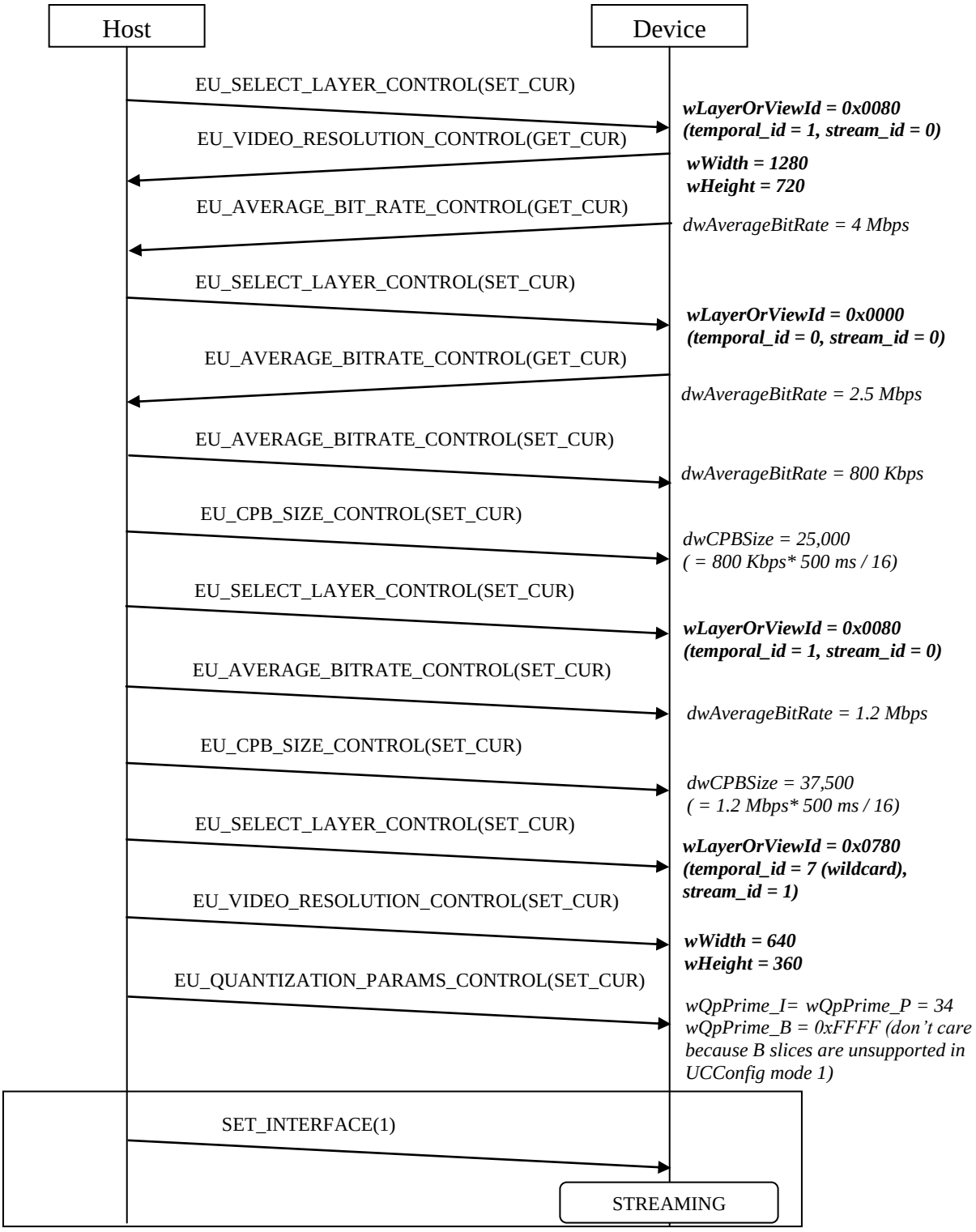

Figure 1-3 Configuration of each stream in the simulcast payload prior to streaming

1.6 Dynamic Configuration using the Encoding Units While Streaming

Figure 1-4 shows a sample of USB requests the host issues while streaming. In this example, the host increases the QP value of the base and enhancement layers of the 360p stream to 37 and 40, respectively. The bit rate of that stream needs to be further reduced and the host reduces the frame rate of that stream to 15 Hz. Later on, the host issues a request for an IDR frame for both 720p and 360p streams.

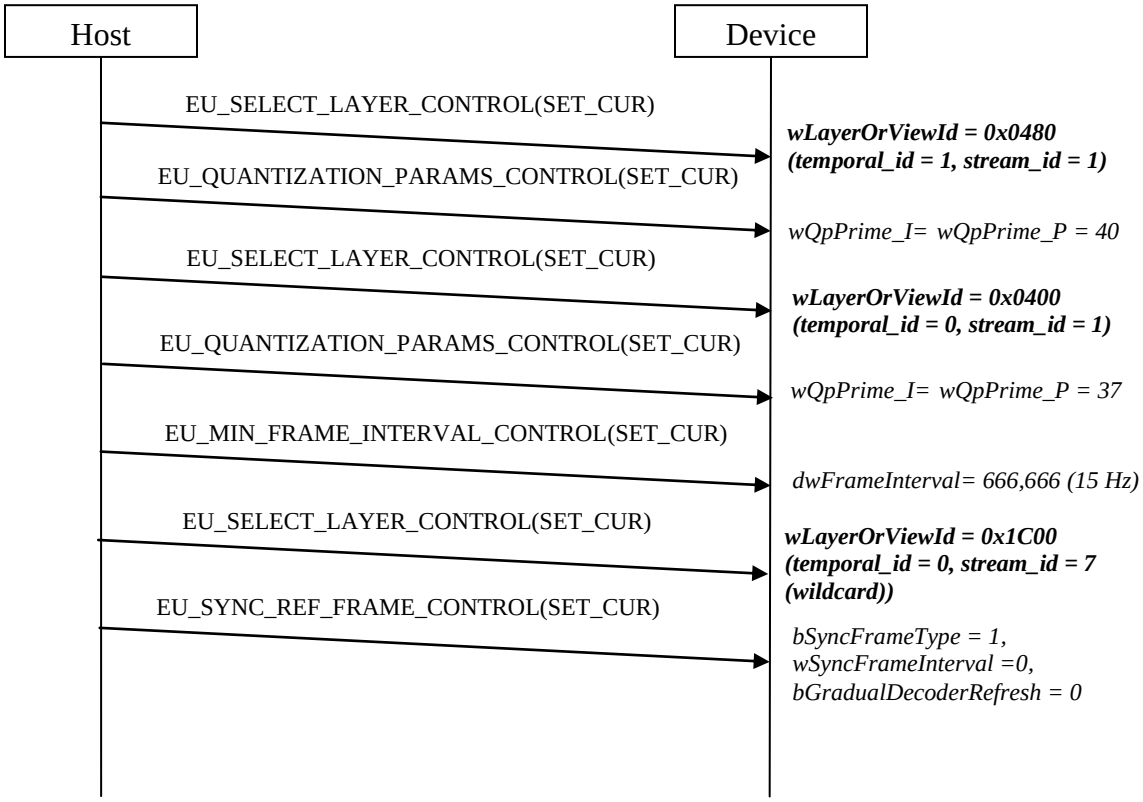

Figure 1-4 Dynamic configuration while streaming

2 Webcam with VP8 Encoding Capability

2.1 Product Description

The device described in this section is a high-speed enabled USB webcam with encoding capability. This example implementation has an image sensor and it streams uncompressed in YUY2 format at VGA resolution (640 x 480) at varying frame rates (7.5, 15, 30 fps) and compressed video data in VP8 format at many possible frame sizes (at maximum 1920x1080) at varying frame rates (7.5, 15, 30 fps), and functions as an asynchronous source, using its internal clock as a reference. The device also contains an image signal processor that is capable of adjusting the brightness and contrast levels of the video stream and encoder which is capable of compressing video signal into compressed VP8 video bitstream. This example implementation uses one Video Interface Collection. The VideoControl interface (interface number 0), the VideoStreaming interface 1 (interface number 1) and the VideoStreaming interface 2 (interface number 2) are part of this Video Interface Collection.
The following figure represents the internal topology of the camera.

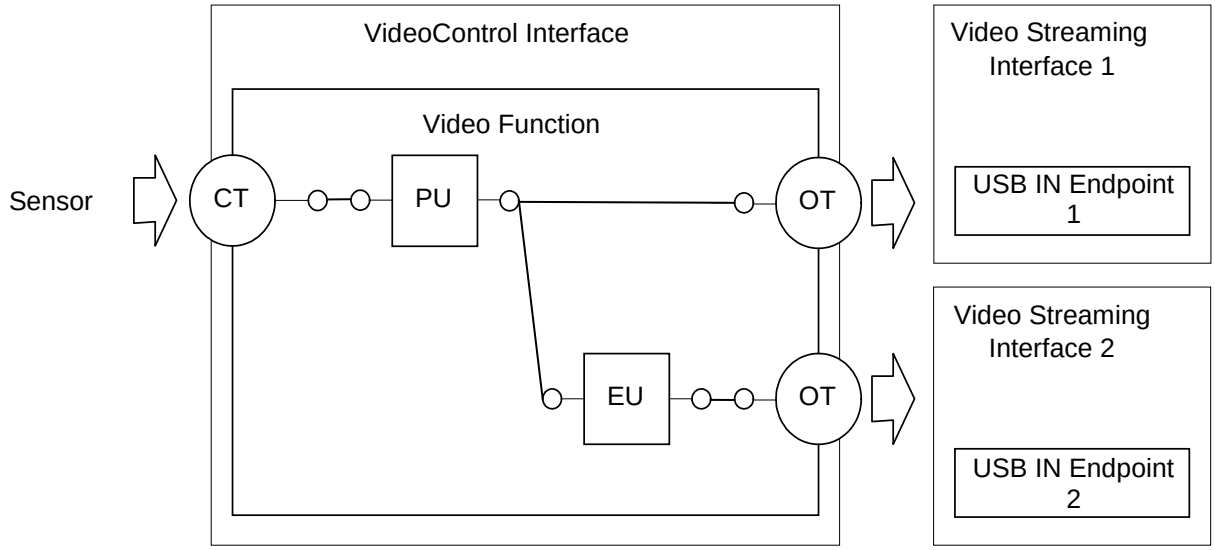

Figure 2-1 USB Video Camera Topology

The video function contains a Camera Terminal representing the sensor. The video streams captured by the Camera Terminals go through any necessary analogue-to-digital conversion, and are routed into a Processing Unit for video signal processing. The output from Processing Unit fans out. It is routed to both Output Terminal for preview stream – which transmits the uncompressed preview bitstream to the host via an USB IN endpoint – and Encoding Unit for video compression. The output from Encoding Unit is routed to an Output Terminal which transmits the compressed video bitstream to the host via another USB IN endpoint. Both USB-IN endpoints are part of the single VideoStreaming interface that this device contains. The internals of the video function (unit and terminal topology) are presented to the host through the (mandatory) VideoControl interface.

2.2 Descriptor Hierarchy

This USB camera device uses a Video Interface Collection that includes:

1. VideoControl interface (interface 0),

2. VideoStreaming interface 1 (interface 1) for uncompressed preview bitstream, and
3. VideoStreaming interface 2 (interface 2) for compressed video bitstream.

VideoStreaming interface 1 features two alternate settings. The first alternate setting (0) has zero bandwidth associated with it (implied by the lack of an isochronous endpoint), so switching to this alternate setting frees all allocated bandwidth on the USB for this device. Alternate setting 1 is the operational part of the interface and contains the isochronous endpoint to supply the host with uncompressed video data.

VideoStreaming interface 2 features also two alternate settings. The first alternate setting (0) has zero bandwidth associated with it. Alternate setting 1 is the operational part of the interface and contains the isochronous endpoint to supply the host with VP8 encoded video data.

Following figure illustrates the descriptor hierarchy.

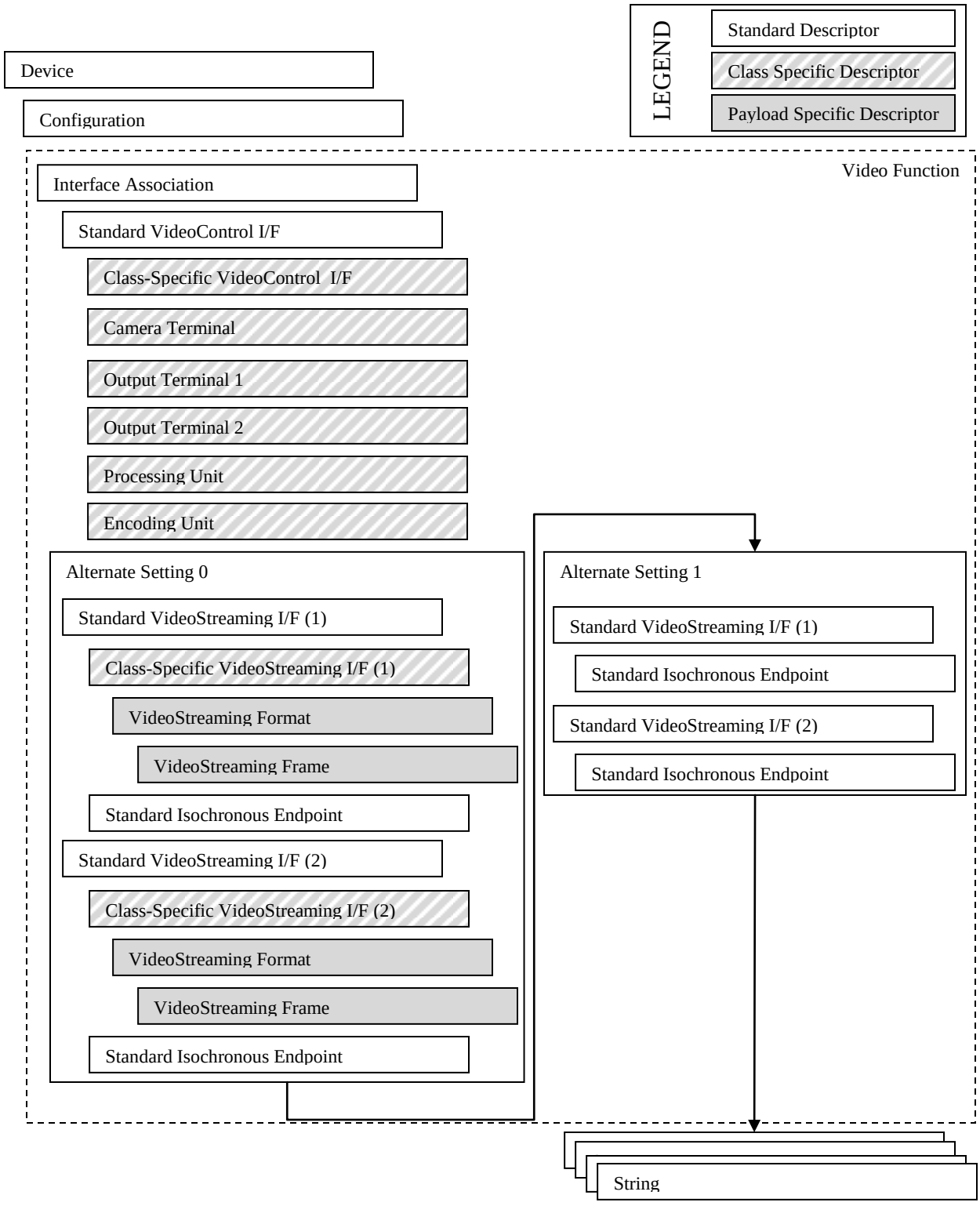

Figure 2-2 USB Video Camera Descriptor Hierarchy.

2.3 Descriptors

The following sections present the class-specific Encoding Unit and all VP8 payload specific descriptors that are used to describe the device to the host. For reference on how to configure the generic and other class-specific descriptors see other examples.

2.3.1 Encoding Unit Descriptor

This descriptor describes the encoding unit that processed the video stream data that is delivered by the processing unit. This implementation supports Select Layer, Video Resolution and Start or Stop Layer/View controls both at initialization time and runtime.

Table 2-1 Encoding Unit Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0E	Size of this descriptor, 14 bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x07	VC_ENCODING_UNIT
3	**bUnitID**	1	0x05	This unit is #5.
4	**bSourceID**	1	0x04	This input pin of this unit is connected to the output pin of unit #4.
5	**iEncoding**	1	0x02	Index of the string descriptor identifying the encoding unit (Product string).
7	**bControlSize**	1	0x03	Size of the **bmControls** and **bmControlsRuntime** fields, in bytes.
8	**bmControls**	3	0x010005	Supports Select Layer (D0), Video Resolution (D2) and Start or Stop Layer/View (D16) controls at initialization time.
11	**bmControlsRuntime**	3	0x010005	Supports Select Layer (D0), Video Resolution (D2) and Start or Stop Layer/View (D16) controls at runtime.

2.3.2 Class-specific VS Format Descriptor

This descriptor describes the video formats supported by the device. This implementation only supports VP8 format on varying frame sizes.

Table 2-2 Class-specific VS Format Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x0D	Size of this descriptor, 13 bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x18	VS_FORMAT_VP8_SIMULCAST
3	**bFormatIndex**	1	0x01	First (and only) format descriptor
4	**bNumFrameDescriptors**	1	0x01	One frame descriptor for this format follows.
5	**bDefaultFrameIndex**	1	0x01	Frame index #1 is default.
6	**bMaxCodecConfigDelay**	1	0x03	Encoder will assume new configuration within three frames from receiving it.

7	**bSupportedPartitionCount**	1	0x01	Implementation supports only one partition per frame.
8	**bmSupportedSyncFrameTypes**	1	0x02	Only supported sync frame type is Intra Frame.
9	**bResolutionScaling**	1	0x03	Limited to resolutions reported by the associated Frame Descriptors.
10	**bmSupportedRateControlModes**	1	0x01	Implementation supports variable bitrate mode.
11	**wMaxMBPerSec**	2	0x0003BC40	Device supports maximum throughput of 244,800 macroblocks per second.

2.3.3 Class-specific VS Frame Descriptor

This descriptor describes the frame and bandwidth settings supported by the device with the video format described by the preceding format descriptor. Supported frame rates are 30, 15 and 7.5 frames per second.

Table 2-3 Class-specific VS Frame Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	0x2B	Size of this descriptor, 43 bytes.
1	**bDescriptorType**	1	0x24	CS_INTERFACE
2	**bDescriptorSubtype**	1	0x17	VS_FRAME_VP8
3	**bFrameIndex**	1	0x01	Index #1.
4	**wWidth**	2	0x0280	Width of frame is 1280 pixels.
6	**wHeight**	2	0x0168	Height of frame is 720 pixels.
8	**bmSupportedUsages**	4	0x00008003	Real-time (D0), Real-time with temporal layering (D1) and File Storage Mode with I and P frames (D16) supported.
12	**bmCapabilities**	2	0x04	Supports Constant Frame Rate.
14	**bmScalabilityCapabilities**	4	0x0004	At maximum three temporal enhancement layers supported.
18	**dwMinBitRate**	4	0x0007A120	Minimum bitrate 500,000 bits/s.
22	**dwMaxBitRate**	4	0x001E8480	Maximum bitrate 20,000,000 bits/s.
26	**dwDefaultFrameInterval**	4	0x00051615	Default frame interval is 333,333ns (30fps).
30	**bNumFrameIntervals**	1	0x0000003	Three supported frame intervals.
31	**dwFrameInterval(1)**	4	0x00051615	Default supported frame interval is 333,333ns (30fps).
35	**dwFrameInterval(2)**	4	0x000A2C28	Supported frame interval of 666,666ns (15fps).
39	**dwFrameInterval(3)**	4	0x00145850	Longest supported frame interval is 1,333,333ns (7,5fps).

2.4 Requests

Following example shows how the host and device collaborate through controls to configure the device to stream a preview stream for local preview and two compressed simulcast streams, both with one temporal enhancement layer. First simulcast stream will be configured to a resolution of 1280x720 with global average bit rate of 1 Mbit/s. Second simulcast stream will be configured to a resolution of 640x360 with global average bit rate of 400 kbit/s. After that, the host sets the device into a streaming state and video starts to stream through the Video Streaming Interfaces of the device.

Once streaming host decides to stop streaming of the temporal enhancement layer on both simulcast streams and enhances the quality of picture in a region of interest that is placed on the bottom right corner of the viewport.

2.4.1 Probe & Commit for Video Streaming Interface Two

Probe & Commit for the Video Streaming Interface streaming VP8 Payload goes through the usual negotiation according to the rules set forth in the USB-UVC 1.5 specification. Table 2-4 presents a valid negotiated value for the VS_COMMIT_CONTROL(SET_CUR) request.

Table 2-4 VS_COMMIT_CONTROL(SET_CUR) Request to VSI Two.

Control Selector		VS_COMMIT_CONTROL		
Request		SET_CUR		
wLength		34		
Offset	Field	Size	Value	Description
0	**bmHint**	2	0x000F	**dwFrameInterval** (D0), **wKeyFrameRate** (D1), **wPFrameRate** (D2), **wCompQuality** (D3), and **wCompWindowSize** (D4) to be kept fixed.
2	**bFormatIndex**	1	0x01	First video payload format.
3	**bFrameIndex**	1	0x01	First frame descriptor frame type.
4	**dwFrameInterval**	4	0x00051615	30 frames per second. (333,333 ns)
8	**wKeyFrameRate**	2	0x0000	N/A
10	**wPFrameRate**	2	0x0000	N/A
12	**wCompQuality**	2	0x0000	N/A
14	**wCompWindowSize**	2	0x0000	N/A
16	**wDelay**	2	0x0021	33ms internal latency.

18	**dwMaxVideoFrameSize**	4	0x00100000	1,048,576 bytes.
22	**dwMaxPayloadTransferSize**	4	0x00100000	1,048,576 bytes.
26	**dwClockFrequency**	4	0x00001F40	Device has 8 kHz clock.
30	**bmFramingInfo**	1	0x01	Frame ID bit is toggling per each frame in the video payload header.
31	**bPreferedVersion**	1	0x00	Version 1.0
32	**bMinVersion**	1	0x00	Version 1.0
33	**bMaxVersion**	1	0x00	Version 1.0
34	**bUsage**	1	0x02	Mode 2: Real-time with Temporal Layering.
35	**bBitDepthLuma**	1	0x08	8 bits.
36	**bmSetting**	1	0x00	No special settings.
37	**bMaxNumberOfRefFramesPlus1**	1	0x03	Supports previous, golden and alternate reference frames.
38	**bmRateControlModes**	2	0x0044[1]	Streams 0 and 1 will have Global Variable Bitrate Rate Control mode.
40	**bmLayoutPerStream**	8	0x0000000001EB01EB[2]	Two Simulcast Streams (stream_id 0 and 1) are enabled, both with one

[1] Field consist of four 8-bit values for simulcast streams with stream_id={0, 1, 2, 3}. Streams 2 and 3 are disabled. Streams 0 and 1 are configured for mode 4, which is the Global VBR mode.
[2] Field consists of four 16-bit values for simulcast streams with stream_id={0, 1, 2, 3}. Streams 2 and 3 are disabled. Streams 0 and 1 have the following configuration (for field spec see **bmLayoutPerStream** explanation and for temporal layer explanation see section "Temporal Layering with VP8 Encoders" in VP8 Payload Format Specification):

		Temporal enhancement layer #3			Temporal enhancement layer #2			Temporal enhancement layer #1			Temporal base layer					
	Reserved	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	Golden allowed	Alt allowed	Prev allowed	# of temporal enh		stream enabled number
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	1
Hex	0				1			E			B					

				temporal enhancement layer enabled. According to the structure set temporal base layer may depend on previous and golden frame and temporal enhancement layer may depend on all available reference frames on both simulcast streams.

2.4.2 Sequence Diagram

Figure 2-3 presents the sequence diagram for the scenario. It is important to note that host sets up the simulcast in a way that the total throughput stays under the device's total maximum throughput as specified by **bMaxMBPerSec** field of the VP8 format descriptor. The 1280x720 stream at 30 frames per second has throughput of 108,000 MB/s and the 640x360 stream at 30 frames per second has throughput of 27,000 MB/s, which yields total throughput of 135,000 MB/s. This is significantly lower than the maximum throughput of this device (244,800 MB/s). In this example wildcard masks for **wLayerOrViewID** are used to select all the temporal layers on each of the two streams before streaming. After streaming has started, a wildcard mask is also used to select temporal enhancement layer 1 on both simulcast streams.
Finally, the CT_REGION_OF_INTEREST (RoI) control is applied to the Camera Terminal in the device topology. This means that it will affect both simulcast streams, and that the specified RoI is expressed in global sensor coordinates prior to any scaling by encoding unit.

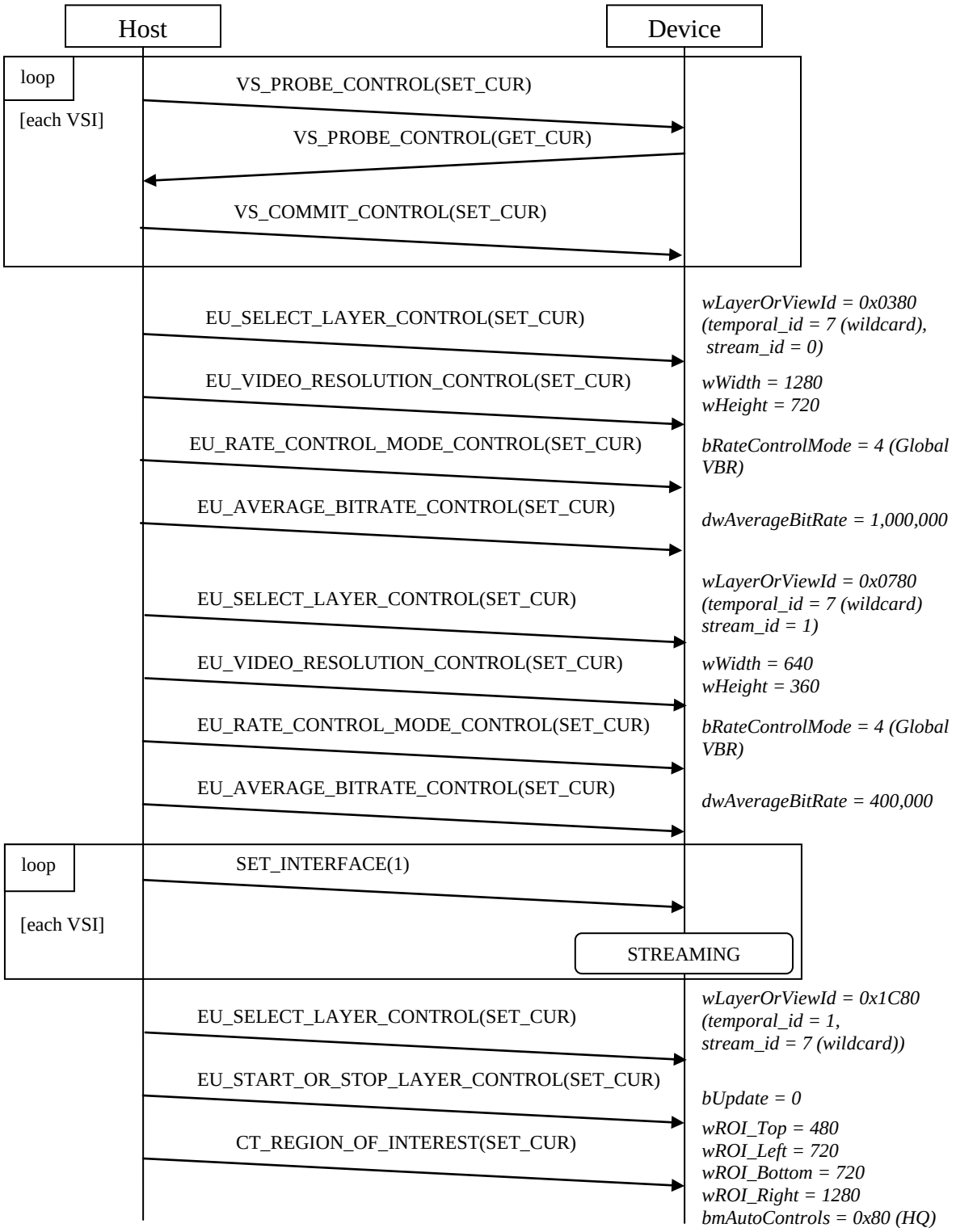

Figure 2-3 Sequence for Configuring Preview Stream and Two Simulcast Streams.

3 The UVC 1.5 backward compatibility example:

This example demonstrates how to create a device that supports both UVC 1.0 and UVC 1.5 behavior.

3.1 Device Descriptor

The device will expose the Device Descriptor with bNumConfigurations = 0x02. The device can have multiple Configuration Descriptors to support UVC 1.5 or UVC 1.0. The host may select the specific Configuration Descriptor or scan through the entire available Configuration Descriptors and select the most appropriate one

===>Device Descriptor<===

bLength:	0x12
bDescriptorType:	0x01
bcdUSB:	0x0200
bDeviceClass:	0xEF →This is a Multi-interface Function Code Device
bDeviceSubClass:	0x02 →This is the Common Class Sub Class
bDeviceProtocol:	0x01 →This is the Interface Association Descriptor protocol
bMaxPacketSize0:	0x40 = (64) Bytes
idVendor:	0x046D
idProduct:	0x0823
bcdDevice:	0x0010
iManufacturer:	0x00
iProduct:	0x00
iSerialNumber:	0x00
bNumConfigurations:	**0x02**

3.2 First Configuration:

The first configuration is indicated by bConfigurationValue=0x01.

===>Configuration Descriptor<===

bLength:	0x09
bDescriptorType:	0x02
wTotalLength:	0x0CC2
bNumInterfaces:	0x04
bConfigurationValue:	**0x01 (first configuration)**
iConfiguration:	0x00
bmAttributes:	0x80
MaxPower:	0xFA

===>Class-Specific Video Control Interface Header Descriptor<===

bLength:	0x0D
bDescriptorType:	0x24
bDescriptorSubtype:	0x01
bcdUVC:	**0x0100 (UVC Version 1.0)**
wTotalLength:	0x00BD
dwClockFrequency:	0x02DC6C00

bInCollection: 0x01
baInterfaceNr[1]: 0x01

3.3 Second Configuration:

The Second configuration is indicated by bConfigurationValue=0x02.

===>Configuration Descriptor<===
bLength: 0x09
bDescriptorType: 0x02
wTotalLength: 0x0CC2
bNumInterfaces: 0x04
bConfigurationValue: 0x02 (second configuration)
iConfiguration: 0x00
bmAttributes: 0x80
MaxPower: 0xFA

===>Class-Specific Video Control Interface Header Descriptor<===
bLength: 0x0D
bDescriptorType: 0x24
bDescriptorSubtype: 0x01
bcdUVC: 0x0150 (UVC Version 1.5)
wTotalLength: 0x00BD
dwClockFrequency: 0x02DC6C00
bInCollection: 0x01
baInterfaceNr[1]: 0x01