[MS-H264PF]:

RTP Payload Format for H.264 Video Streams Extensions

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	2.0	Major	Significantly changed the technical content.
7/30/2013	2.1	Minor	Clarified the meaning of the technical content.
11/18/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.2	Minor	Clarified the meaning of the technical content.
7/31/2014	3.0	Major	Significantly changed the technical content.
10/30/2014	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	4.0	Major	Significantly changed the technical content.
6/30/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	5.0	Major	Significantly changed the technical content.
7/1/2016	6.0	Major	Significantly changed the technical content.
9/14/2016	6.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	7.0	Major	Significantly changed the technical content.
12/12/2017	7.1	Minor	Clarified the meaning of the technical content.
4/27/2018	8.0	Major	Significantly changed the technical content.
7/24/2018	9.0	Major	Significantly changed the technical content.
8/28/2018	10.0	Major	Significantly changed the technical content.
12/11/2018	10.1	Minor	Clarified the meaning of the technical content.
3/19/2019	10.2	Minor	Clarified the meaning of the technical content.
4/7/2021	11.0	Major	Significantly changed the technical content.
8/17/2021	12.0	Major	Significantly changed the technical content.
4/29/2022	13.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The RTP Payload Format for H.264 Video Streams Extensions protocol describes the payload format to carry real-time video streams in the payload of the **Real-Time Transport Protocol (RTP)**. It is used to transmit and receive real-time video streams in two-party peer-to-peer calls and in multi-party conference calls.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

codec: An algorithm that is used to convert media between digital formats, especially between raw media data and a format that is more suitable for a specific purpose. Encoding converts the raw data to a digital format. Decoding reverses the process.

contributing source (CSRC): A source of a stream of **RTP packets** that has contributed to the combined stream produced by an RTP **mixer**. The **mixer** inserts a list of the synchronization source (SSRC) identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. This list is called the CSRC list. An example application is audio conferencing where a **mixer** indicates all the talkers whose speech was combined to produce the outgoing packet, allowing the receiver to indicate the current talker, even though all the audio packets contain the same SSRC identifier (that of the **mixer**). See [RFC3550] section 3.

forward error correction (FEC): A process in which a sender uses redundancy to enable a receiver to recover from packet loss.

maximum transmission unit (MTU): The size, in bytes, of the largest packet that a given layer of a communications protocol can pass onward.

mixer: An intermediate system that receives a set of media streams of the same type, combines the media in a type-specific manner, and redistributes the result to each participant.

network byte order: The order in which the bytes of a multiple-byte number are transmitted on a network, most significant byte first (in big-endian storage). This may or may not match the order in which numbers are normally stored in memory for a particular processor.

padding: Bytes that are inserted in a data stream to maintain alignment of the protocol requests on natural boundaries.

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

RTP packet: A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources, and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets can be contained if permitted by the encapsulation method. See [RFC3550] section 3.

RTP payload: The data transported by **RTP** in a packet, for example audio samples or compressed video data. For more information, see [RFC3550] section 3.

RTP session: An association among a set of participants who are communicating by using the **Real-Time Transport Protocol (RTP)**, as described in [RFC3550]. Each RTP session maintains a full, separate space of **Synchronization Source (SSRC)** identifiers.

RTVideo: A video stream that carries an RTVC1 bit stream.

synchronization source (SSRC): The source of a stream of **RTP packets**, identified by a 32-bit numeric SSRC identifier carried in the RTP header so as not to be dependent upon the network address. All packets from a synchronization source form part of the same timing and sequence number space, so a receiver groups packets by synchronization source for playback. Examples of synchronization sources include the sender of a stream of packets derived from a signal source such as a microphone or a camera, or an RTP **mixer**. A synchronization source may change its data format (for example, audio encoding) over time. The SSRC identifier is a randomly chosen value meant to be globally unique within a particular **RTP session**. A participant need not use the same SSRC identifier for all the **RTP sessions** in a multimedia session; the binding of the SSRC identifiers is provided through RTCP. If a participant generates multiple streams in one **RTP session**, for example from separate video cameras, each MUST be identified as a different SSRC. See [RFC3550] section 3.

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

video frame: A single still image that is shown as part of a quick succession of images in a video.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[ISO/IEC 14496-10:2014] ISO/IEC, "Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding", http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=66069

[MS-RTP] Microsoft Corporation, "Real-time Transport Protocol (RTP) Extensions".

[MS-SDP] Microsoft Corporation, "Session Description Protocol (SDP) Extensions".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/rfc/rfc2119.html

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003, http://www.ietf.org/rfc/rfc3550.txt

[RFC5109] A. Li, Ed., "RTP Payload Format for Generic Forward Error Correction", December 2007, http://www.ietf.org/rfc/rfc5109.txt

[RFC6184] Wang, Y. K., Even, R., Kristensen, T. et al., "RTP Payload Format for H.264 Video", May 2011, http://www.ietf.org/rfc/rfc6184.txt

[RFC6190] Wenger, S., Wang, Y. K., Schierl, T., et al., "RTP Payload Format for Scalable Video Coding", May 2011, http://www.ietf.org/rfc/rfc6190.txt

1.2.2 Informative References

None.

1.3 Overview

This protocol specifies a payload format to transport an H.264 bitstream using Real-Time Transport Protocol (RTP).

The syntax of this protocol follows the definition in [RFC6190] with the following extensions:

1. Customized Payload Content Scalability Information (PACSI) packet is used to signal the stream layout, **video frame** cropping information, and elementary bitstream information.
2. Simulcast streams are supported. A sender capable of simulcast can send the same video coded sequence in different video resolutions and different video **codecs** at the same time.

1.4 Relationship to Other Protocols

This protocol carries H.264 bitstream, described in [ISO/IEC 14496-10:2014], as a payload, and in turn is carried as a payload in RTP, as described in [MS-RTP].

1.5 Prerequisites/Preconditions

This protocol specifies only the payload format for H.264 video streams. This protocol requires the establishment of an RTP stream, a mechanism to obtain H.264 video access units for it to packetize, and a mechanism to render H.264 video access units that it has depacketized.

Higher layers are required to provide H.264 access units.

1.6 Applicability Statement

This protocol is only applicable for transporting video access units encoded using the H.264 codec.

1.7 Versioning and Capability Negotiation

This protocol has the following versioning constraints:

- **Supported Transports:** This protocol uses **RTP** as its transport as discussed in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol is a payload for the [MS-RTP] transport protocol and therefore relies on RTP for providing the means to transport its payload over the network.

2.2 Message Syntax

The Network Abstraction Layer (NAL) unit format, transmission mode, and packetization mode are the same as defined in [RFC6184] and [RFC6190] with a few extensions<1>.

The Payload Content Scalability Information (PACSI) packet, specified in [RFC6190], MAY be extended by incorporating one or more customized Supplemental Enhancement Information (SEI) NAL units. This protocol defines three types of SEI messages:

1. Stream Layout SEI Message
2. Cropping Info SEI Message
3. Bitstream Info SEI Message

All fields in the messages specified in this protocol are in **Network Byte Order** unless explicitly called out. No emulation prevention byte and no training bit are inserted in these three types of SEI messages.

The start code prefix of a NAL unit can be removed on the wire as RTP packetization is sufficient to identify the beginning of a new NAL unit.

2.2.1 RTP Header Usage

The syntax of the RTP header is specified in [MS-RTP] section 2.2.1. The fields of the fixed RTP header have their usual meaning with the following additional notes:

Marker (M): This bit MUST be set to 1 if the **RTP packet** contains the last packet of a layer of an access unit. The RTP packet MAY be a Video Coding Layer (VCL) NAL unit, as defined in [RFC6184] section 4.1, or an H.264 **forward error correction (FEC)** packet associated with one or more VCL NAL units.

Sequence number: The syntax of this field is defined in [RFC3550], section 5.1.<2>

Timestamp: The syntax of this field is defined in [RFC3550], section 5.1. The sampling clock frequency MUST be 90000 Hz. All RTP packets of the same access unit of a simulcast stream MUST carry the same timestamp. The timestamps of two different simulcast streams are not required to be equal, even if the RTP packets contain VCL NAL units for the same coded picture.

2.2.2 Transmission Mode

The syntax of transmission mode follows the syntax defined in [RFC6190] section 4.4.

This protocol only supports multi-session transmission (MST).

2.2.3 Packetization Mode

The syntax of packetization mode used in this protocol follows the syntax defined in [RFC6184] section 5.4 and [RFC6190] section 4.5.

This protocol only supports Non-interleaved combined timestamp and CS-DON (NI-TC) packetization mode.

2.2.4 NAL Unit Usage

The syntax of the NAL unit format and the meaning of the NAL unit header fields are as defined in [RFC6184] section 5.3 and [RFC6190] section 4.2 with the following additional notes<3>:

- PACSI NAL unit MUST be present in each layer in each access unit. It MUST be the first NAL unit of the layer. The PACSI NAL unit might be aggregated with NAL units into one STAP-A NAL unit. In that case it MUST be the first NAL unit present in the aggregated Single-Time Aggregation Packet type A (STAP-A) NAL unit.
- PACSI NAL unit MUST NOT be fragmented.
- When a NAL unit is larger than the **maximum transmission unit (MTU)** size, it MUST be fragmented into multiple Fragmentation Unit type A (FU-A) NAL units.
- Multiple small NAL units of the same layer of the same access unit MAY be aggregated into one STAP-A NAL unit. The size of STAP-A NAL unit MUST NOT exceed the MTU size.
- All other NAL unit types are passed to the decoder without any processing<4>.

2.2.5 Stream Layout SEI Message

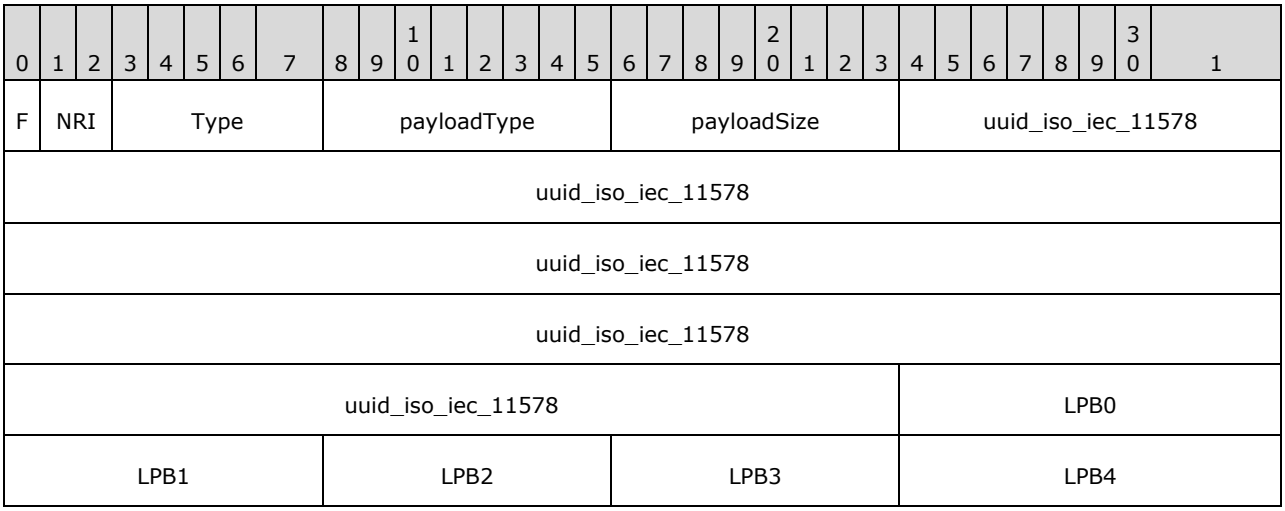
The stream layout is a structure that describes information about all layers present in the current simulcast streams. This provides a reliable way for the receiver to retrieve the information about the simulcast streams without waiting to receive NAL units from all layers.

This protocol defines a User Data Unregistered SEI message as the stream layout message.

The syntax of the User Data Unregistered SEI message followed in this protocol is as defined in [ISO/IEC 14496-10:2014] Annex D.

The stream layout SEI message MUST be embedded in a PACSI NAL unit. The PACSI NAL containing the stream layout SEI message MAY be present in any layer and MAY be followed by any VCL NAL unit.

The format of stream layout SEI message is defined as follows:



LPB5	LPB6	LPB7	R	P
LDSize	Layer Description			
More Layer Description …				

F (1 bit): A forbidden_zero_bit, as specified in [RFC6184], section 1.3.

NRI (2 bits): A nal_ref_idc, as specified in [RFC6184], section 1.3.

Type (5 bits): A nal_unit_type, as specified in [RFC6184], section 1.3. MUST be set to 6.

payloadType (1 byte): A SEI payload type. MUST be set to 5 to indicate User Data Unregistered SEI message. The syntax used by this protocol is as defined in [ISO/IEC 14496-10:2014], section 7.3.2.3.1.

payloadSize (1 byte): SEI payload size. The syntax that is used by this protocol for this field is the same as defined in [ISO/IEC 14496-10:2014], section 7.3.2.3.1. The payloadSize value is the size of the stream layout SEI message excluding the F, NRI, Type, payloadType, and payloadSize fields.

uuid_iso_iec_11578 (16 bytes): A **universally unique identifier (UUID)** to indicate the SEI message is the stream layout. The value MUST be set to {0x139FB1A9-446A-4DEC-8CBF-65B1E12D2CFD}.

LPB0 (1 byte): A layer presence byte #0. The value of each bit indicates whether the layer identified by the corresponding PRID (priority ID) is present in the current simulcast streams. The value of 1 means the layer is present. The value of 0 means the layer is not present. The less significant bit corresponds to a smaller PRID. This byte corresponds to layer PRID 0~7.

LPB1 (1 byte): Layer presence byte #1. This byte corresponds to layer PRID 8~15.

LPB2 (1 byte): Layer presence byte #2. This byte corresponds to layer PRID 16~23.

LPB3 (1 byte): Layer presence byte #3. This byte corresponds to layer PRID 24~31.

LPB4 (1 byte): Layer presence byte #4. This byte corresponds to layer PRID 32~39.

LPB5 (1 byte): Layer presence byte #5. This byte corresponds to layer PRID 40~47.

LPB6 (1 byte): Layer presence byte #6. This byte corresponds to layer PRID 48~55.

LPB7 (1 byte): Layer presence byte #7. This byte corresponds to layer PRID 56~63.

R (7 bits): Reserved bits. MUST be set to 0.

P (1 bit): A layer description presence flag. The value of 1 indicates that the **Layer Description** table and **Layer Description** size (**LDSize**) field are present. The value of 0 indicates that these are not present.

LDSize (1 byte): A Layer Description size. The value indicates the size of the **Layer Description** table. MUST be larger than or equal to 16.

Layer Description (variable): The size is defined by the **LDSize** field. If the value of the **P** field is 1, there MUST be at least one and there MAY be more than one **Layer Description**. If there is no need for **Layer Description**, the **P** field MUST be set to 0 and **LDSize** and the **Layer Description** fields MUST NOT be present.

The format of **Layer Description** is defined as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Coded Width																Coded Height															
Display Width																Display Height															
Bitrate																															
FPSIdx					LT			PRID							CB	R	R2															

Coded Width (2 bytes): A coded width of the resolution of the coded picture.

Coded Height (2 bytes): A coded height of the resolution of the coded picture.

Display Width (2 bytes): A display width of the resolution of the coded picture.

Display Height (2 bytes): A display height of the resolution of the coded picture.

Bitrate (4 bytes): The target bitrate of the coded NAL units in this layer, in bits-per-second unit.

FPSIdx (5 bits): Index of a predefined frame rate table. Each FPSIdx value corresponds to one frame rate value. The frame rate value is the target frame rate represented by this layer. The frame rate table is defined as follows:

FPSIdx	Frame Rate (fps)
0	7.5
1	12.5
2	15
3	25
4	30
5	50
6	60

Only **FPSIdx** values 0~6 are defined.

LT (3 bits): A layer type specifies the values of temporal_id, quality_id and dependency_id allowed within this layer as defined in [ISO/IEC 14496-10:2014].

0: Base layer. The layer contains coded pictures with H.264 syntax elements temporal_id, quality_id and dependency_id all equal to 0.

1: Temporal layer. The layer contains coded pictures with H.264 syntax elements temporal_id greater than 0, and quality_id and dependency_id equal to 0.

2~7: Reserved for future use.

PRID (6 bits): The priority ID associated with this layer. It is used to determine the RTP stream that the Layer Description describes. The PRID value of Layer Description MUST match the priority ID of the Scalable Video Coding (SVC) stream.

CB (1 bit): Constrained baseline profile. The value of 1 indicates the layer is coded in constrained baseline profile. The value of 0 indicates the layer SHOULD NOT be coded in constrained baseline profile.

R (1 bit): Reserved bit for future use. The sender MUST set it to 0. The receiver SHOULD ignore it.

R2 (2 bytes): Reserved for future use. The sender MUST set it to 0. The receiver SHOULD ignore it.

2.2.5.1 Stream Layout Types

There are two types of stream layout SEI messages. They differ by the presence of the **Layer Description** field.

1. Full stream layout.

   The **P** field MUST be "1". There MUST be at least one **Layer Description** field. For each bit of value "1" in the layer presence bytes, there MUST be a **Layer Description** field with PRID as the layer index.

2. Update stream layout.

   The **P** field MUST be "0". The **LDSize** and **Layer Description** fields are not present.

A full stream layout defines the complete stream layout information of simulcast streams. It specifies how many layers are contained in the simulcast streams and the information about each layer. There MUST be one full stream layout PACSI inserted prior to any other NAL unit. More full stream layout PACSIs MAY be inserted as needed. The PACSI NAL unit of the base layer of the first simulcast stream of an IDR access unit MUST carry a full stream layout. The PACSI NAL units of the base layers of other simulcast streams of the same IDR access unit MAY also carry full stream layout optionally to reduce the impact of network packet loss.

An update stream layout defines an update to a previous full stream layout or update stream layout. An update stream layout does not contain any detailed **Layer Description** field, so it MUST only be used to indicate the adding or removing of layers from the simulcast streams through changing the bit values of the layer presence bytes. A layer presence bit MUST NOT be set to "1" if the value of the same bit is "0" in the most recent full stream layout. This means that an update stream layout MUST NOT contain a new layer that is not defined in the previous full stream layout.

2.2.6 Cropping Info SEI Message

The cropping info is a structure that describes the frame coordinates of one or multiple cropping windows. Cropping windows are regions of interest that can be decided by a face tracker or by senders' preference at runtime. The provision of cropping info provides a reliable way for the receiver to properly crop a window out of the original frame without loss of important information when necessary.

This protocol defines a User Data Unregistered SEI message as the cropping info message.

The syntax of the User Data Unregistered SEI message is defined in [ISO/IEC 14496-10:2014] Annex D.

The cropping info SEI message can be embedded in a PACSI NAL unit. The PACSI NAL containing the cropping info SEI message MUST be present in the base layer and SHOULD be followed by any VCL NAL unit.

The format of cropping info SEI message is defined as follows:

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>20</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>30</th><th>1</th></tr>
<tr><td>F</td><td colspan="2">NRI</td><td colspan="3">Type</td><td colspan="8">payloadType</td><td colspan="8">payloadSize</td><td colspan="8">uuid_iso_iec_11578</td></tr>
<tr><td colspan="32">uuid_iso_iec_11578</td></tr>
<tr><td colspan="32">uuid_iso_iec_11578</td></tr>
<tr><td colspan="32">uuid_iso_iec_11578</td></tr>
<tr><td colspan="24">uuid_iso_iec_11578</td><td colspan="8">numOfCropData</td></tr>
<tr><td colspan="8">crop_info_type</td><td colspan="8">frame_crop_confidence_level1</td><td colspan="16">frame_crop_left_offset1</td></tr>
<tr><td colspan="16">frame_crop_right_offset1</td><td colspan="16">frame_crop_top_offset1</td></tr>
<tr><td colspan="16">frame_crop_bottom_offset1</td><td colspan="8">frame_crop_confidence_level2</td><td colspan="8">frame_crop_left_offset2</td></tr>
<tr><td colspan="8">frame_crop_left_offset2</td><td colspan="16">frame_crop_right_offset2</td><td colspan="8">frame_crop_top_offset2</td></tr>
<tr><td colspan="8">frame_crop_top_offset2</td><td colspan="16">frame_crop_bottom_offset2</td><td colspan="8">...</td></tr>
<tr><td colspan="8">frame_crop_confidence_levelN</td><td colspan="16">frame_crop_left_offsetN</td><td colspan="8">frame_crop_right_offsetN</td></tr>
<tr><td colspan="8">frame_crop_right_offsetN</td><td colspan="16">frame_crop_top_offsetN</td><td colspan="8">frame_crop_bottom_offsetN</td></tr>
<tr><td colspan="8">frame_crop_bottom_offsetN</td></tr>
</table>

F bit (1 bit): See the definition in [RFC6190] section 1.1.3.

NRI (2 bits): See the definition in [RFC6190] section 1.1.3.

Type (5 bits): See the definition in [RFC6190] section 1.1.3.

payloadType (1 byte): MUST be 5. See the definition in [ISO/IEC 14496-10:2014].

payloadSize (18 + 9 * N bytes): N is the value of the numOfCropData field. See the definition in [ISO/IEC 14496-10:2014].

uuid_iso_iec_11578 (16 bytes): [Guid("BB7FC1A0-6986-4052-90F0-0929217539CF")]

numOfCropData (1 byte): Total number of cropping info set in the message.

crop_info_type (1 byte): MUST be zero.

frame_crop_confidence_level1 (1 byte): Specify the confidence level, quantified with a value between 0 and 100 with higher value meaning higher confidence, of the first cropping window. The cropping info confidence level is decided by the crop info processor. The value zero indicates the cropping confidence is indeterminate.

frame_crop_right_offset1 (2 bytes): Specify the offset from the right edge of the first cropping window to the right edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_left_offset1 (2 bytes): Specify the offset from the left edge of the first cropping window to the left edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_top_offset1 (2 bytes): Specify the offset from the top edge of the first cropping window to the top edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_bottom_offset1 (2 bytes): Specify the offset from the bottom edge of the first cropping window to the bottom edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_confidence_level2 (1byte): Specify the confidence level, quantified with a value between 0 and 100 with higher value meaning higher confidence, of the second cropping window (if present). The cropping info confidence level is decided by the crop info processor. The value zero indicates the cropping confidence is indeterminate.

frame_crop_right_offset2 (2 bytes): Specify the offset from the right edge of the second cropping window (if present) to the right edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_left_offset2 (2 bytes): Specify the offset from the left edge of the second cropping window (if present) to the left edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_top_offset2 (2 bytes): Specify the offset from the top edge of the second cropping window (if present) to the top edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_bottom_offset2 (2 bytes): Specify the offset from the bottom edge of the second cropping window (if present) to the bottom edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_confidence_levelN (1 byte): Specify the confidence level, quantified with a value between 0 and 100 with higher value meaning higher confidence, of the nth cropping window (if present). The cropping info confidence level is decided by the crop info processor. The value zero indicates the cropping confidence is indeterminate.

frame_crop_right_offsetN (2 bytes): Specify the offset from the right edge of the nth cropping window (if present) to the right edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_left_offsetN (2 bytes): Specify the offset from the left edge of the nth cropping window (if present) to the left edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_top_offsetN (2 bytes): Specify the offset from the top edge of the nth cropping window (if present) to the top edge of the rectangular region of the coded video sequence, in terms of pixels.

frame_crop_bottom_offsetN (2 bytes): Specify the offset from the bottom edge of the nth cropping window (if present) to the bottom edge of the rectangular region of the coded video sequence, in terms of pixels.

The following diagram is an example of the offsets defined earlier in this section.

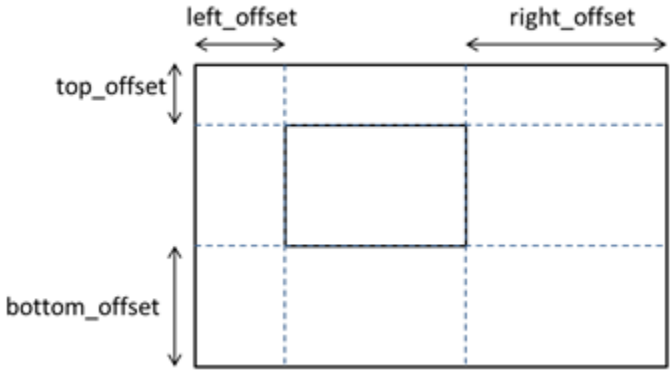


Figure 1: Offsets example

2.2.7 Bitstream Info SEI Message

The bitstream info is a structure that describes the attributes of an H.264 frame. The provision of bitstream info provides a reliable way for the receiver to infer lost or incomplete frames and apply necessary error handling to avoid video artifacts.

This protocol defines a User Data Unregistered SEI message as the bitstream info message.

The syntax of the User Data Unregistered SEI message is defined in [ISO/IEC 14496-10:2014] Annex D.

The bitstream info SEI message MUST be embedded in a PACSI NAL unit. When present, the bitstream info SEI message MUST exist in every frame from a source.

The format of bitstream info SEI message is defined as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
F	NRI		Type					payloadType								payloadSize								uuid_iso_iec_11578							
uuid_iso_iec_11578																															
uuid_iso_iec_11578																															
uuid_iso_iec_11578																															
uuid_iso_iec_11578																								ref_frm_cnt							
num_of_nal_unit																															

F bit (1 bit): See the definition in [RFC6190].

NRI (2 bits): See the definition in [RFC6190].

Type (5 bits): See the definition in [RFC6190].

payloadType (1 byte): MUST be 5. See the definition in [ISO/IEC 14496-10:2014].

payloadSize (1 byte): MUST be greater than or equal to 18. Any additional payload after the first 18 bytes will be ignored and reserved for future use. See the definition in [ISO/IEC 14496-10:2014].

uuid_iso_iec_11578 (16 bytes): [Guid("05FBC6B9-5A80-40E5-A22A-AB4020267E26")]

ref_frm_cnt (1 byte): Specifies the reference frame count of the most recent reference frame (including the current frame). The reference frame count increments by 1 for each reference frame (that is, H.264 syntax element **nal_ref_idc** not equal to 0). The initial value of reference frame count is random. The reference frame count shall continue across coded video sequences (defined in [ISO/IEC 14496-10:2014]) if the bitstream is from the same source.

Note: ref_frm_cnt has a more strict definition than H.264 syntax element **frame_num**. The former is to allow the inference of reference frame losses/corruption so as to avoid video artifacts due to predictively inter-frame coded nature of H.264 bitstreams, while the latter is simply to indicate the decoding order of coded pictures.

num_of_nalu_unit (1 byte): Specifies the number of H.264 NAL units in a frame described by a PACSI NAL unit.

2.2.8 H.264 Forward Error Correction (FEC) Payload Format

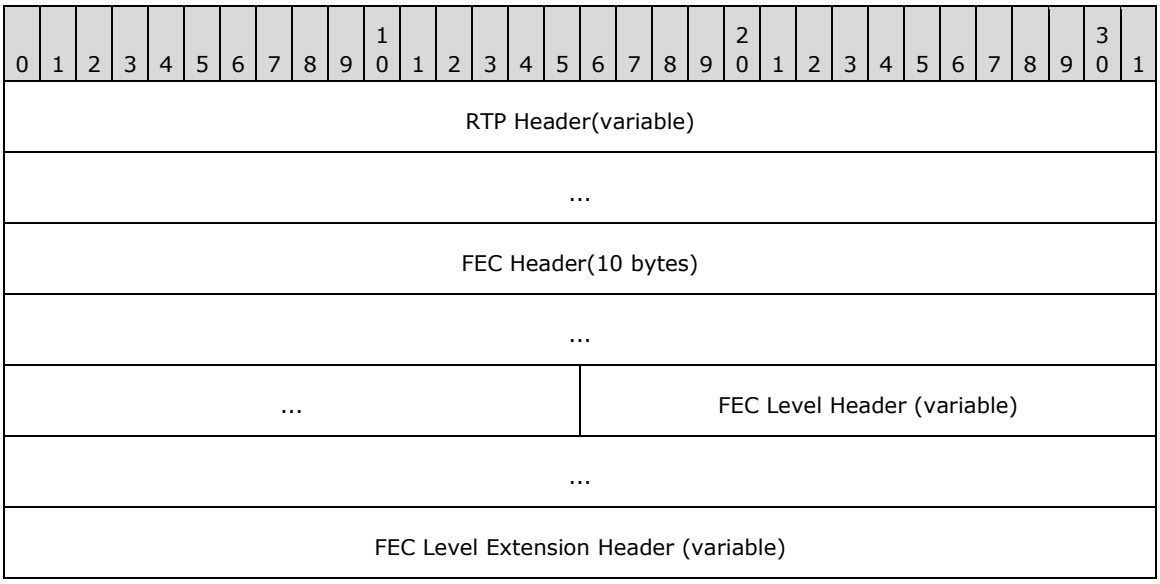
FEC is to use redundant packets to reduce the effect of network packet loss. This protocol uses the FEC payload format defined in [RFC5109] section 7 with some extensions.

2.2.8.1 H.264 FEC Packet Structure

The FEC packet is constructed by placing an FEC header, FEC level header, FEC level extension header, and FEC level payload into the **RTP payload**, as shown in the following table.

There is an extra FEC level extension header in this protocol (section 2.2.8.1.4) that differs from the FEC packet structure, as specified in [RFC5109].

Only one FEC level is allowed, which is similar to FEC level 0, as specified in [RFC5109] section 7.1.



...
FEC Level Payload (variable)
...

2.2.8.1.1 RTP Header for FEC Packets

FEC packets MUST be added to the end of H.264 data packets for each layer for each access unit.

All the fields in the RTP header of FEC packets are used according to [MS-RTP], with some additional notes:

Marker: This field MUST be set to 1 for the last FEC packet of each layer of each access unit. MUST be set to 0 otherwise.

SSRC: The **Synchronization Source (SSRC)** value MUST be the same as the data packets protected by this FEC packet.

Sequence Number: FEC packets MUST share the same numbering space as the data packets it protects.

Timestamp: The timestamp MUST be set to the same value as the data packets protected by this FEC packet.

Payload Type: The payload type for the FEC packets is determined through dynamic, out of band means. The payload type for FEC packets MUST be different from the payload type for the data packets.

2.2.8.1.2 FEC Header for FEC Packets

The FEC header format in this protocol adheres to the FEC header format as specified in [RFC5109] section 7.3 with the exception that it defines a SN Offset field instead of a SN base field.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>20</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>30</th><th>1</th></tr>
<tr><td>E</td><td>L</td><td>P</td><td>X</td><td colspan="4">CC</td><td>M</td><td colspan="6">PT Recovery</td><td colspan="16">SN Offset</td></tr>
<tr><td colspan="32">TS Recovery</td></tr>
<tr><td colspan="16">Length Recovery</td></tr>
</table>

E (1 bit): An Extension flag. It MUST be set to 1.

L (1 bit): A Long mask flag, as defined in [RFC5109], section 7.3.

P (1 bit): A P recovery field, as defined in [RFC5109], section 7.3.

X (1 bit): An X recovery field, as defined in [RFC5109], section 7.3.

CC (4 bits): A CC recovery field, as defined in [RFC5109], section 7.3.

M (1 bit): An M recovery field, as defined in [RFC5109], section 7.3.

PT Recovery (7 bits): A PT recovery field, as defined in [RFC5109], section 7.3.

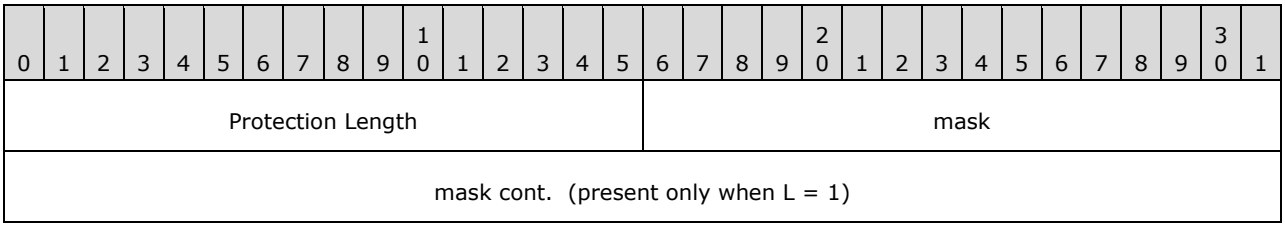
SN Offset (2 bytes): A sequence number (SN) offset field. The value is the difference between the RTP sequence number of the FEC packet and the lowest RTP sequence number of the data packets the FEC packet protects. It is the unsigned 32-bit value calculated by subtracting the lowest RTP sequence number from the RTP sequence number of the FEC packet with sequence number wraparound considered.

TS Recovery (4 bytes): A timestamp (TS) recovery field, as defined in [RFC5109], section 7.3.

Length Recovery (2 bytes): Used to determine the length of any recovered packets. This field is as defined in [RFC5109], section 7.3, with the exception that it does not include the length of the **contributing source (CSRC)** list, extension, and **padding**.

2.2.8.1.3 FEC Level Header for FEC Packets

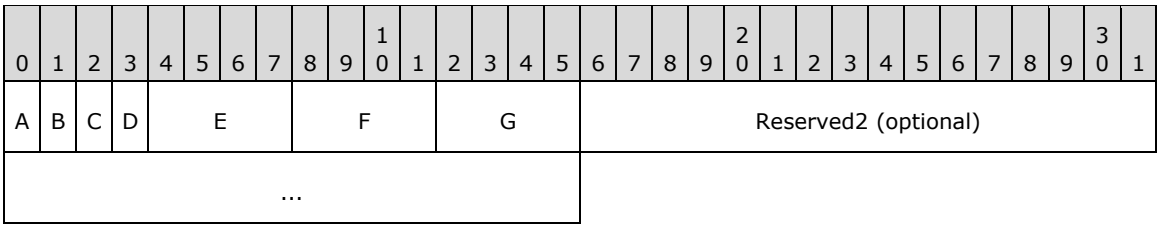
The format of FEC level header is defined in [RFC5109] section 7.4.



The protection length field and the mask field have the same meaning as defined in [RFC5109].

2.2.8.1.4 FEC Level Extension Header

This protocol defines a FEC level extension header between the FEC level header and FEC level payload.



A - V (1 bit): Version field. It indicates whether Reserved2 field is present.

B - C (1 bit): C field. This field MUST be set to 0.

C - HR1 (1 bit): Header recovery bit1 field. It is used to store the one bit in the FEC protection string.

D - HR2 (1 bit): Header recovery bit2 field. It is used to store the one bit in the FEC protection string.

E - Reserved (4 bits): Reserved for future use. It MUST be set to 0.

F - FEC Count (4 bits): The number of FEC packets generated by a protection operation.

G - FEC Index (4 bits): The index of an FEC packet in all the FEC packets generated by a protection operation. The value ranges from 0 to 15. At most 15 FEC packets can be generated by one protection operation.

Reserved2 (4 bytes, optional): Reserved for future use. It is present only when V field is 1.

3 Protocol Details

3.1 Sender Details

This section covers the role of the sender of H.264 NAL units.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Send an H.264 NAL Unit

Whenever higher layers send an H.264 NAL unit and the NAL unit is very small, the sender is not required to send it out right away. Instead, it can buffer the NAL unit, wait to receive the next NAL unit from higher layers, and aggregate the next NAL unit with the previously received or aggregated NAL units into a new STAP-A aggregation packet.

Whenever higher layers send an H.264 NAL unit and the NAL unit does not fit into one RTP packet because the NAL unit size is larger than the **maximum transmission unit (MTU)** size, the NAL unit MUST be fragmented into multiple FU-A NAL units.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Packetization Rules

The sender applies the same packetization rule as defined in [RFC6184] section 6 with the following additional notes<5>:

1. The sender MUST work in non-interleaved mode. Only single NAL unit packets, STAP-A units, and FU-A units are allowed. Single-Time Aggregation Packet type B (STAP-B), Multi-Time Aggregation Packet (MTAP), and Non-interleaved Multi-time Aggregation Packet (NI-MTAP) MUST NOT be used.

2. One **RTP session** MUST only carry NAL units for the same layer. This means all PACSI units and type 20 units (coded slice in scalable extension) MUST have the same PRID value in the same coded sequence.

3. PACSI MUST be the first NAL unit on its own, or the STAP-A packet containing PACSI MUST be the first NAL unit of a layer in an access unit when PACSI is aggregated. A PACSI MUST be the first NAL unit in the aggregated STAP-A packet when the PACSI NAL unit is aggregated. There MUST be only one PACSI NAL unit in any layer in any access unit.

4. The stream layout SEI message MUST be contained in a PACSI NAL unit.

3.1.5.2 Generation of Forward Error Correction (FEC) Packet

The FEC packets are generated by applying a protection operation on protected bit strings derived from the protected media RTP packets.

3.1.5.2.1 Generation of the FEC Header, FEC Level Extension Header and FEC Level Header

For each media packet, the protected bit string (64 bits long) for the generation of the FEC header is formed by concatenating the following fields together in the order specified:

- 2 bits of value 0 (2 bits)
- Padding bit of the media packet (1 bit)
- Extension bit of the media packet (1 bit)
- 4 bits of value 0 (4 bits)
- Marker bit of the media packet (1 bit)
- PT field of the media packet (7 bits)
- 32 bits of value 0 (32 bits)
- Unsigned network-ordered 16-bit representation of the media payload. It MUST NOT include the 12-byte fixed RTP header, CSRC list, extension, and **padding**.

Then the FEC bit strings are formed by applying protection operation on the protected bit strings. There might be multiple FEC bit strings generated by one protection operation. Each FEC bit string is used to form a FEC packet.

The FEC header and the FEC level extension header of each FEC packet are generated from the corresponding FEC bit string as follows:

The first (most significant) bit in the FEC bit string is written into the HR1 field of the FEC level extension header. The next bit in the FEC bit string is written into the HR2 field of the FEC level extension header. The next bit in the FEC bit string is written into the P recovery field of the FEC header. The next bit in the FEC bit string is written into the X recovery field of the FEC header. The next 4 bits in the FEC bit string are written into the CC recovery field of the FEC header. The next bit in the FEC bit string is written into the M recovery field of the FEC header. The next 7 bits in the FEC bit string are written into the PT recovery field of the FEC header. The next 32 bits in the FEC bit string are written into the TS recovery field of the FEC header. The next 16 bits are written into the length recovery field in the FEC header.

The extension (E) field of the FEC header MUST be set to 1. The long mask (L) field MUST be set to 1 if the number of protected media packets is larger than 16 and MUST be set to 0 otherwise.

The SN Offset field of the FEC header is set to an unsigned network-ordered value calculated by subtracting the smallest RTP sequence number from the FEC packet sequence number.

The V field, C field, and Reserved field of the FEC level extension header MUST be set to 0.

The FEC Count field of the FEC level extension header MUST be set to the number of FEC packets generated by the protection operation. The FEC Index field of the FEC level extension header MUST be set to the index of the FEC packet starting from 0.

The protection length field and the mask field of the FEC level header MUST be written as defined in [RFC5109].

3.1.5.2.2 FEC Protection Operation Algorithms

This protocol allows two types of algorithms for FEC protection operation.

- Exclusive OR (XOR)

  XOR is used when the number FEC packet applied to protect one or more media packets is 1.

- Reed-Solomon

  The detail of this algorithm is beyond the discussion of this protocol.

3.1.5.3 Signaling of Simulcast

Multiple simulcast streams MAY be encoded using the same camera feed at the same time. Each simulcast stream MAY be decoded by itself and might contain multiple layers, including one base layer and 0 or more enhancement layers.

The sender signals the simulcast by using stream layout PACSI NAL units<6>. The format of a stream layout PACSI is defined in section 2.2.6 of this protocol.

A full stream layout (section 2.2.5.1) PACSI NAL unit MUST be sent as the very first NAL unit.

Whenever one or more layers are removed, an update stream layout (section 2.2.5.1) or a full stream layout PACSI NAL unit MUST be sent. The update stream layout PACSI NAL unit SHOULD be used if no layer information has changed since the previous full stream layout and there is no need for the layer presence byte fields to be present.

Whenever one or more layers are added and the layer descriptions are not changed, an update stream layout or a full stream layout PACSI NAL unit MUST be sent. For the same reason noted in the previous paragraph, the update stream layout PACSI NAL unit is preferred.

For other changes in the simulcast streams, a full stream layout PACSI NAL unit MUST be sent. This includes a new layer that is never present before it is added or any **Layer Description** field changes.

A stream layout PACSI MAY be sent in any layer. There SHOULD NOT be any VCL data NAL units following it in the same access unit. A stream layout PACSI MUST be sent prior to the change it signals. This means that when a layer is added into the simulcast streams, the stream layout PACSI NAL unit MUST be sent before any other NAL unit of the new layer is sent. After a stream layout PACSI NAL unit is sent with a layer removed from the simulcast streams, no NAL unit of the removed layer is allowed to be sent. This is needed so that the receiver has timely, accurate information about the simulcast streams.

3.1.5.3.1 RTVideo Simulcast Stream

An **RTVideo** simulcast stream is also allowed to coexist with H.264 simulcast streams. Only at most one RTVideo simulcast stream is allowed. The RTVideo simulcast stream is encoded using the same camera feed as other H.264 simulcast streams. An RTVideo simulcast stream is sent on its own RTP stream with its own SSRC.

The presence of RTVideo simulcast stream is not signaled through the stream layout PACSI. Instead, the receiver MUST recognize an RTVideo simulcast stream by looking at RTP payload type in the RTP packets. The RTVideo payload type is negotiated through SDP, as defined in [MS-SDP].

A **mixer** receiving simulcast streams MUST only forward at most one simulcast stream to the downstream endpoints. It MUST find the presence by looking at the stream layout PACSI and forward one of the H.264 simulcast streams if present. If there is no H.264 simulcast stream present, then it MUST forward RTVideo stream when present.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Receiver Details

This section covers the role of receiver of H.264 NAL units.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 DePacketization Rules

When receiving an H.264 RTP packet, the de-packetization process specified in [RFC6184] section 7.1 applies.

Additionally<7>,

- Each layer associated with the same priority ID has its own receiver buffer.
- If the first packet in a RTP stream identified by a SSRC of the same timestamp is neither a PACSI NAL unit nor an STAP-A packet with the PACSI NAL unit as the first NAL unit, then all the packets of the same SSRC and the same timestamp MUST be discarded.
- If a packet is received and it is not stream layout PACSI NAL unit and is not STAP-A packet containing stream layout PACSI NAL unit, it MUST be discarded when any of the following conditions is met:
  - There is no full stream layout PACSI NAL unit received.
  - The Layer Presence bit in the previous stream layout is not set or there is no Layer Description associated with the priority ID of the NAL unit.

3.2.5.2 Recovery Procedures

When the loss of media packets happens, the FEC packets can be used to recover the lost media packets.

This protocol only allows FEC level 0 and only allows the recovery of the full media packet. Partial media packet recovery is not supported.

When an FEC packet is received, check whether any media packet protected by the FEC packet is lost and whether the received FEC packets are enough to recover the loss. In general the lost media packets can be recovered if the number of FEC packets is equal to or larger than the number of lost media packets.

3.2.5.2.1 Recovery of the RTP Header

For each received media packet, the protected bit string is formed as specified in section 3.1.5.2.1.

For each received FEC packet, the FEC bit string (64 bits long) is formed by concatenating the following fields together in the order specified:

- HR1 field of the FEC level extension header (1 bit)
- HR2 field of the FEC level extension header (1 bit)
- P recovery field of the FEC header (1 bit)
- X recovery field of the FEC header (1 bit)
- CC recovery field of the FEC header (4 bits)
- M recovery field of the FEC header (1 bit)
- PT recovery field of the FEC header (7 bits)
- TS recovery field of the FEC header (32 bits)
- Length recovery field of the FEC header

The protected bit strings of the media packets with the corresponding indices, and the FEC bit strings of the FEC packets with the corresponding indices are passed to the recovery operation to generate the recovery bit strings for the lost media packets.

For each recovery bit string, the RTP header of the lost media packet is recovered as follows:

1. Create a new RTP packet.
2. Set the version field in the new packet to 2. Skip the first 2 bits in the recovery bit string.
3. Set the Padding bit in the new packet to the next bit in the recovery bit string.
4. Set the Extension bit in the new packet to the next bit in the recovery bit string.
5. Set the CC field in the new packet to the CC field of the FEC packet. Skip the next 4 bits in the recovery bit string.
6. Set the marker bit in the new packet to the next bit in the recovery bit string.
7. Set the payload type in the new packet to the next 7 bits in the recovery bit string.
8. Set the SN field in the new packet to SNi. The lost packet corresponds to Bit I in the mask field of the FEC header. Then SNi = SN of the FEC packet – SN Offset field + i.

9. Set the TS field to the TS field of the FEC packet. Skip the next 32 bits in the recovery bit string.
10. Set the SSRC of the new packet to the SSRC of the FEC packet.
11. Set the CSRC list of the new packet to be the CSRC list of the FEC packet.
12. Read the next 16 bits in the recovery bit string; this is the length of payload of the new packet in network order.

3.2.5.2.2 Recovery of the RTP Payload

For each received media packet, the protected bit string is the RTP payload excluding the 12-byte fixed RTP header and CSRC list.

For each received FEC packet, the FEC bit string is the FEC level 0 payload.

The protection length field of the FEC level 0 header is the length of the FEC bit string in byte.

If any protected bit string is shorter than protection length, then it MUST be padded to the protection length with 0 at the end.

After the recovery operation is complete, the length of RTP payload of the recovered media packet MAY be shorter than the protection length and MUST be set to the value obtained in section 3.2.5.2.1.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Stream Layout SEI Message

Stream Layout SEI message bytes in **network byte order**:

0x06, 0x05, 0x3a, 0x13, 0x9f, 0xb1, 0xa9, 0x44, 0x6a, 0x4d, 0xec, 0x8c, 0xbf, 0x65, 0xb1, 0xe1, 0x2d, 0x2c, 0xfd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x01, 0x10

The Stream Layout SEI contains fields of the following values:

F=0, NRI=0, Type=6 (SEI)

payloadType=5 (user data unregistered user SEI), payloadSize=0x3a

uuid_iso_iec_11578 = 0x13, 0x9f, 0xb1, 0xa9, 0x44, 0x6a, 0x4d, 0xec, 0x8c, 0xbf, 0x65, 0xb1, 0xe1, 0x2d, 0x2c, 0xfd

LPB0~6=0, LPB7=03 (PRID 56 and PRID 57 are present in this message)

R=0, P=1 (Layer description table is present)

LDsize=0x10

Followed by Layer Description for PRID 56 in network byte order:

0x05, 0x00, 0x02, 0xd0, 0x05, 0x00, 0x02, 0xd0, 0x00, 0x16, 0xe3, 0x60, 0x10, 0xe0, 0x00, 0x00,

The Layer Description for PRID 56 contains fields of the following values:

Coded Width=1280(0x500), Coded Height=720(0x2d0)

Display Width=1280(0x500), Display Height=720(0x2d0)

Bitrate=1500000(0x0016e360)

FPSIndex=2,LayerType=0,PRID=56, CB=0, R=0, R2=0

Followed by Layer Description for PRID 57 in network byte order:

0x05, 0x00, 0x02, 0xd0, 0x05, 0x00, 0x02, 0xd0, 0x00, 0x0f, 0x42, 0x40, 0x21, 0xe4, 0x00, 0x00

The Layer Description for PRID 57 contains fields of the following values:

Coded Width=1280(0x500), Coded Height=720(0x2d0)

Display Width=1280(0x500), Display Height=720(0x2d0)

Bitrate=1000000(0x000f4240)

FPSIndex=4,LayerType=1,PRID=57, CB=0, R=0, R2=0

4.2 Cropping Info SEI Message

Cropping Info SEI message bytes in **network byte order**:

0x06, 0x05, 0x1b, 0xbb, 0x7f, 0xc1, 0xa0, 0x69, 0x86, 0x40, 0x52, 0x90, 0xf0, 0x09, 0x29, 0x21, 0x75, 0x39, 0xcf, 0x01, 0x00, 0xff, 0x01, 0x18, 0x01, 0x18, 0x00, 0x00, 0x00, 0x00

The Cropping Info SEI message contains fields of the following values:

F=0, NRI=0, Type=6(SEI)

payloadType=5 (user data unregistered user SEI), payloadSize=0x1b

uuid_iso_iec_11578=0xbb, 0x7f, 0xc1, 0xa0, 0x69, 0x86, 0x40, 0x52, 0x90, 0xf0, 0x09, 0x29, 0x21, 0x75, 0x39, 0xcf

numOfCropData=1, crop_info_type=0

frame_crop_confidence_level1=255,

frame_crop_left_offset=280(0x118)

frame_crop_right_offset=280(0x118)

frame_crop_top_offset=0

frame_crop_bottom_offset=0

4.3 Bitstream Info SEI

Bitstream Info SEI message bytes in **network byte order**:

0x06, 0x05, 0x12, 0x05, 0xfb, 0xc6, 0xb9, 0x5a, 0x80, 0x40, 0xe5, 0xa2, 0x2a, 0xab, 0x40, 0x20, 0x26, 0x7e, 0x26, 0x00, 0x06

The Bitstream Info SEI message contains fields of the following values:

F=0, NRI=0, Type=6(SEI)

payloadType=5 (user data unregistered user SEI), payloadSize=0x12

uuid_iso_iec_11578=0x05, 0xfb, 0xc6, 0xb9, 0x5a, 0x80, 0x40, 0xe5, 0xa2, 0x2a, 0xab, 0x40, 0x20, 0x26, 0x7e, 0x26

ref_frm_cnt=0, num_of_nal_unit =6

4.4 H.264 Forward Error Correction

H264 FEC Payload Header bytes in **network byte order**:

0x80,0x00,0x00,0x07,0x00,0x00,0x00,0x00,0x03,0x7B

The payload header contains fields of the following values:

E=1, L=0, P=0,X=0,CC=0,M=0,PT Recovery=0,SN offset=7,TS Recovery=0,Length Recovery=37B

Followed by FEC Level Header bytes in network byte order:

0x03, 0x68, 0xFC, 0x00

The FEC Level Header contains fields of the following values:

Protection Length=0x368, Mask=0xFC00

Followed by FEC Level Extension Header bytes in network byte order:

0x00, 0x10

The FEC Level Extension Header contains fields of the following values:

V=0, C=0, HR1=0, HR2=0, Reserved=0, FEC Count=1, FEC Index=0;

Followed by FEC payload:

0x64, 0x05, 0xD5, 0xA8 … (total 872 bytes)

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business 2019
- Microsoft Skype for Business Server 2015
- Microsoft Skype for Business Server 2019
- Windows 10 v1511 operating system
- Windows Server 2016 operating system
- Windows Server 2022 operating system
- Microsoft Skype for Business 2021
- Windows 11 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2: Microsoft Edge supports H264 AVC stream in which case these extensions are not used.

<2> Section 2.2.1: Lync Server 2013, Skype for Business Server 2015, and Skype for Business Server 2019 do not support a zero value of sequence number.

<3> Section 2.2.4: Microsoft Edge supports H264 AVC stream where PACSI are not present.

<4> Section 2.2.4: If start code prefix is removed on the wire after RTP packetization, it needs to be added back after RTP de-packetization before passing a NAL unit to the decoder.

<5> Section 3.1.5.1: Microsoft Edge supports H264 AVC stream where PACSI are not present.

<6> Section 3.1.5.3: Microsoft Edge supports H264 AVC stream in which case PACSI are not present.

<7> Section 3.2.5.1: Microsoft Edge supports H264 AVC stream where PACSI are not present and de-packetization happens without PACSI.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
6 Appendix A: Product Behavior	Updated list of supported products.	Major

8 Index