Stream: Internet Engineering Task Force (IETF)
RFC: 9626
Category: Experimental
Published: February 2025
ISSN: 2070-1721
Authors: M. Zanaty *Cisco Systems* E. Berger *Cisco Systems* S. Nandakumar *Cisco Systems*

RFC 9626
Video Frame Marking RTP Header Extension

Abstract

This document describes a Video Frame Marking RTP header extension used to convey information about video frames that is critical for error recovery and packet forwarding in RTP middleboxes or network nodes. It is most useful when media is encrypted and essential when the middlebox or node has no access to the media decryption keys. It is also useful for codec-agnostic processing of encrypted or unencrypted media, while it also supports extensions for codec-specific information.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9626.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Many widely deployed RTP [RFC3550] topologies [RFC7667] used in modern voice and video conferencing systems include a centralized component that acts as an RTP switch. It receives voice and video streams from each participant, which may be encrypted using Secure Real-time Transport Protocol (SRTP) [RFC3711] or extensions that provide participants with private media [RFC8871] via end-to-end encryption where the switch has no access to media decryption keys. The goal is to provide a set of streams back to the participants, which enable them to render the right media content. For example, in a simple video configuration, the goal will be that each participant sees and hears just the active speaker. In that case, the goal of the switch is to receive the voice and video streams from each participant, determine the active speaker based on energy in the voice packets, possibly using the client-to-mixer audio level RTP header extension [RFC6464], and select the corresponding video stream for transmission to participants; see Figure 1.

In this document, an "RTP switch" is used as shorthand for the terms "switching RTP mixer", "source projecting middlebox", "source forwarding unit/middlebox" and "video switching Multipoint Control Unit (MCU)", as discussed in [RFC7667].

```
+---+      +------------+      +---+
| A |<---->|            |<---->| B |
+---+      |            |      +---+
           |    RTP     |
+---+      |   Switch   |      +---+
| C |<---->|            |<---->| D |
+---+      +------------+      +---+
```

Figure 1: RTP Switch

In order to properly support the switching of video streams, the RTP switch typically needs some critical information about video frames in order to start and stop forwarding streams.

- Because of inter-frame dependencies, it should ideally switch video streams at a point where the first frame from the new speaker can be decoded by recipients without prior frames, e.g., switch on an intra-frame.
- In many cases, the switch may need to drop frames in order to realize congestion control techniques, and it needs to know which frames can be dropped with minimal impact to video quality.
- For scalable streams with dependent layers, the switch may need to selectively forward specific layers to specific recipients due to recipient bandwidth or decoder limits.

Furthermore, it is highly desirable to do this in a payload format-agnostic way that is not specific to each different video codec. Most modern video codecs share common concepts around frame types and other critical information to make this codec-agnostic handling possible.

It is also desirable to be able to do this for SRTP without requiring the video switch to decrypt the packets. SRTP will encrypt the RTP payload format contents; consequently, this data is not usable for the switching function without decryption, which may not even be possible in the case of end-to-end encryption of private media [RFC8871].

By providing meta-information about the RTP streams outside the encrypted media payload, an RTP switch can do codec-agnostic selective forwarding without decrypting the payload. This document specifies the necessary meta-information in an RTP header extension.

2. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Video Frame Marking RTP Header Extension

This specification uses RTP header extensions as defined in [RFC8285]. A subset of meta-information from the video stream is provided as an RTP header extension to allow an RTP switch to do generic selective forwarding of video streams encoded with potentially different video codecs.

The Video Frame Marking RTP header extension is encoded using the one-byte header or two-byte header as described in [RFC8285]. The one-byte header format is used for examples in this document. The two-byte header format is used when other two-byte header extensions are present in the same RTP packet since mixing one-byte and two-byte extensions is not possible in the same RTP packet.

This extension is only specified for Source (not Redundancy) RTP Streams [RFC7656] that carry video payloads. It is not specified for audio payloads, nor is it specified for Redundancy RTP Streams. The (separate) specifications for Redundancy RTP Streams often include provisions for recovering any header extensions that were part of the original source packet. Such provisions can be followed to recover the Video Frame Marking RTP header extension of the original source packet. Source packet frame markings may be useful when generating Redundancy RTP Streams; for example, the I (Independent Frame) and D (Discardable Frame) bits, defined in Section 3.1, can be used to generate extra or no redundancy, respectively, and redundancy schemes with source blocks can align source block boundaries with independent frame boundaries as marked by the I bit.

A frame, in the context of this specification, is the set of RTP packets with the same RTP timestamp from a specific RTP Synchronization Source (SSRC). A frame within a layer is the set of RTP packets with the same RTP timestamp, SSRC, Temporal-layer ID (TID), and Layer ID (LID).

3.1. Long Extension for Scalable Streams

The following RTP header extension is **RECOMMENDED** for scalable streams. It **MAY** also be used for non-scalable streams, in which case the TID, LID, and TL0PICIDX **MUST** be 0 or omitted. The ID is assigned per [RFC8285]. The length is encoded as follows:

- L=2 to indicate 3 octets of data when nothing is omitted,
- L=1 for 2 octets when TL0PICIDX is omitted, or
- L=0 for 1 octet when both the LID and TL0PICIDX are omitted.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |   LID         |    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           or
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=1  |S|E|I|D|B| TID |   LID         | (TL0PICIDX omitted)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           or
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=0  |S|E|I|D|B| TID | (LID and TL0PICIDX omitted)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following information is extracted from the media payload and sent in the Video Frame Marking RTP header extension.

S: Start of Frame (1 bit)
: **MUST** be 1 in the first packet in a frame within a layer; otherwise, **MUST** be 0.

E: End of Frame (1 bit)
: **MUST** be 1 in the last packet in a frame within a layer; otherwise, **MUST** be 0. Note that the RTP header marker bit **MAY** be used to infer the last packet of the highest enhancement layer in payload formats with such semantics.

I: Independent Frame (1 bit)
: **MUST** be 1 for a frame within a layer that can be decoded independent of temporally prior frames, e.g., intra-frame, VPX keyframe, H.264 Instantaneous Decoding Refresh (IDR) [RFC6184], or H.265 IDR / Clean Random Access (CRA) / Broken Link Access (BLA) / Random Access Point (RAP) [RFC7798]; otherwise, **MUST** be 0. Note that this bit only signals temporal independence, so it can be 1 in spatial or quality enhancement layers that depend on temporally co-located layers but not temporally prior frames.

D: Discardable Frame (1 bit)
: **MUST** be 1 for a frame within a layer the sender knows can be discarded and still provide a decodable media stream; otherwise, **MUST** be 0.

B: Base Layer Sync (1 bit)
When the TID is not 0, this **MUST** be 1 if the sender knows this frame within a layer only depends on the base temporal layer; otherwise, **MUST** be 0. When the TID is 0 or if no scalability is used, this **MUST** be 0.

TID: Temporal-layer ID (3 bits)
Identifies the temporal layer/sub-layer encoded, starting with 0 for the base layer and increasing with higher temporal fidelity. If no scalability is used, this **MUST** be 0. It is implicitly 0 in the short extension format.

LID: Layer ID (8 bits)
Identifies the spatial and quality layer encoded, starting with 0 for the base layer and increasing with higher fidelity. If no scalability is used, this **MUST** be 0 or omitted to reduce length. When the LID is omitted, TL0PICIDX **MUST** also be omitted. It is implicitly 0 in the short extension format or when omitted in the long extension format.

TL0PICIDX: Temporal Layer 0 Picture Index (8 bits)
When the TID is 0 and the LID is 0, this is a cyclic counter labeling base layer frames. When the TID is not 0 or the LID is not 0, the indication is that a dependency on the given index, such that this frame within this layer depends on the frame with this label in the layer with a TID 0 and LID 0. If no scalability is used, or the cyclic counter is unknown, TL0PICIDX **MUST** be omitted to reduce length. Note that 0 is a valid index value for TL0PICIDX.

The layer information contained in the TID and LID convey useful aspects of the layer structure that can be utilized in selective forwarding.

Without further information about the layer structure, these TID/LID identifiers can only be used for relative priority of layers and implicit dependencies between layers. They convey a layer hierarchy with TID = 0 and LID = 0 identifying the base layer. Higher values of TID identify higher temporal layers with higher frame rates. Higher values of LID identify higher spatial and/or quality layers with higher resolutions and/or bitrates. Implicit dependencies between layers assume that a layer with a given TID/LID **MAY** depend on a layer or layers with the same or lower TID/LID, but they **MUST NOT** depend on a layer or layers with higher TID/LID.

With further information, for example, possible future RTCP source description (SDES) items that convey full layer structure information, it may be possible to map these TIDs and LIDs to specific absolute frame rates, resolutions, bitrates, and explicit dependencies between layers. Such additional layer information may be useful for forwarding decisions in the RTP switch but is beyond the scope of this document. The relative layer information is still useful for many selective forwarding decisions, even without such additional layer information.

3.2. Short Extension for Non-Scalable Streams

The following RTP header extension is **RECOMMENDED** for non-scalable streams. It is identical to the shortest form of the extension for scalable streams, except the last four bits (B and TID) are replaced with zeros. It **MAY** also be used for scalable streams if the sender has limited or no information about stream scalability. The ID is assigned per [RFC8285]; the length is encoded as L=0, which indicates 1 octet of data.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=0  |S|E|I|D|0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following information is extracted from the media payload and sent in the Video Frame Marking RTP header extension.

S: Start of Frame (1 bit)
: **MUST** be 1 in the first packet in a frame; otherwise, **MUST** be 0.

E: End of Frame (1 bit)
: **MUST** be 1 in the last packet in a frame; otherwise, **MUST** be 0. **SHOULD** match the RTP header marker bit in payload formats with such semantics for marking end of frame.

I: Independent Frame (1 bit)
: **MUST** be 1 for frames that can be decoded independent of temporally prior frames, e.g., intra-frame, VPX keyframe, H.264 IDR [RFC6184], or H.265 IDR/CRA/BLA/IRAP [RFC7798]; otherwise, **MUST** be 0.

D: Discardable Frame (1 bit)
: **MUST** be 1 for frames the sender knows can be discarded and still provide a decodable media stream; otherwise, **MUST** be 0.

The remaining (4 bits)
: These are reserved/fixed values and not used for non-scalable streams; they **MUST** be set to zero upon transmission and ignored upon reception.

3.3. LID Mappings for Scalable Streams

This section maps the specific Layer ID (LID) information contained in specific scalable codecs to the generic LID and TID fields.

Note that non-scalable streams have no LID information; thus, they have no mappings.

3.3.1. VP9 LID Mapping

The VP9 [RFC9628] Spatial-layer ID (SID, 3 bits) and Temporal-layer ID (TID, 3 bits) in the VP9 payload descriptor are mapped to the generic LID and TID fields in the header extension as shown in the following figure.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |0|0|0|0|0| SID |    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The S bit **MUST** match the B bit in the VP9 payload descriptor.

The E bit **MUST** match the E bit in the VP9 payload descriptor.

The I bit **MUST** match the inverse of the P bit in the VP9 payload descriptor.

The D bit **MUST** be 1 if the refresh_frame_flags bits in the VP9 payload uncompressed header are all 0; otherwise, it **MUST** be 0.

The B bit **MUST** be 0 if the TID is 0; if the TID is not 0, it **MUST** match the U bit in the VP9 payload descriptor.

> Note: when using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP9 U bit will always be 1 if the TID is not 0 since it is always possible to switch up to a higher temporal layer in such nested structures.

The TID, SID, and TL0PICIDX **MUST** match the correspondingly named fields in the VP9 payload descriptor, with SID aligned in the least significant 3 bits of the 8-bit LID field and zeros in the most significant 5 bits.

3.3.2. H265 LID Mapping

The H265 [RFC7798] layer ID (6 bits), and TID (3 bits) from the Network Abstraction Layer (NAL) unit header are mapped to the generic LID and TID fields in the header extension as shown in the following figure.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |0|0| layer ID  |    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The S and E bits **MUST** match the correspondingly named bits in PACI:PHES:TSCI payload structures.

The I bit **MUST** be 1 when the NAL unit type is 16-23 (inclusive) or 32-34 (inclusive), or an aggregation packet or fragmentation unit encapsulating any of these types; otherwise, it **MUST** be 0. These ranges cover intra (IRAP) frames as well as critical parameter sets (Video Parameter Set (VPS), Sequence Parameter Set (SPS), Picture Parameter Set (PPS)).

The D bit **MUST** be 1 if either:

- the payload's NAL unit header's NRI field is 0, or
- the payload is an aggregation packet or fragmentation unit encapsulating only NAL units with NRI = 0.

Otherwise, it **MUST** be 0.

The NRI = 0 condition signals non-reference frames.

The B bit cannot be determined reliably from simple inspection of payload headers; therefore, it is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

The TID and layer ID **MUST** match the correspondingly named fields in the H265 NAL unit header, with layer ID aligned in the least significant 6 bits of the 8-bit LID field and zeros in the most significant 2 bits.

3.3.3. H264 Scalable Video Coding (SVC) LID Mapping

The following shows H264-SVC [RFC6190] Layer encoding information (3 bits for spatial/dependency layer (DID), 4 bits for quality layer (QID), and 3 bits for temporal layer) mapped to the generic LID and TID fields.

The S, E, I, and D bits **MUST** match the correspondingly named bits in Payload Content Scalability Information (PACSI) payload structures.

The I bit **MUST** be 1 when the NAL unit type is 5, 7, 8, 13, 15, or an aggregation packet or fragmentation unit encapsulating any of these types; otherwise, it **MUST** be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS variants).

The D bit **MUST** be 1 if either:

- the payload's NAL unit header's NRI field is 0, or
- the payload is an aggregation packet or fragmentation unit encapsulating only NAL units with NRI = 0.

Otherwise, it **MUST** be 0.

The NRI = 0 condition signals non-reference frames.

The B bit cannot be determined reliably from simple inspection of payload headers; therefore, it is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |0| DID |  QID  |    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3.4. H264 Advanced Video Coding (AVC) LID Mapping

The following shows the header extension for H264 (AVC) [RFC6184] that contains only temporal layer information.

The S bit **MUST** be 1 when the timestamp in the RTP header differs from the timestamp in the prior RTP sequence number from the same SSRC; otherwise, it **MUST** be 0.

The E bit **MUST** match the M bit in the RTP header.

The I bit **MUST** be 1 when the NAL unit type is 5, 7, or 8, or an aggregation packet or fragmentation unit encapsulating any of these types; otherwise, it **MUST** be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS).

The D bit **MUST** be 1 if either:

- the payload's NAL unit header's NRI field is 0, or
- the payload is an aggregation packet or fragmentation unit encapsulating only NAL units with NRI = 0.

Otherwise, it **MUST** be 0.

The NRI = 0 condition signals non-reference frames.

The B bit cannot be determined reliably from simple inspection of payload headers; therefore, it is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |0|0|0|0|0|0|0|0|    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3.5. VP8 LID Mapping

The following shows the header extension for VP8 [RFC7741] that contains only temporal layer information.

The S bit **MUST** match the correspondingly named bit in the VP8 payload descriptor when PID=0; otherwise, it **MUST** be 0.

The E bit **MUST** match the M bit in the RTP header.

The I bit **MUST** match the inverse of the P bit in the VP8 payload header.

The D bit **MUST** match the N bit in the VP8 payload descriptor.

The B bit **MUST** match the Y bit in the VP8 payload descriptor.

> Note: when using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP8 Y bit will always be 1 if the TID is not 0 since it is always possible to switch up to a higher temporal layer in such nested structures.

The TID and TL0PICIDX **MUST** match the correspondingly named fields in the VP8 payload descriptor.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID=? |  L=2  |S|E|I|D|B| TID |0|0|0|0|0|0|0|0|    TL0PICIDX  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3.6. Future Codec LID Mapping

The RTP payload format specification for future video codecs **SHOULD** include a section describing the LID mapping and TID mapping for the codec.

3.4. Signaling Information

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdrext:framemarking". It does not contain any extension attributes.

An example attribute line in SDP:

```
    a=extmap:3 urn:ietf:params:rtp-hdrext:framemarking
```

3.5. Usage Considerations

The header extension values **MUST** represent what is already in the RTP payload.

When an RTP switch needs to discard received video frames due to congestion control considerations, it is **RECOMMENDED** that it drop:

- frames marked with the D bit set, or
- frames with the highest values of TID and LID (which indicate the highest temporal and spatial/quality enhancement layers) since those typically have fewer dependencies on them than lower layers.

When an RTP switch wants to forward a new video stream to a receiver, it is **RECOMMENDED** to select the new video stream from the first switching point with the I bit set in all spatial layers and forward the video stream from that point on. An RTP switch can request that a media source generate a switching point by sending an RTCP Full Intra Request (FIR) as defined in [RFC5104], for example.

3.5.1. Relation to Layer Refresh Request (LRR)

Receivers can use the Layer Refresh Request (LRR) [RFC9627] RTCP feedback message to upgrade to a higher layer in scalable encodings. The TID/LID values and formats used in LRR messages **MUST** correspond to the same values and formats specified in Section 3.1.

Because frame marking can only be used with temporally nested streams, temporal-layer refreshes requested with an LRR message are unnecessary for frame-marked streams. Other refreshes can be detected based on the I bit being set for the specific spatial layers.

3.5.2. Scalability Structures

The LID and TID information is most useful for fixed scalability structures, such as nested hierarchical temporal layering structures, where each temporal layer only references lower temporal layers or the base temporal layer. The LID and TID information is less useful, or even not useful at all, for complex, irregular scalability structures that do not conform to common, fixed patterns of inter-layer dependencies and referencing structures. Therefore, it is **RECOMMENDED** to use LID and TID information for RTP switch forwarding decisions only in the case of temporally nested scalability structures, and it is **NOT RECOMMENDED** for other (more complex or irregular) scalability structures.

4. Security and Privacy Considerations

In "The Secure Real-time Transport Protocol (SRTP)" [RFC3711], RTP header extensions are authenticated and optionally encrypted [RFC9335]. When unencrypted header extensions are used, some metadata is exposed and visible to middleboxes on the network path, while encrypted media data and metadata in encrypted header extensions are not exposed.

The primary utility of this specification is for RTP switches to make proper media forwarding decisions. RTP switches are the SRTP peers of endpoints, so they can access encrypted header extensions, but not end-to-end encrypted private media payloads. Other middleboxes on the network path can only access unencrypted header extensions since they are not SRTP peers.

RTP endpoints that negotiate this extension should consider whether:

- this video frame marking metadata needs to be exposed to the SRTP peer only, in which case the header extension can be encrypted; or
- other middleboxes on the network path also need this metadata, for example, to optimize packet drop decisions that minimize media quality impacts, in which case the header extension can be unencrypted, if the endpoint accepts the potential privacy leakage of this metadata.

For example, it would be possible to determine keyframes and their frequency in unencrypted header extensions. This information can often be obtained via statistical analysis of encrypted data. For example, keyframes are usually much larger than other frames, so frame size alone can leak this in the absence of any unencrypted metadata. However, unencrypted metadata provides a reliable signal rather than a statistical probability; so endpoints should take that into consideration to balance the privacy leakage risk against the potential benefit of optimized media delivery when deciding whether to negotiate and encrypt this header extension.

5. IANA Considerations

This document defines a new extension URI listed in the "RTP Compact Header Extensions" subregistry of the "Real-Time Transport Protocol (RTP) Parameters" registry, according to the following data:

Extension URI: urn:ietf:params:rtp-hdrext:framemarking

Description: Frame marking information for video streams

Contact: mzanaty@cisco.com

Reference: RFC 9626

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <https://www.rfc-editor.org/info/rfc6184>.

[RFC6190] Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <https://www.rfc-editor.org/info/rfc6190>.

[RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <https://www.rfc-editor.org/info/rfc7741>.

[RFC7798] Wang, Y.-K., Sanchez, Y., Schierl, T., Wenger, S., and M. M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <https://www.rfc-editor.org/info/rfc7798>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <https://www.rfc-editor.org/info/rfc8285>.

6.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <https://www.rfc-editor.org/info/rfc5104>.

[RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <https://www.rfc-editor.org/info/rfc6464>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

[RFC8871] Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy-Enhanced RTP Conferencing (PERC)", RFC 8871, DOI 10.17487/RFC8871, January 2021, <https://www.rfc-editor.org/info/rfc8871>.

[RFC9335] Uberti, J., Jennings, C., and S. Murillo, "Completely Encrypting RTP Header Extensions and Contributing Sources", RFC 9335, DOI 10.17487/RFC9335, January 2023, <https://www.rfc-editor.org/info/rfc9335>.

[RFC9627] Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", RFC 9627, DOI 10.17487/RFC9627, February 2025, <https://www.rfc-editor.org/info/rfc9627>.

[RFC9628] Uberti, J., Holmer, S., Flodman, M., Hong, D., and J. Lennox, "RTP Payload Format for VP9 Video", RFC 9628, DOI 10.17487/RFC9628, February 2025, <https://www.rfc-editor.org/info/rfc9628>.

Acknowledgements

Many thanks to Bernard Aboba, Jonathan Lennox, Stephan Wenger, Dale Worley, and Magnus Westerlund for their inputs.

Authors' Addresses

Mo Zanaty
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States of America
Email: mzanaty@cisco.com

Espen Berger
Cisco Systems
Email: espeberg@cisco.com

Suhas Nandakumar
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States of America
Email: snandaku@cisco.com