

---

Stream: Internet Engineering Task Force (IETF)  
RFC: [9993](#)  
Updates: [9695](#)  
Category: Standards Track  
Published: May 2026  
ISSN: 2070-1721  
Authors: HS Yang X. de Foy  
*InterDigital InterDigital*

# RFC 9993

## RTP Payload Format for Haptics

---

### Abstract

This memo specifies an RTP payload format for MPEG-I haptic data. A haptic media stream is composed of MPEG-I Haptic Stream (MIHS) units including a MIHS unit header and zero or more MIHS packets. The RTP payload header format allows for packetization of a MIHS unit in an RTP packet payload as well as fragmentation of a MIHS unit into multiple RTP packets. The original subtype registration for 'haptics/hmpg' (RFC 9695) did not include any required or optional parameters. This memo updates RFC 9695 and the 'haptics/hmpg' registration to add optional parameters. It also provides Session Description Protocol (SDP) usage information for the 'haptics' media type.

### Status of This Memo

This is an Internet Standards Track document.

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). Further information on Internet Standards is available in 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/rfc9993>.

### Copyright Notice

Copyright (c) 2026 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	3
2. Conventions	3
3. Definitions	3
4. Haptic Format Description	5
4.1. Overview of Haptic Coding	5
4.2. MIHS Format	5
5. Payload Format for Haptics	6
5.1. RTP Header Usage	6
5.2. Payload Header	6
5.3. Payload Structures	7
5.3.1. Single Unit Payload Structure	8
5.3.2. Fragmented Unit Payload Structure	9
5.3.3. Aggregation Packet Payload Structure	10
5.4. MIHS Units Transmission and Reception Considerations	12
6. Payload Format Parameters	13
6.1. Optional Parameters Definition	13
6.2. SDP Parameter Registration	14
7. SDP Considerations	15
7.1. SDP Offer/Answer Considerations	15
7.2. Declarative SDP Considerations	17
8. Congestion Control Considerations	17
9. Security Considerations	18
10. IANA Considerations	19
10.1. Media Type Registration Update	19
10.2. New SDP Parameters Media Registration	20

---

11. References	20
11.1. Normative References	20
11.2. Informative References	20
Acknowledgments	21
Authors' Addresses	22

## 1. Introduction

Haptics provides users with tactile effects in addition to audio and video, allowing them to experience sensory immersion. Haptic data is mainly transmitted to devices that act as actuators, providing them with information to operate according to the values defined in haptic effects. The IETF registered 'haptics' as a primary media type, akin to 'audio' and 'video' [[RFC9695](#)].

The MPEG Haptics Coding standard [[ISO.IEC.23090-31](#)] defines the data formats, metadata, and codec architecture to encode, decode, synthesize, and transmit haptic signals. Within this MPEG standard, a haptic media stream is composed of MIHS units including a MIHS unit header and zero or more MIHS packets. The MIHS unit is a unit of packetization suitable for streaming and is similar in essence to the Network Abstraction Layer (NAL) unit defined in some video specifications. This document specifies how haptic data (MIHS units) can be transmitted using the RTP protocol. This document follows recommendations in [[RFC8088](#)] and [[RFC2736](#)] for RTP payload format writers. This document does not specify synchronization (lip sync) mechanisms between haptics and audio/video components. In addition, this document specifies the associated SDP parameters and SDP offer/answer considerations for the 'haptics' media type.

## 2. Conventions

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. Definitions

This document uses the definitions of the MPEG Haptics Coding standard [[ISO.IEC.23090-31](#)]. Some of these terms are provided here for convenience.

Actuator: Component of a device for rendering haptic sensations.

Avatar: Body (or part of body) representation.

**Band:** Component in a channel for containing effects for a specific range of frequencies.

**Channel:** Component in a perception containing one or more bands rendered on a device at a specific body location.

**Device:** Physical system having one or more actuators configured to render a haptic sensation corresponding with a given signal.

**Effect:** Component of a band for defining a signal, consisting of a haptic waveform or one or more haptic keyframes.

**Experience:** Top-level haptic component containing perceptions and metadata.

**Haptics:** Tactile sensations.

**Keyframe:** Component of an effect mapping a position in time or space to an effect parameter such as amplitude or frequency.

**Metadata:** Global information about an experience, perception, channel, or band.

**MIHS unit:** Unit of packetization of the MPEG-I Haptic Stream format, which is used as unit of payload in the format described in this memo. See [Section 4](#) for details.

**Modality:** Type of haptics, such as vibration, force, pressure, position, velocity, or temperature.

**Perception:** Haptic perception containing channels of a specific modality.

**Signal:** Representation of the haptics associated with a specific modality to be rendered on a device.

**Hmpg format:** A binary compressed format for haptics data. Information is stored in a binary form, and data compression is applied on data at the band level. The 'haptics/hmpg' media subtype is registered in [\[RFC9695\]](#) and updated by this memo.

**Independent unit:** A MIHS unit is independent if it can be decoded independently from earlier units. Independent units contain timing information and are also called "sync units" in [\[ISO.IEC.23090-31\]](#).

**Dependent unit:** A MIHS unit is dependent if it requires earlier units for decoding. Dependent units do not contain timing information and are also called "non-sync units" in [\[ISO.IEC.23090-31\]](#).

**Time-independent effect:** A haptic effect that occurs regardless of time. The tactile feedback of a texture is a representative example. Time-independent effects are encoded in spatial MIHS units, as defined in [Section 4.2](#).

**Time-dependent effect:** A haptic effect that varies over time. For example, tactile feedback for vibration and force are time-dependent effects and are encoded in temporal MIHS units, as defined in [Section 4.2](#).

## 4. Haptic Format Description

### 4.1. Overview of Haptic Coding

The MPEG Haptics Coding standard specifies methods for efficient transmission and the rendering of haptic signals, to enable immersive experiences. It supports multiple types of perceptions, including the most common vibrotactile (sense of touch that perceives vibrations) and kinesthetic perceptions (tactile resistance or force), and also other less common perceptions, such as the sense of temperature or texture, for example. It also supports two approaches for encoding haptic signals: a "quantized" approach based on samples of measured data and a "descriptive" approach where the signal is synthesized using a combination of functions. Both quantized and descriptive data can be encoded in a text-based exchange format based on JSON (.hjif) or in a binary packetized format for distribution and streaming (.hmpg). This last format is referred to as the MIHS format and is a base for the RTP payload format described in this document.

### 4.2. MIHS Format

MIHS is a stream format used to transport haptic data. Haptic data, including haptic effects, is packetized according to the MIHS format and delivered to actuators, which operate according to the provided effects. The MIHS format has two levels of packetization: MIHS units and MIHS packets.

MIHS units are composed of a MIHS unit header and zero or more MIHS packets. Four types of MIHS units are defined. An initialization MIHS unit contains MIHS packets carrying metadata necessary to reset and initialize a haptic decoder, including a timestamp. A temporal MIHS unit contains one or more MIHS packets defining time-dependent effects and provides modalities such as pressure, velocity, and acceleration. The duration of a temporal unit is a positive number. A spatial MIHS unit contains one or more MIHS packets providing time-independent effects, such as vibrotactile texture, stiffness, and friction. The duration of a spatial unit is always zero. A silent MIHS unit indicates that there is no effect during a time interval, and its duration is a positive number.

A MIHS unit can be marked as independent or dependent. When a decoder processes an independent unit, it resets the previous effects and therefore provides a haptic experience independent from any previous MIHS unit. A dependent unit is the continuation of previous MIHS units and cannot be independently decoded and rendered without having decoded a previous MIHS unit(s). Initialization and spatial MIHS units are always independent units. Temporal and silent MIHS units can be dependent or independent units.

[Figure 1](#) illustrates a succession of MIHS units in a MIHS stream.

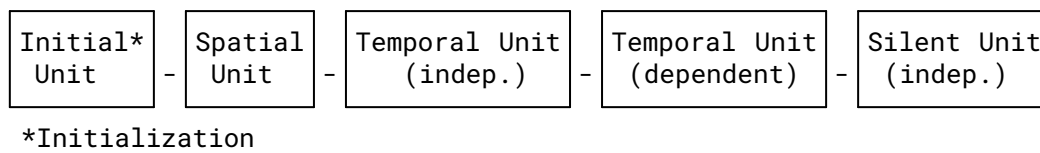


Figure 1: Example of MIHS Stream

## 5. Payload Format for Haptics

### 5.1. RTP Header Usage

The RTP header is defined in [RFC3550] and represented in Figure 2. Unless contextualized below, the meaning of the fields depicted in Figure 2 is the same as in Section 5.1 of [RFC3550].

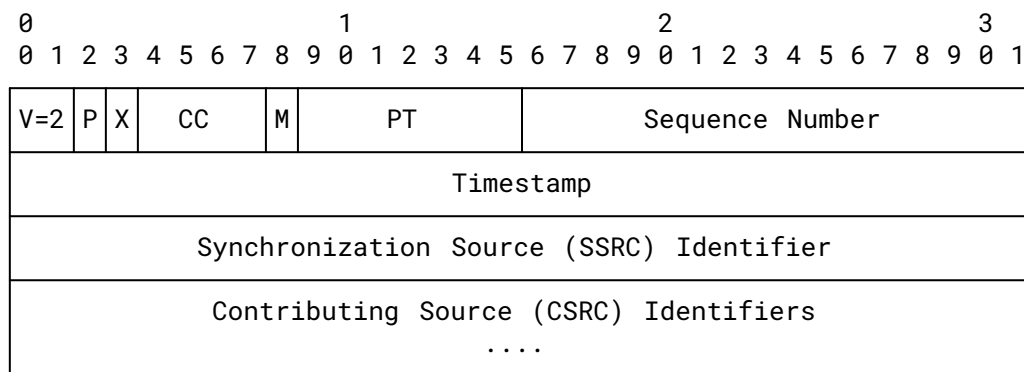


Figure 2: RTP Header for Haptic

**Marker bit (M):** 1 bit. The marker bit **SHOULD** be set to one in the first non-silent RTP packet after a period of haptic silence. This enables jitter buffer adaptation and haptics device washout (i.e., reset to a neutral position) prior to the beginning of the burst with minimal impact on the quality of experience for the end user. The marker bit in all other packets **MUST** be set to zero.

**Timestamp (TS):** 32 bits. A timestamp representing the sampling time of the first sample of the MIHS unit in the RTP payload. The clock frequency **MUST** be set to the sample rate of the encoded haptic data and is conveyed out of band (e.g., as an SDP parameter).

### 5.2. Payload Header

The RTP payload header follows the RTP header. Figure 3 describes the RTP payload header for Haptic.

0	1	2	3	4	5	6	7
D	UT			L			

Figure 3: RTP Payload Header for Haptic

D (Dependency, 1 bit): This field indicates whether the MIHS unit included in the RTP payload is dependent (when its value is one) or independent (when its value is zero).

UT (Unit Type, 3 bits): This field indicates the type of the MIHS unit included in the RTP payload. UT field values are listed in [Table 1](#).

L (MIHS Layer, 4 bits): This field is an integer value that indicates the priority order of the MIHS unit included in the RTP payload, as determined by the haptic sender (e.g., by the haptic codec), based on application-specific needs. For example, the sender may use the MIHS layer to prioritize perceptions with the largest impact on the end-user experience. Zero corresponds to the highest priority. The semantic of individual MIHS layers are not specified and are left for the application to assign. In cases where the sender does not use the L field to indicate the priority order of the MIHS unit, the L value is '0'.

### 5.3. Payload Structures

Three different types of RTP packet payload structures are specified. A single unit packet contains a single MIHS unit in the payload. A fragmentation unit contains a subset of a MIHS unit. An aggregation packet contains multiple MIHS units in the payload. The unit type (UT) field of the RTP payload header, as shown in [Table 1](#), identifies both the payload structure and, in the case of a single-unit structure, the type of MIHS unit present in the payload.

Unit Type	Payload Structure	Packet Type Name
0	N/A	Unassigned
1	Single	Initialization MIHS Unit
2	Single	Temporal MIHS Unit
3	Single	Spatial MIHS Unit
4	Single	Silent MIHS Unit
5	Aggr	Single-Time Aggregation Packet (STAP)
6	Aggr	Multi-Time Aggregation Packet (MTAP)

Unit Type	Payload Structure	Packet Type Name
7	Frag	Fragmentation Unit

Table 1: Payload Structure Type for Haptic

The payload structures are represented in Figure 4. The single unit payload structure is specified in Section 5.3.1. The fragmented unit payload structure is specified in Section 5.3.2. The aggregation packet payload structure is specified in Section 5.3.3. The padding in the figures of these sections refers to the RTP padding defined in [RFC3550].

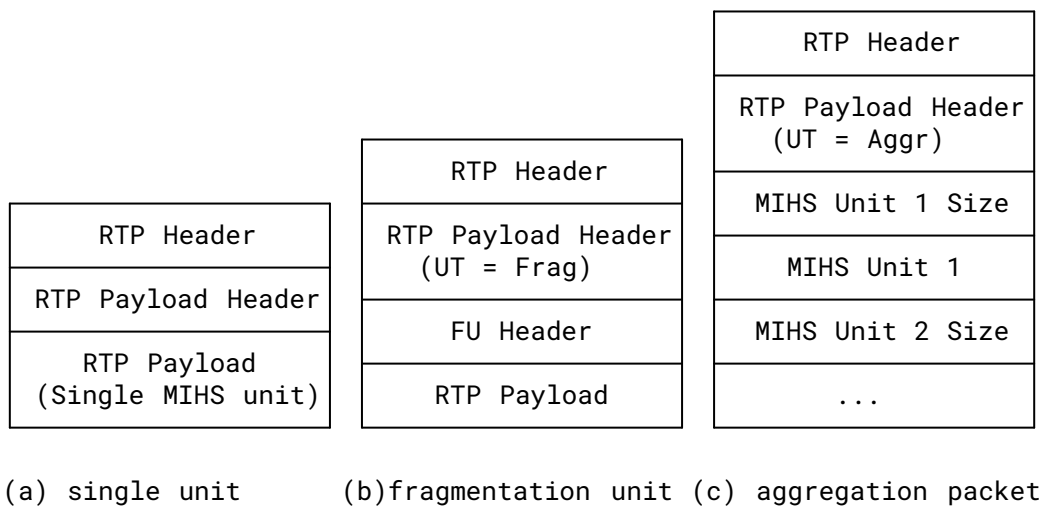


Figure 4: RTP Transmission Modes

### 5.3.1. Single Unit Payload Structure

In a single unit payload structure, as described in Figure 5, the RTP packet contains the RTP header, followed by the payload header and one single MIHS unit. The payload header follows the structure described in Section 5.2. The payload contains a MIHS unit as defined in [ISO.IEC.23090-31].

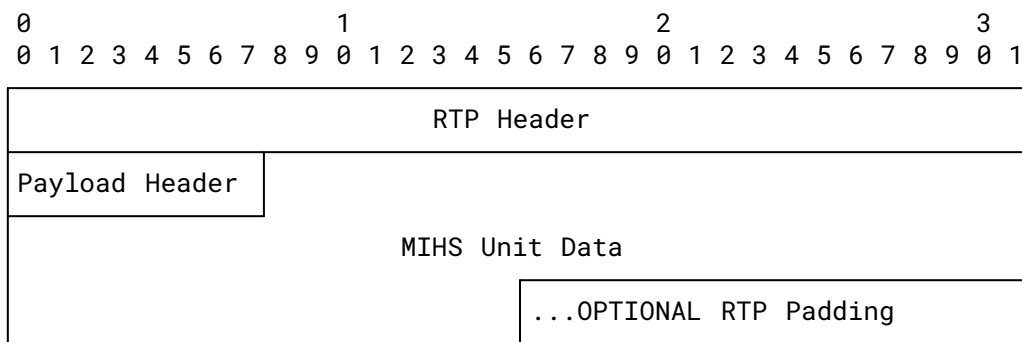


Figure 5: Single Unit Payload Structure

### 5.3.2. Fragmented Unit Payload Structure

In a fragmented unit payload structure, as described in [Figure 6](#), the RTP packet contains the RTP header, followed by the payload header, a Fragmented Unit (FU) header, and a MIHS unit fragment. The payload header follows the structure described in [Section 5.2](#). The value of the UT field of the payload header is 7. The FU header follows the structure described in [Figure 7](#). In the case of fragmentation, all RTP payload header fields **MUST** remain unchanged across all fragments.

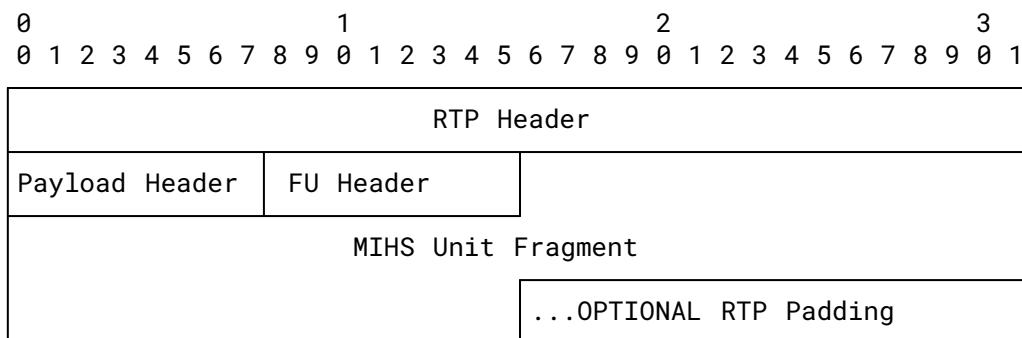


Figure 6: Fragmentation Unit Payload Structure

FU headers are used to enable fragmenting a single MIHS unit into multiple RTP packets. Fragments of the same MIHS unit **MUST** be sent in consecutive order with ascending RTP sequence numbers (with no other RTP packets within the same RTP stream being sent between the first and last fragment). FUs **MUST NOT** be nested, i.e., an FU **MUST NOT** contain a subset of another FU.

[Figure 7](#) describes an FU header, including the following fields:

0	1	2	3	4	5	6	7
FUS	FUE	RSV			UT		

Figure 7: Fragmentation Unit Header

FUS (Fragmented Unit Start, 1 bit): This field **MUST** be set to 1 for the first fragment and 0 for the other fragments.

FUE (Fragmented Unit End, 1 bit): This field **MUST** be set to 1 for the last fragment and 0 for the other fragments.

The combination FUS=1 and FUE=1 **MUST NOT** occur; such packets are invalid.

RSV (Reserved, 3 bits): These bits **MUST** be set to 0 by the sender and ignored by the receiver.

UT (Unit Type, 3 bits): This field indicates the type of the MIHS unit this fragment belongs to, using values defined in [Table 1](#).

The use of MIHS unit fragmentation in RTP means that a media receiver can receive some fragments, but not other fragments. The missing fragments will typically not be retransmitted by RTP. This results in partially received MIHS units, which can be either dropped or used by the decoding application, based on implementation. In cases where consecutive fragments with FUE and FUS are lost, the receiver may be able to detect that surrounding fragments belong to a different partially received MIHS unit (e.g., if the UT field holds a different value).

### 5.3.3. Aggregation Packet Payload Structure

In an aggregation packet, as described in [Figure 8](#), the RTP packet contains an RTP header, followed by a payload header, and (for each aggregated MIHS unit) a MIHS unit size followed by the MIHS unit. The payload header follows the structure described in [Section 5.2](#).

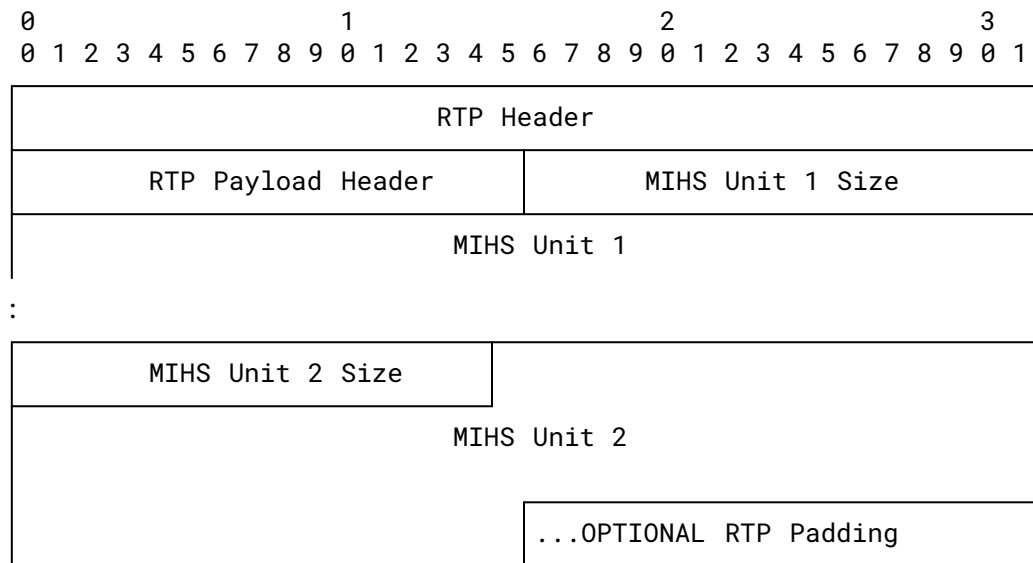


Figure 8: Single-Time Aggregation Packet

Figure 8 shows a Single-Time Aggregation Packet (STAP), which can be used to transmit multiple MIHS units that correspond to the same timestamp. For example, if two frequencies are used for the same content, they can be transmitted at once in a STAP. Multiple spatial units can also be sent together in a STAP, since this type of haptics data is time independent. The MIHS unit length field (16 bits) holds the length of the MIHS unit following it, in bytes. The value of the UT field of the payload header is 5.

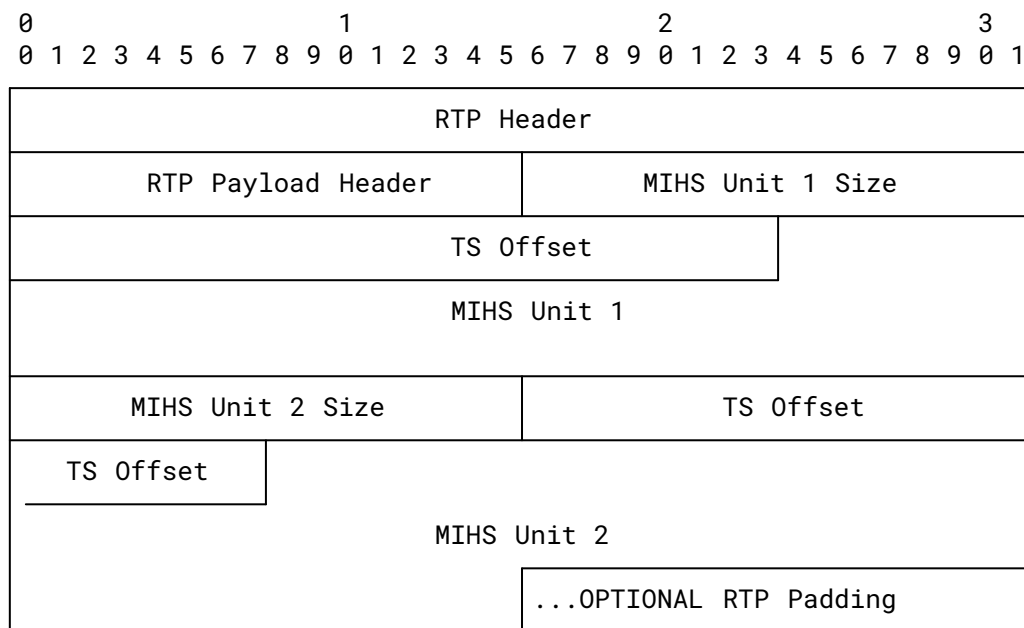


Figure 9: Multiple-Time Aggregation Packet

Figure 9 shows a Multi-Time Aggregation Packet (MTAP). It is used to transmit multiple MIHS units with different timestamps, in one RTP packet. Multi-time aggregation can help reduce the number of packets in environments where some delay is acceptable. The value of the UT field of the payload header is 6. The MIHS unit length field (16 bits) holds the length of the MIHS unit following it, in bytes. The timestamp offset field (TS offset, 16 bits) is present in the MTAP case and **MUST** be set to the value of (time of the MIHS unit - RTP timestamp of the packet). The timestamp offset of the earliest aggregation unit **MUST** always be zero. Therefore, the RTP timestamp of the MTAP is identical to the earliest MIHS unit time.

#### 5.4. MIHS Units Transmission and Reception Considerations

The following considerations apply for the streaming of MIHS units over RTP.

The MIHS format enables variable duration units and uses initialization MIHS units to declare the duration of subsequent non-zero duration MIHS units, as well as the maximum variation of this duration. A sender **SHOULD** set constant or low-variability (e.g., lower than the playout buffer) durations in initialization MIHS units, for RTP streaming. This enables the receiver to determine early (e.g., using a timer) when a unit has been lost and to make the decoder more robust to RTP packet loss. If a sender sends MIHS units with high duration variations, the receiver **MAY** need to wait for a long period of time (e.g., the upper bound of the duration variation) to determine if a MIHS unit was lost in transmission. Whether this behavior is acceptable or not is application dependent, and the application can configure the encoder to generate MIHS unit of lengths with the appropriate variation.

The MIHS format uses silent MIHS units to signal haptic silence. A sender **MAY** decide not to send silent units, to save network resources. Since, from a receiver standpoint, a missed MIHS unit may originate from a not-sent silent unit or a lost packet, a sender **MAY** send one, or a few, MIHS silent units at the beginning of a haptic silence. If a media receiver receives a MIHS silent unit, the receiver **SHOULD** assume that silence is intended until the reception of a non-silent MIHS unit. This can reduce the number of false detections of lost RTP packets by the decoder.

In some multimedia conference scenarios using an RTP video mixer (e.g., when adding or selecting a new source), it is recommended to use Full Intra Request (FIR) feedback messages [RFC5104] with Haptics. The purpose of the FIR message is to cause an encoder to send a decoder refresh point at the earliest opportunity. In the context of haptics, an appropriate decoder refresh point is an initialization MIHS unit. The initialization MIHS unit point enables a decoder to be reset to a known state and to decode all MIHS units following it.

## 6. Payload Format Parameters

This section describes payload format parameters. Section 6.1 specifies new optional parameters, and Section 6.2 further registers a new token in the media subregistry of the "Session Description Protocol (SDP) Parameters" registry group.

### 6.1. Optional Parameters Definition

It is optional to include the SDP parameters in this section. Some parameters have a default value that **MUST** be inferred if the parameter is not present in the SDP, unless an out-of-band agreement indicates a different value, as described in Section 7.1. The values of the SDP parameters indicated in this section are based on the current version of the MPEG Haptics Coding standard (ISO/IEC 23090-31:2025) and may be different in future versions of [ISO.IEC.23090-31].

ver:

This parameter provides the year of the edition and amendment of ISO/IEC 23090-31 that this file conforms to, as defined in [ISO.IEC.23090-31]: MPEG\_haptics object.version is a string that may hold values such as XXXX or XXXX-Y where XXXX is the year of publication and Y is the amendment number, if any. For the initial (and current) version of the MPEG Haptics Coding standard (ISO/IEC 23090-31:2025), the value is "2025". When ver is not present, a default value of "2025" **SHOULD** be inferred.

profile: This parameter indicates the profile used to generate the encoded stream, as defined in [ISO.IEC.23090-31]: MPEG\_haptics object.profile is a string that may hold the values "simple-parametric" or "main". When profile is not present, the default value "main" **SHOULD** be inferred.

lvl: This parameter indicates the level used to generate the encoded stream, as defined in [ISO.IEC.23090-31]: MPEG\_haptics object.level is an integer that may hold the values 1 or 2. When lvl is not present, the default value 2 **SHOULD** be inferred.

**maxlod:** This parameter indicates the maximum level of details (LODs) to use for the avatar(s). The avatar LOD is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.avatar.object.lod is an integer that may hold the value 0 or a positive integer.

**avtypes:** This parameter indicates, using a comma-separated list, the types of haptic perception represented by the avatar(s). The avatar type is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.avatar.object.type is a string that may hold values among "Vibration", "Pressure", "Temperature", or "Custom".

**modalities:** This parameter indicates, using a comma-separated list, haptic perception modalities (e.g., pressure, acceleration, velocity, position, temperature, etc.). The perception modality is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.perception.object.perception\_modality is a string that may hold values among "Pressure", "Acceleration", "Velocity", "Position", "Temperature", "Vibrotactile", "Water", "Wind", "Force", "Electrotactile", "Vibrotactile Texture", "Stiffness", "Friction", "Humidity", "User-defined Temporal", "User-defined Spatial", or "Other".

**bodypartmask:** This parameter is an integer that indicates, using a bitmask, the location of the devices or actuators on the body. The body part mask is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.reference\_device.object.body\_part\_mask is a 32-bit integer that may hold a bit mask using bit positions defined in Table 7 of [\[ISO.IEC.23090-31\]](#).

**maxfreq:** This parameter is an integer that indicates the maximum frequency of haptic data for vibrotactile perceptions (Hz). Maximum frequency is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.reference\_device.object.maximum\_frequency.

**minfreq:** This parameter is an integer that indicates the minimum frequency of haptic data for vibrotactile perceptions (Hz). Minimum frequency is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.reference\_device.object.minimum\_frequency.

**dvctypes:** This parameter is an integer that indicates, using a comma-separated list, the types of actuators. The device type is defined in [\[ISO.IEC.23090-31\]](#): MPEG\_haptics.reference\_device.object.type is a string that may hold values among "LRA", "VCA", "ERM", "Piezo", or "Unknown".

**silencesupp:** This parameter is an integer that indicates whether silence suppression should be used (value 1) or not (value 0). When silencesupp is not present, the default value 0 **SHOULD** be inferred.

## 6.2. SDP Parameter Registration

This memo registers a 'haptics' token in the media subregistry of the "Session Description Protocol (SDP) Parameters" registry group. This registration contains the required information elements outlined in the SDP registration procedure defined in [Section 8.2](#) of [\[RFC8866\]](#).

Contact name: Hyunsik Yang

Contact email address: [hyunsik.yang@interdigital.com](mailto:hyunsik.yang@interdigital.com)

Name being defined (as it will appear in SDP): haptics

Type of name: media

Description: The 'haptics' media type for the Session Description Protocol is used to describe a media stream whose content can be rendered as touch-related sensations. The media subtype further describes the specific format of the haptics stream. The 'haptics' media type for SDP is used to establish haptics media streams.

Reference: RFC 9993

## 7. SDP Considerations

The mapping of the above-defined payload format media type to the corresponding fields in SDP is done according to [\[RFC8866\]](#).

The media name in the "m=" line of SDP **MUST** be haptics.

The encoding name in the "a=rtpmap" line of SDP **MUST** be hmpg.

The clock rate in the "a=rtpmap" line may be any sampling rate, typically 8000.

The optional parameters (defined in [Section 6.1](#)), when present, **MUST** be included in the "a=fmtp" line of SDP. This is expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs. Parameter values, including string values, **MUST** be written without quotation marks (") in SDP. Parameter values that are strings are not case sensitive and **SHOULD** be written in lowercase.

An example of media representation corresponding to the hmpg RTP payload in SDP is as follows:

```
m=haptics 43291 UDP/TLS/RTP/SAVPF 115
a=rtpmap:115 hmpg/8000
a=fmtp:115 profile=main;lvl=1;ver=2025
```

### 7.1. SDP Offer/Answer Considerations

When using the offer/answer procedure described in [\[RFC3264\]](#) to negotiate the use of haptic, the following considerations apply:

When used for a unidirectional stream, the SDP parameters represent the properties of the sender (on the sending side) and of the receiver (on the receiving side). When used for a sendrecv stream, the SDP parameters represent the properties of the receiver.

The receiver properties expressed using the SDP parameters 'ver', 'profile', and 'lvl' correspond to implementation capabilities. The ver, profile, and lvl parameters **MUST** be used symmetrically in SDP offer and answer. That is, their values in the answer **MUST** match those in the offer, either explicitly signaled or implicitly inferred. In the same session, ver, profile, and lvl **MUST NOT** be changed in subsequent offers or answers.

The properties expressed using SDP parameters other than 'ver', 'profile', and 'lvl' are provided as recommendations for efficient data transmission and are not binding, meaning that a sender is encouraged but not required to conform to the parameters specified by the receiver. These properties **MAY** be set to different values in offers and answers. These properties **MAY** be updated in subsequent offers or answers.

Any receiver compliant with [ISO.IEC.23090-31] **MUST** be capable of decoding any stream with a compatible version, profile, and level. A receiver supporting a more general profile will accept a stream corresponding to the same or a less general profile (e.g., "main" is more general than "simple-parametric"). A receiver supporting a given level will accept a stream corresponding to the same or a lower level. A receiver supporting a given version will accept a stream corresponding to the same version and **MAY** accept other versions. A receiver **MAY** ignore any part of a received stream, e.g., that it does not have support for rendering.

The haptic signal can be sampled at different rates. The MPEG Haptics Coding standard does not mandate a specific frequency. A typical sample rate is 8000Hz.

The parameter 'ver' indicates the version of the haptic standard specification. If it is not specified, the parameter 'ver' indicates the version of the haptic standard specification. If it is not specified, the value "2025" indicating the MPEG Haptics Coding standard ISO/IEC 23090-31:2025 [ISO.IEC.23090-31] **SHOULD** be inferred, although the sender and receiver **MAY** use a specific value based on an out-of-band agreement. The parameter 'profile' is used to restrict the number of tools used (e.g., the simple-parametric profile enables simpler implementations than the main profile). If it is not specified, the most general profile "main" **SHOULD** be inferred, although the sender and receiver **MAY** use a specific value based on an out-of-band agreement. The parameter 'lvl' is used to further characterize implementations within a given profile, e.g., according to the maximum supported number of channels, bands, and perceptions. If it is not specified, the most general level "2" **SHOULD** be inferred, although the sender and receiver **MAY** use a specific version based on an out-of-band agreement.

Other parameters can be used to indicate bitstream properties as well as receiver capabilities. The parameters 'maxlod', 'avtypes', 'bodypartmask', 'maxfreq', 'minfreq', 'dvctypes', and 'modalities' can be sent by a sender to reflect the characteristics of bitstreams and can be set by a receiver to reflect the nature and capabilities of local actuator devices or a preferred set of bitstream properties. For example, different receivers **MAY** have different sets of local actuators, in which case these parameters can be used to select a stream adapted to the receiver. In some other cases, some receivers **MAY** indicate a preference for a set of bitstream properties such as perceptions, min/max frequency, or body-part-mask, which contribute the most to the user experience for a given application, in which case these parameters can be used to select a

stream that includes and possibly prioritizes those properties. For example, if the haptic stream server provides more information than the body mask specified by the receiver, the additional information can be either integrated into a single effect or ignored by the receiver.

The parameter 'silencesupp' can be used to indicate sender and receiver capabilities or preferences. This parameter indicates whether silence suppression should be used, as described in [Section 5.4](#). If it is not specified, the value "0", indicating no silence suppression, **SHOULD** be inferred, although the sender and receiver **MAY** use silence suppression based on an out-of-band agreement.

## 7.2. Declarative SDP Considerations

When haptic content over RTP is offered with SDP in a declarative style, the parameters capable of indicating both bitstream properties as well as receiver capabilities are used to indicate only bitstream properties. For example, in this case, the parameters 'maxlod', 'bodypartmask', 'maxfreq', 'minfreq', 'dvctypes', and 'modalities' declare the values used by the bitstream, not the capabilities for receiving bitstreams. A receiver of the SDP is required to support all parameters and values of the parameters provided; otherwise, the receiver **MUST** reject or not participate in the session. It falls on the creator of the session to use values that are expected to be supported by the receiving application.

## 8. Congestion Control Considerations

The general congestion control considerations for transporting RTP data apply to HMPG haptics over RTP as well [[RFC3550](#)].

It is possible to adapt network bandwidth usage by adjusting either the encoder bit rate or the stream content (e.g., the LOD, body parts, actuator frequency range, target device types, and modalities). The considerations in this section are applicable to best-effort networks and controlled environments.

In case of congestion, a receiver or intermediate node **MAY** prioritize independent packets over dependent ones, since the non-reception of an independent MIHS unit can prevent the decoding of multiple subsequent dependent MIHS units. In case of congestion, a receiver or intermediate node **MAY** prioritize initialization MIHS units over other units, since initialization MIHS units contain metadata used to reinitialize the decoder, and **MAY** drop silent MIHS units before other types of MIHS units, since a receiver **MAY** interpret a missing MIHS unit as a silence. It is also possible, using the layer field of the RTP payload header, to allocate MIHS units to different layers based on their content to prioritize haptic data that contributes the most to the user experience. In case of congestion, intermediate nodes and receivers **SHOULD** use the MIHS layer value to determine the relative importance of haptic RTP packets.

Receivers should monitor timestamps and treat gaps as loss of the corresponding MIHS units. MIHS units, as defined in [[ISO.IEC.23090-31](#)], should be checked for structural integrity according to their type. When CRC16 or CRC32 information is present in MIHS units, receivers must validate data integrity, and units failing Cyclic Redundancy Checks (CRCs) should be treated as lost. Receivers should further monitor indicators of service degradation such as

unexpected silent gaps, repeated decoder reinitializations, or decoding failures. Receivers should report packet loss to the sender using RTCP Receiver Reports [RFC3550] and, when available, may report detailed loss and jitter metrics using mechanisms described in [RFC4585].

## 9. Security Considerations

The RTP payload format is subject to security threats commonly associated with RTP payload formats, as well as threats specific to the interaction of haptic devices with the physical world and threats associated with the use of compression by the codec. Security considerations for threats commonly associated with RTP payload formats are outlined in [RFC3550], as well as in RTP profiles such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], and RTP/SAVPF [RFC5124].

Haptic sensors and actuators operate within the physical environment. This introduces the potential for information leakage through sensors or damage to actuators due to data tampering. Additionally, misusing the functionalities of actuators (such as force, position, temperature, vibration, electrotactile, etc.) may pose a risk of harm to the user, for example, by setting keyframe parameters (e.g., amplitude, position, and frequency) or channel gain to a value that surpasses a permissible range. While individual devices can implement security measures to reduce or eliminate those risks on a per-device basis, in some cases, harm can be inflicted by setting values that are permissible for the individual device. For example, causing contact with the physical environment or triggering unexpected force feedback can potentially harm the user. Each haptic system should therefore implement system-dependent security measures, which are more error prone. To limit the risk that attackers exploit weaknesses in haptic systems, it is important that haptic transmission be protected against malicious traffic injection or tampering.

However, as "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. The responsibility for implementing security mechanisms lies with the application developer. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201], although [RFC7201] is now considered dated and several mechanisms described therein have since evolved.

Applications **SHOULD** use appropriate and current strong security mechanisms. For modern best practices, applications can consider the following options:

- (D)TLS-based protection: For guidance on using TLS 1.3 and DTLS, applications should refer to BCP 195, including [RFC9325], which provides up-to-date recommendations.
- IPsec-based protection: Relevant and current protocol specifications include [RFC4303] ("IP Encapsulating Security Payload (ESP)") and [RFC7296] ("Internet Key Exchange Protocol Version 2 (IKEv2)").

This document does not mandate a specific security mechanism. Instead, applications are responsible for selecting mechanisms that follow current best practices for confidentiality, integrity, and source authentication and that reflect the evolving security landscape beyond what is covered in [\[RFC7201\]](#).

The haptic codec used with this payload format uses a compression algorithm (see Sections 8.2.8.5 and 8.3.3.2 in [\[ISO.IEC.23090-31\]](#)). An attacker may inject pathological datagrams into the stream that are complex to decode and cause the receiver to be overloaded, similarly to [\[RFC3551\]](#).

End-to-end security with authentication, integrity, or confidentiality protection will prevent a Media-Aware Network Element (MANE) from performing media-aware operations other than discarding complete packets. In the case of confidentiality protection, it will even be prevented from discarding packets in a media-aware way. To be allowed to perform such operations, a MANE is required to be a trusted entity that is included in the security context establishment.

## 10. IANA Considerations

### 10.1. Media Type Registration Update

This memo updates the 'hmpg' haptic subtype defined in [\[RFC9695\]](#) for use with the MPEG-I haptics streamable binary coding format described in ISO/IEC 23090-31: Haptics coding [\[ISO.IEC.23090-31\]](#). This memo defines optional parameters for this type in [Section 6.1](#). The original subtype registration for 'haptics/hmpg', registered with IANA in [\[RFC9695\]](#), did not include any required or optional parameters. This document introduces optional parameters to enable extended functionality while maintaining backward compatibility.

A mapping of the parameters into SDP [\[RFC8866\]](#) is also provided for applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use SDP. The receiver **MUST** ignore any parameter unspecified in this memo.

IANA has updated the registration for 'haptics', described in [Section 6.2](#), in the "haptics" registry within the "Media Types" registry group and listed document as an additional reference.

The following entries identify the updates to the 'media/haptics' registration:

Type name: haptics

Subtype name: hmpg

The following entries are replaced by this memo:

Optional parameters: See [Section 6.2](#) of RFC 9993

Person & email address to contact for further information: Yeshwant Muthusamy (yeshwant@yeshvik.com) and Hyunsik Yang (hyunsik.yang@interdigital.com)

## 10.2. New SDP Parameters Media Registration

IANA has registered the following in the "media" registry within the "Session Description Protocol (SDP) Parameters" registration group.

Type	SDP Name	Reference
media	haptics	RFC 9993

Table 2

## 11. References

### 11.1. Normative References

- [ISO.IEC.23090-31] ISO/IEC, "Information technology - Coded representation of immersive media - Part 31: Haptics coding", ISO/IEC 23090-31:2025, 2025, <<https://www.iso.org/standard/86122.html>>.
- [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>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [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>>.
- [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>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.
- [RFC9695] Muthusamy, Y. K. and C. Ullrich, "The 'haptics' Top-Level Media Type", RFC 9695, DOI 10.17487/RFC9695, March 2025, <<https://www.rfc-editor.org/info/rfc9695>>.

### 11.2. Informative References

- [RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, DOI 10.17487/RFC2736, December 1999, <<https://www.rfc-editor.org/info/rfc2736>>.

- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [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>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [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>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<https://www.rfc-editor.org/info/rfc7296>>.
- [RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <<https://www.rfc-editor.org/info/rfc8088>>.
- [RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <<https://www.rfc-editor.org/info/rfc9325>>.

## Acknowledgments

Thanks to Philippe Guillotel, Quentin Galvane, Jonathan Lennox, Marius Kleidl, and Stephan Wenger for the comments and discussions about this document.

## Authors' Addresses

**Hyunsik Yang**

InterDigital

United States of America

Email: [hyunsik.yang@interdigital.com](mailto:hyunsik.yang@interdigital.com)**Xavier de Foy**

InterDigital

Canada

Email: [xavier.defoy@interdigital.com](mailto:xavier.defoy@interdigital.com)