

avtcore
Internet-Draft
Intended status: Standards Track
Expires: 23 August 2026

HS Yang
X. de Foy
A. Hamza
InterDigital
I. Bouazizi
Qualcomm
19 February 2026

RTP Payload Format for Avatar Representation Format (ARF) Animations
draft-ietf-avtcore-rtp-avatar-00

Abstract

This memo outlines RTP payload formats for the animation stream format as defined in the ISO/IEC 23090-39 specification (MPEG-I Avatar Representation Format). An animation stream format is composed of Avatar Animation Units (AAU) including an AAU header and zero or more AAU packets. The RTP payload header format allows for packetization of an AAU unit in an RTP packet payload as well as fragmentation of an AAU into multiple RTP packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 August 2026.

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. Definition, and abbreviations	3
3.1. General	3
3.2. Definitions	3
3.3. Abbreviation	3
4. Avatar Representation Format	4
4.1. Overview of Avatar Representation Format (informative)	4
4.2. Avatar Animation Streams	4
5. Payload format for ARF Animations	5
5.1. General	6
5.2. RTP Header Usage	6
5.3. RTP Payload Header for Avatar Animation Unit	7
5.4. Payload structures	7
5.4.1. General	8
5.4.2. Single Unit Payload Structure	9
5.4.3. Fragmented Unit Payload Structure	9
5.4.4. Aggregation Packet Payload Structure	10
6. AAU Transmission Considerations	12
7. Payload Format Parameters	13
7.1. Media Type Registration Update	13
7.2. Optional Parameters Definition	14
8. Congestion Control Consideration	14
9. SDP Considerations	14
9.1. SDP Offer/Answer Considerations	15
9.2. Declarative SDP Considerations	16
10. IANA Considerations	16
10.1. Avatar Animation Media Registration	16
11. Security Considerations	17
12. References	17
12.1. Normative References	17
12.2. Informative References	17
Authors' Addresses	19

1. Introduction

Avatars are digital representations of users in the metaverse, a set of virtual worlds where people can interact with each other in real-time. Users can customize different aspects of their avatars, such as clothing, accessories, and even physical attributes. Avatars allow users to express themselves and create a unique digital identity within the metaverse. The integration, animation, and representation of avatars in real-time communication services is essential to enable immersive experiences.

[ISO.IEC.23090-39] specifies the Avatar Representation Format (ARF) to offer an interoperable exchange format for the storage, carriage and animation of 3D avatars. It defines the "Avatar Animation Unit"(AAU) as a unit of packetization suitable for Avatar animation streaming, and similar in essence to the NAL unit defined in some video specifications. This document describes how AAUs can be transmitted using the RTP protocol. This document followed recommendations in [RFC8088] and [RFC2736] for RTP payload format writers.

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. Definition, and abbreviations

3.1. General

This document uses the definitions of the Avatar Representation Format [ISO.IEC.23090-39]. Some of these terms are provided here for convenience.

3.2. Definitions

Animation Streams: timed data used to animate the base avatar.

3.3. Abbreviation

ARF Avatar Representation Format

AAU Avatar Animation Unit

LoD Level of Detail

4. Avatar Representation Format

4.1. Overview of Avatar Representation Format (informative)

The Avatar Representation Format (ARF) defines two key components of an avatar animation system: the Base Avatar Format and the Animation Stream Format.

The Base Avatar Format defines a standardized structure for avatar models, allowing them to be stored in digital asset repositories. This ensures that core avatar assets can be reliably accessed and animated by receiving systems. In contrast, the Animation Stream Format specifies how animation data is organized and transmitted between sender and receiver. It defines the encoding of facial and body animation, enabling data captured from input devices such as head-mounted displays (HMDs) and sensors to be consistently interpreted across different systems for animating associated avatars. Figure 1 describe an Avatar reference architecture.

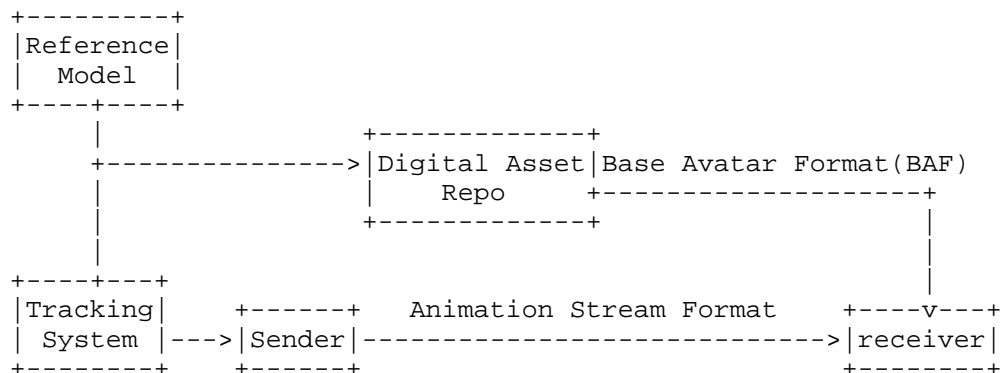


Figure 1: Avatar reference architecture

4.2. Avatar Animation Streams

Animation streams are timed data used to animate an avatar. This data includes skeletal, blend shape set, and other animation-related information. Animation stream format defines how animation data is structured and carried between senders and receivers. This format defines how facial and body animation information is encoded, allowing data captured from input devices like Head-Mounted Displays (HMDs) and sensors to be consistently interpreted across different systems for the animation of associated avatars.

Avatar animation data may be stored as samples in an avatar container, such as the MPEG Avatar Representation Format container [ISO.IEC.23090-39], along with the avatar model representation. This data may also be generated on-the-fly as cameras and sensor capture a person’s motion and generate corresponding commands to mimic this movement for an avatar that represent the user. Avatar animation samples may be structured into a bitstream comprising a sequence of Avatar Animation Units (AAUs), whose general structure is provided in Figure 2.

Each AAU is associated with an Avatar ID that indicates the target avatar to which the animation data applies. In addition, it is also associated with a Level of Detail (LoD), which indicates the quality of the avatar animation. Different LoDs may, for example, correspond to different numbers of animation joints and thus different animation sample sizes. The animation data within an AAU is generated by an tracking/animation framework (e.g., OpenXR or ARKit) based on a schema identified using a URN. An avatar container corresponds to a single avatar ID, and each asset within the container holds data for one or more LoDs.

Avatar animation content can be transmitted over one or more streams, depending on applications. For example, an application may transmit animations for a single avatar in different streams or may transmit animations for multiple avatars in a single stream. In some cases, an application may choose to stream all avatar animations at the same level of detail. In some other cases, an application could associate different avatars or avatar parts with different levels of details, depending on the position of the avatar, and possibly changing the level of detail over time. In all cases, the receiver should be aware of the avatar IDs and/or levels of detail that are transmitted in a stream, to make sure it has the necessary assets to render the avatar animation. The receiver can use the avatar ID and level of detail associated with an AAU to transmit the AAU to an animation player instance that has the proper assets.

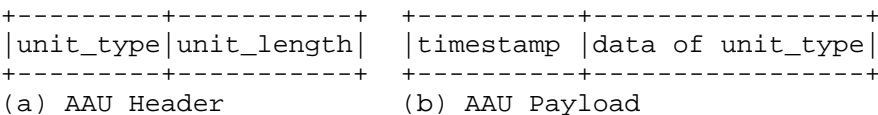


Figure 2: The structure of AAU Header(a) and Payload(b)

5. Payload format for ARF Animations

5.1. General

This section describes details related to the RTP payload format definitions for the Avatar codec defined in [ISO.IEC.23090-39]. Aspects related to RTP header, RTP payload header and general payload structure are considered.

5.2. RTP Header Usage

The RTP header is defined in [RFC3550] and represented in Figure 3. Some of the header field values are interpreted as follows.

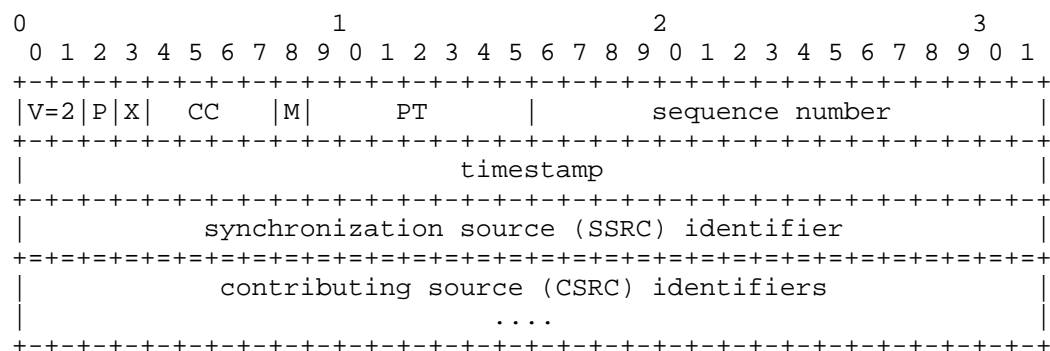


Figure 3: RTP header for Avatar Animation Unit

Marker bit (M): 1 bit.

The marker bit SHOULD be set to one in the first RTP packet after an any idle period. This is aligned with the use of the marker bit in audio codecs. This can for example be used for jitter buffer adaptation. The marker bit in all other packets MUST be set to zero.

Payload type (PT): 7 bits

The assignment of a payload type MUST be performed either through the profile used or in a dynamic way.

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550]

Timestamp: 32 bits

A timestamp representing the sampling time of the earliest AAU (Avatar Animation Unit) in the payload. The AAU defines `aau_timestamp` in its payload. The timestamp in seconds can be calculated as: `timestamp / timescale`.

Synchronization source (SSRC): 32 bits

Used to identify the source of the RTP packets. By definition a single SSRC is used for all parts of a single bitstream. The remaining RTP header fields are used as specified in [RFC3550].

5.3. RTP Payload Header for Avatar Animation Unit

The RTP Payload Header follows the RTP header. Figure 4 describes RTP Payload Header.

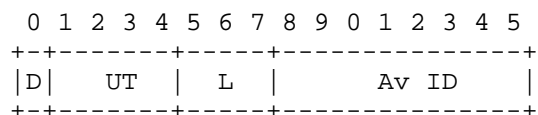


Figure 4: RTP Payload header for Avatar Animation

D (Dependency, 1 bit): this field indicates whether an AAU included in the avatar animation packet payload is an independent AAU (D=0) or dependent (D=1). If D=1, the AAU is dependent on other AAUs for decoding. If D=0, the AAU can be decoded independently.

UT (Unit Type, 4 bits): this field indicates the type of the payload, which can be the type of the AAU for single unit payload, or the type of the payload otherwise, as shown in Figure 5.

L (Level of Detail, 3 bits): this field indicates the level of detail to which the AAU(s) within the RTP packet applies. If the RTP packet includes multiple AAUs, L MUST indicate the lowest LoD.

AvID (Avatar ID, 8 bits): this field identifies the avatar to which the animation data in the payload of the packet applies. The avatar corresponds to the digital assets to be animated.

5.4. Payload structures

5.4.1. General

Three different types of RTP packet payload structures are specified. A single unit packet contains a single AAU in the payload. A fragmentation unit contains a subset of an AAU. An aggregation packet contains multiple Avatar animation units in the payload. The unit type (UT) field of the RTP payload header, as shown in Figure 5, identifies both the payload structure and, in the case of a single-unit structure, also identifies the type of AAU present in the payload.

Unit Type	Payload Structure	Name
0	N/A	Reserved
1	Single	Configuration AAU
2	Single	Blendshape AAU
3	Single	Joint AAU
4	Single	Landmark AAU
5	Single	Texture AAU
13	Aggr	Aggregation Packet (STAP)
14	Aggr	Aggregation Packet (MTAP)
15	Frag	Fragmentation Unit

Figure 5: Payload structure type for Avatar

The payload structures are represented in Figure 6. The single unit payload structure is specified in Section 5.4.2. The fragmented unit payload structure is specified in Section 5.4.3. The aggregation unit payload structure is specified in Section 5.4.4.

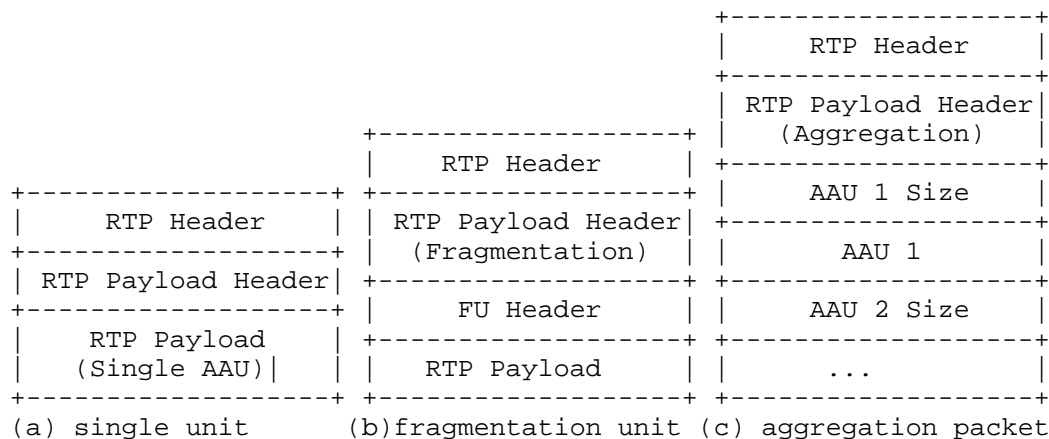


Figure 6: RTP Transmission mode

5.4.2. Single Unit Payload Structure

In a single unit payload structure, as described in Figure 7, the RTP packet contains the RTP header, followed by the Payload Header and one single AAU. The Payload Header follows the structure described in Section 5.3. The payload contains an AAU as defined in [ISO.IEC.23090-39].

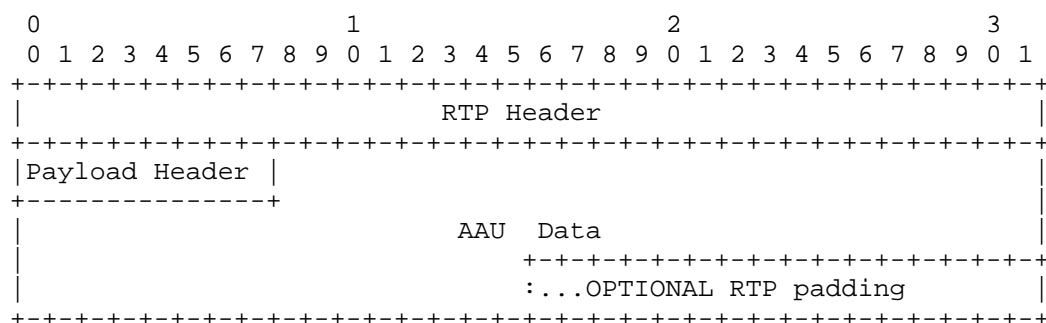


Figure 7: Single AAU payload structure

5.4.3. Fragmented Unit Payload Structure

In a fragmented unit payload structure, as described in Figure 8, the RTP packet contains the RTP header, followed by the Payload Header, a Fragmented Unit (FU) header, and an AAU fragment. The Payload Header follows the structure described in Section 5.3. The value of the UT field of the Payload Header is 15. The FU header follows the structure described in Figure 9.

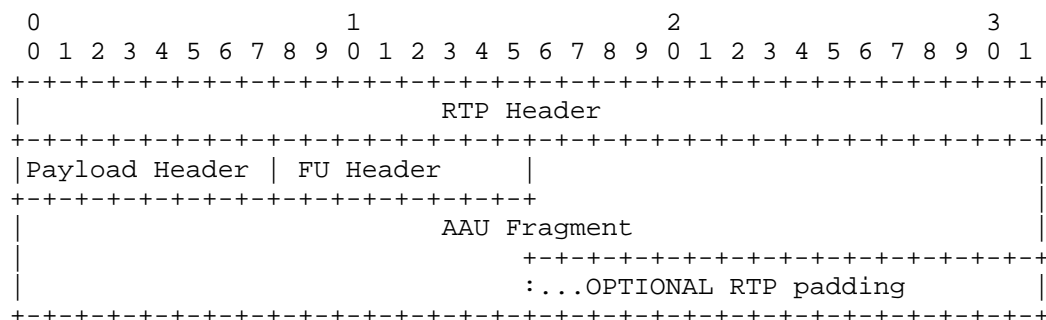


Figure 8: Fragmentation unit header

FU headers are used to enable fragmenting a single AAU into multiple RTP packets. Fragments of the same AAU 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 9 describes a FU header, including the following fields:

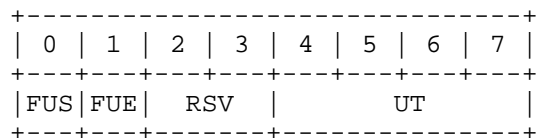


Figure 9: Fragmentation unit header

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.

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

UT (Unit Type, 4 bits): this field indicates the type of the AAU this fragment belongs to, using values defined in Figure 5.

5.4.4. Aggregation Packet Payload Structure

In an aggregation packet, as described in Figure 10, the RTP packet contains an RTP header, followed by a Payload Header, and, for each aggregated AAU, an AAU size followed by the AAU. The Payload Header follows the structure described in Section 5.3.

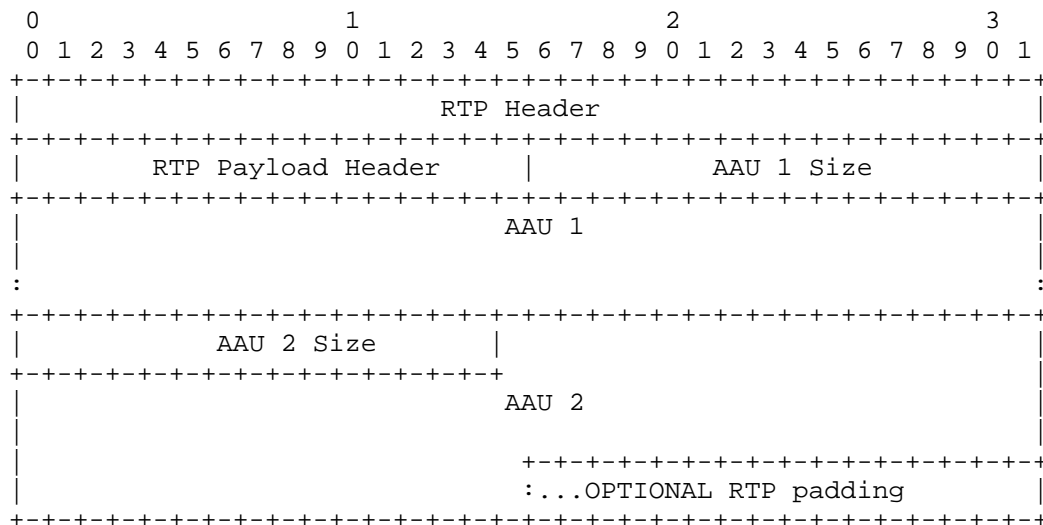


Figure 10: Single-Time Aggregation Packet

Figure 10 shows a Single-Time Aggregation Packet (STAP), which can be used to transmit multiple avatar animation units that correspond to the same timestamp. For example, if two different AAUs are used for different animations for different parts of the avatar, they can be transmitted together in a single STAP. The default sizes of the avatar animation unit length field is 16 bits. The value of the UT field of the Payload Header is 13.

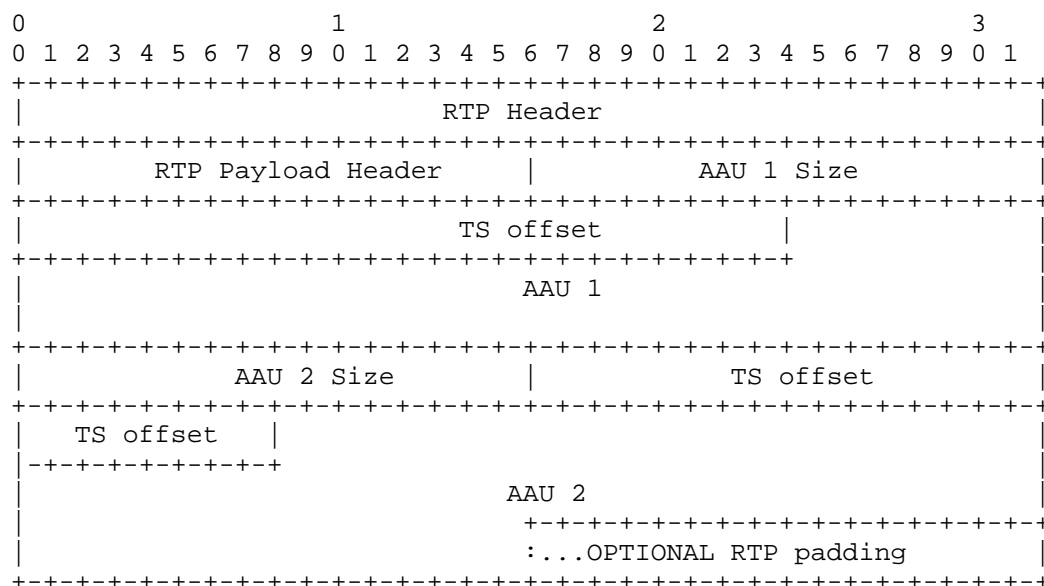


Figure 11: Multiple-time aggregation packet

Figure 11 shows a multi-time aggregation packet. It is used to transmit multiple Avatar animation 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 default sizes of the TS offset and the AAU length fields are 16 bits each. The value of the UT field of the Payload Header is 14. In case of MTAP, the timestamp offset field **MUST** be set to the value of (AAU-time of the animation 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 AAU-time.

6. AAU Transmission Considerations

The following considerations apply for the streaming of avatar animation units over RTP:

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 [RFC5104] messages with avatar animation. 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 avatar animation, an appropriate decoder refresh point is a configuration AAU. The configuration AAU point enables a decoder to be reset to a known state and be able to decode all AAUs following it.

7. Payload Format Parameters

This section describes payload format optional parameters. A mapping of the parameters into the Session Description Protocol (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.

7.1. Media Type Registration Update

The receiver MUST ignore any parameter unspecified in this memo.

Type name: application

Subtype name: ampg

Required parameters: N/A

Optional parameters: Optional parameters are defined in the following section.

Encoding considerations: This type is only defined for transfer via RTP [RFC3550].

Security considerations: Please see section 11.

Interoperability considerations: N/A

Published specification: Please refer to [ISO.IEC.23090-39]

Applications that use this media type: Any application that relies on Avatar media services over RTP

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information:

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors' Address section of this memo.

Change controller: IETF avtcore@ietf.org (mailto:avtcore@ietf.org)

Provisional registration? (standards tree only): No

7.2. Optional Parameters Definition

`_version_` provides the year of the edition and amendment of the specifications followed by this RTP payload type.

`_frameworks_` provides a comma-separated list of the tracking framework names (URNs) used to generate the encoded stream.

`_avatar-ids_` provides an associations between avatar IDs for which animation data is carried in the animation stream, and their corresponding ARF containers. This parameter is provided as a comma-separated list of "key/value" pairs, where the key is the avatar id (an integer between 0 and 255 inclusive) and the value is a base64 encoded string. The semantic of the value is application dependent and can for example be a URL to the ARF container.

`_avatar-lods_` indicates which levels of detail are used in the avatar animation stream. This parameter is a comma-separated list of integers.

8. Congestion Control Consideration

General congestion control considerations for RTP transmission, as described in [RFC3550], also apply to avatar streaming over RTP. By adjusting the SDP 'avatar-lod' parameter, it is possible to reduce processing load and optimize bandwidth usage, thereby partially mitigating congestion issues. The ability to adapt the level of detail dynamically allows senders or receivers to manage computational complexity and network resource consumption based on system constraints or user context. Moreover, in use cases such as video conferencing, different levels of detail may be applied to different parts of the avatar and transmitted via separate streams.

9. SDP Considerations

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

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

The encoding name in the "a=rtpmap" line of SDP MUST be ampg

The clock rate in the "a=rtpmap" line may be any sampling rate and SHOULD match the acu timescale value of the AAU CONFIG unit.

The OPTIONAL parameters (defined in Section 7.2), 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.

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

```
m=application 43291 UDP/TLS/RTP/SAVPF 120 a=rtpmap:120 ampg/8000
a=fmtp:120
frameworks=urn:mpeg:avatar:v1:openxr:face,urn:mpeg:avatar:v1:
openxr:body;version=2025;avatar-ids=1/
aHR0cDovL2V4YWlwbGUuY29tL2F2YXRhcjEuYXJm,
2/aHR0cDovL2V4YWlwbGUuY29tL2F2YXRhcjIuYXJm;avatar-lods=0,1,2
```

9.1. SDP Offer/Answer Considerations

When using the offer/answer procedure described in [RFC3264] to negotiate the use of avatar animations, the following considerations apply:

The SDP parameter version identifies the version of the avatar animation specification. It MUST be used symmetrically in SDP offer and answer, and it MUST NOT be changed in subsequent offers or answers within the same session. If it is not specified, the initial version of the specification SHOULD be assumed. Any receiver compliant with [ISO.IEC.23090-39] must accept any stream with a compatible version.

The properties expressed using SDP parameters other than 'version' 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. These properties can be sent by a sender to reflect the characteristics of bitstreams and can be set by a receiver to reflect the capabilities and configurations of the local player device, or a preferred set of bitstream properties.

The parameter `frameworks` indicates that the AAUs of the stream carry animation data that conforms to the one or more framework names (URNs) signalled with this parameter. The sender uses this parameter to indicate the formats of data transported within the AAUs of the stream. The receiver, to be able to render the animations, needs to support the formats associated with signalled frameworks. The receiver uses this parameter to indicate the desired framework names.

The parameter `avatar-ids` indicates that a stream corresponds to the one or more avatar IDs signalled with this parameter. The sender uses this parameter to indicate that the AAUs of the stream carry data corresponding to the signalled avatar IDs. The receiver uses this parameter to indicate the avatar IDs it wishes to receive data for.

The parameter `avatar-lods` indicates that the AAUs of the stream correspond to one or more levels of detail signalled with this parameter. The sender uses this parameter to indicate available LoDs, and the receiver uses it to select the desired LoD. To render the animations, the receiver **MUST** have loaded the corresponding assets associated with the selected level(s) of detail.

A receiver may ignore any part of a received stream, e.g., that it does not have support for rendering.

9.2. Declarative SDP Considerations

When avatar animation 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 `frameworks`, `avatar-ids`, and `avatar-lods` declare the values used by the bitstream, not the capabilities and configurations 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.

10. IANA Considerations

10.1. Avatar Animation Media Registration

New media types will be registered with IANA; see Section 7.1.

11. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124].

For example, an avatar may contain sensitive information derived from a user's personal data, and thus requires protection against leakage or tampering during transmission. When avatar data is delivered over a network or downloaded from a server, it is critical to ensure its integrity and confidentiality to prevent unauthorized access, modification, or confidentiality.

However, as "Securing the RTP Protocol 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. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this Security Considerations section discusses the security impacting properties of the payload format itself.

12. References

12.1. Normative References

[ISO.IEC.23090-39]

ISO/IEC, "Information technology - Coded representation of immersive media - Part 39: Avatar Representation Format", ISO/IEC 23090-39, 2025,
<<https://www.mpeg.org/standards/MPEG-I/39/>>.

12.2. Informative 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/rfc/rfc2119>>.

- [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/rfc/rfc2736>>.
- [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/rfc/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/rfc/rfc3550>>.
- [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/rfc/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/rfc/rfc3711>>.
- [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/rfc/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/rfc/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/rfc/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/rfc/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/rfc/rfc7202>>.
- [RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <<https://www.rfc-editor.org/rfc/rfc8088>>.
- [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/rfc/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/rfc/rfc8866>>.

Authors' Addresses

Hyunsik Yang
InterDigital
United States of America
Email: hyunsik.yang@interdigital.com

Xavier de Foy
InterDigital
Canada
Email: xavier.defoy@interdigital.com

Ahmed Hamza
InterDigital
Canada
Email: ahmed.hamza@interdigital.com

Imed Bouazizi
Qualcomm
Canada
Email: BOUAZIZI@qti.qualcomm.com