
# RTP Payload Format for Avatar Representation Format (ARF) Animations

**https://datatracker.ietf.org/doc/draft-hsyang-avtcore-rtp-avatar/**

H. Yang, X. de Foy, A.Hamza, I. Bouazizi

IETF 124, Nov 2025

# Outline

- **Review IETF123**
- **Updated section**
- **Next Step**

# Review IETF123

- ***We received questions from Magnus, Mojo and Jonathan.***
  - **How large is the data being sent and what are its properties?**
    - The avatar has five types of AAUs: Configuration, Blendshape, Joint, Landmark, and Texture. Each AAU contains only the delta values of positions relative to the reference model. In simple examples including an avatar body and face, AAUs for the body (JOINT) AAUs for face (BLENDSHAPE) could be present, some of which would be larger than the typical MTU
  - **Does it need full reliability, or can it tolerate some loss? Is the animation expressed in absolute coordinates or incremental movements (deltas)?**
    - The animation data can tolerate some packet loss. When data is lost, the motion may not appear perfectly smooth, but the system can recover because each update references a base (or reference) frame. The animation is not expressed in absolute coordinates; however, it behaves as if it were, since each delta is applied relative to a constant reference model (such as a skeleton or mesh).
  - **Are textures transmitted via RTP or referenced separately?**
    - The texture will be downloaded separately before the communication starts. Basically, only animation information is included in the RTP packets.
  - **Will there be an IETF proposal for distributing these graphics assets, or is that handled by the applications themselves?**
    - We are not considering the graphics asset download procedure in the IETF.

# Updated Sections

- ***Revised Abstract***
  - ***We revised Abstract to defined scope clearly.***

Abstract

This memo outlines RTP payload formats for the MPEG-I Avatar data. A Avatar Stream format (ASF) is composed of Avatar animation unit (AAU) including a AAU header and zero or more AAU packets. The RTP Payload header format allows for packetization of a AAU unit in an RTP packet payload as well as fragmentation of a AAU into multiple RTP packets.

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.

- ***Align AAU type with latest MPEG Avatar document***

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

Figure 5: Payload structure type for Avatar

| 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

# Updated Sections

- ***Update optional parameters***
  - ***Add and delete parameters based on MPEG Avatar specifications.***

| 7.2. Optional Parameters Definition | 7.2. Optional Parameters Definition |
| --- | --- |
| _version_ provides the year of the edition and amendment of the specifications followed by this RTP payload type. | _version_ provides the year of the edition and amendment of the specifications followed by this RTP payload type. |
| _profile_ name of the profile used to generate the encoded stream. | _frameworks_ provides a comma-separated list of the tracking framework names (URNs) used to generate the encoded stream. |
| _avatar-id_ identifies the avatars which are the target of the avatar animation stream. This parameter is a comma-separated list of integers. | _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-lod_ indicates which levels of detail are used in the avatar animation stream. This parameter is a comma-separated list of integers. | _avatar-lods_ indicates which levels of detail are used in the avatar animation stream. This parameter is a comma-separated list of integers. |
| _avatar-part-id_ identifies which specific parts of the avatar are associated with the avatar animation stream. This parameter is a comma-separated list of integers. | |
| 8. Congestion Control Consideration | 8. Congestion Control Consideration |

# Next Step

- *Suggestions and feedback are welcome*
- *We request WG adoption call*