**Source: Qualcomm Inc., Lenovo**

**Title: Real-time metadata transport over data channel**

**Agenda Item: 10.5**

**Document for: Discussion and Agreement**

# Introduction

At the 3GPP-SA4#120-e, the Permanent Document for MeCAR v3.0.0 [1] was approved. The Permanent Document specifies AR/MR media type definition and real-time characteristics. The 5G\_RTP Permanent Document v0.0.2 [2] also specifies the requirements tentatively, including reliability, traffic and direction of AR media types.

Based on the metadata definition and the description specified in MeCAR PD, the AR/MR metadata may be categorized into 3 categories.

1. **Device capability**: the metadata specifying the device capabilities and features such as camera information and projection information. The information is usually available at the beginning of an AR/MR session and may not change within an AR/MR session.
2. **Media description**: the metadata describing the space or media object content such as scene description, spatial description, and 3D visual mode. The data size may be large (>10MB) and the information usually does not change frequently and may be event driven.
3. **Interaction**: the metadata representing the user interaction such as FOV, user pose, viewport, gesture, body action, facial expression and AR anchor point. The interaction may trigger the event at the receiver side and the sender may expect low-latency response (50~1000ms). The interaction data size may be small, but the frequency could be high (>1KHz). The data transport may be continuously or may require burst transport. Depending on the applications, the interaction metadata may be synchronized to each other, or synchronized to other media streams, and the reliability may be strict or not strict.

# Real-time interaction metadata transport over data channel

The interaction metadata may be carried in WebRTC data channel [4] given the data size and low-latency requirements. A generic payload format is preferred to carry the real-time interaction metadata.

## Real-time interaction metadata relevant use cases

TS 26.928 [9] defines core use cases and scenarios for XR, and the interaction metadata relevant use cases are as followings:

1. Real-time XR sharing such as real-time 3D communication, AR animated avatar call, and 5G shared spatial data. The relevant metadata such as gesture, position, FOV, viewport, facial expression may be used to render the view, control the animated avatar visual and auditory appearance, The XR clients may also continuously send sensing data to a cloud service which provides the map back to client.
2. XR multimedia Streaming such as immersive 6DoF streaming and emotional streaming. The relevant metadata such as user pose, FOV, viewport and head or body motion may be used to control immersive 3DoF+ or DoF views from different position and angles and help the users to navigate through the scene.
3. Online XR gaming, the relevant metadata such as body movement, viewport and gaming control metadata are essential for the rendering and reaction to enable presence.
4. XR conference such as 360-degree conference meeting, 3D shared experience, 6DoF VR conferencing and XR meeting. The relevant metadata such as pose, FOV, viewport are used to control the view or poster rendering in 360-degree or 6DoF conferencing

## WebRTC data channel

In the WebRTC framework, communication between the parties consists of media (for example, audio and video) and non-media data. Media is sent using the SRTP, and non-media data is handled by using the Stream Control Transmission Protocol (SCTP) [5].

The priority associated with a media flow or data flow is classified as “very-low”, “low”, “medium”, or “high” in the API. WebRTC implementations may attempt to set QoS on the packet sent according to the guidelines in [6].

The SCTP provides the following features for transporting non-media data between browsers:

• Support of multiple unidirectional streams

• Ordered and unordered delivery of user messages

• Reliable and partial reliable transport of user message

An SCTP packet is composed of a common header and chunks. A chunk contains either control information or user data. The SCTP packet format is shown below:

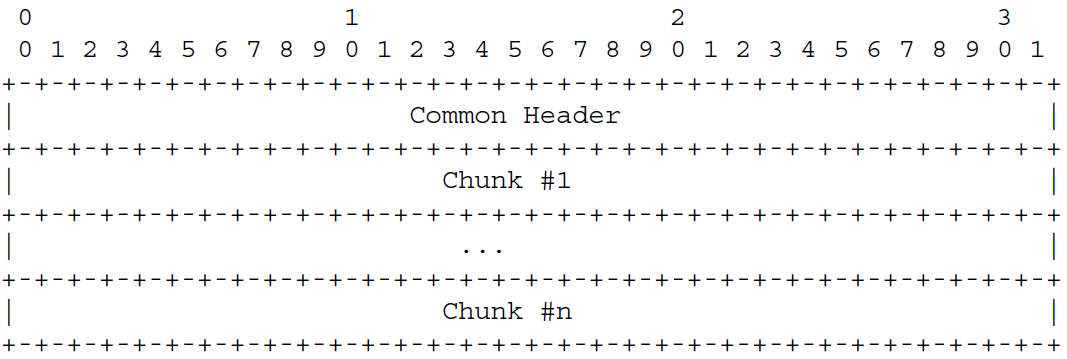


Figure 1 SCTP packet format

The common header field includes a source port number, a destination port number, a verification tag and checksum, as shown in Figure 2.

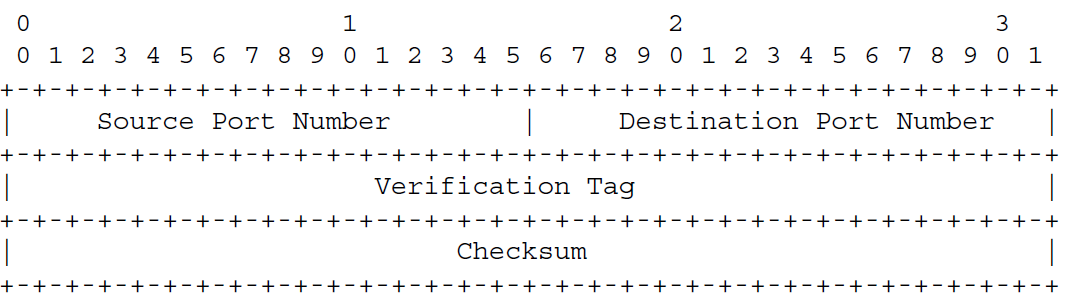


Figure 2 SCTP common header format

Each chunk is formatted with a chunk type field, a chunk-specific Flag field, a Chunk Length field, and a value field as shown below.

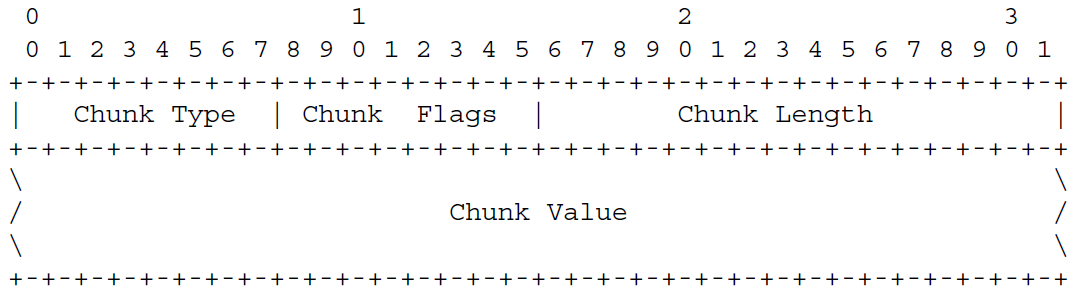


Figure 3 SCTP chunk field format

SCTP defines a number of chunk types, and one of the types is Payload Data (DATA) when chunk type is 0. The payload data format is shown in Figure 4. The U bit indicates unordered data chunk; B and E bit indicate the fragmented user message; the payload protocol identifier represents an application specific protocol identifier and is passed to SCTP by its upper layer and sent to its peer.

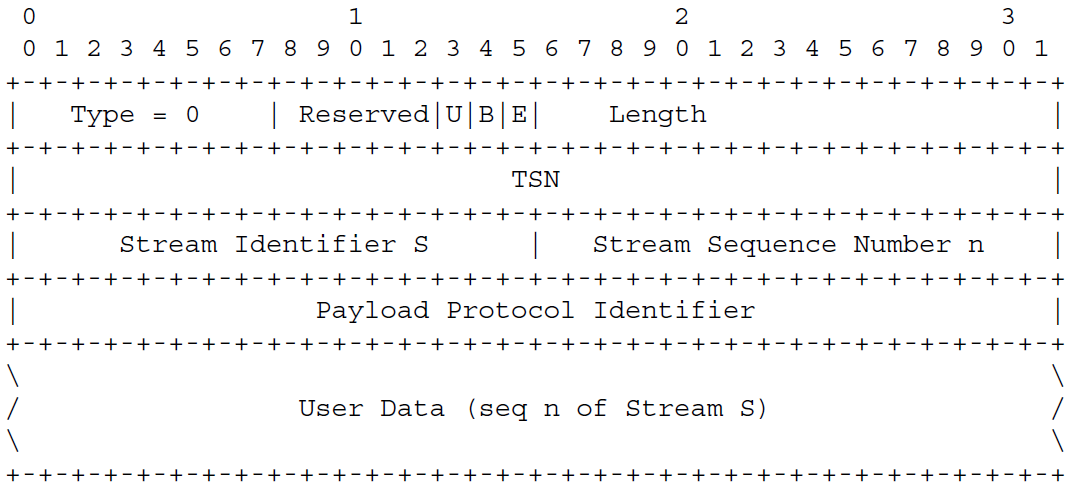


Figure 4 SCTP chunk payload data type format

## Potential Solution for using Data channel for interaction metadata

The WebRTC data channel may be used to convey real-time interactive metadata.

[Editor’s Note: carrying interaction metadata over the Data Channel using this approach may be useful but further study of use cases and other types of interaction metadata will be investigated to identify other potential solutions]

It is proposed to use registered SCTP PPID 53, “WebRTC Binary” [4], to carry the interaction metadata. A generic payload format is shown in Figure 5.

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Type = 0 | Reserved|U|B|E| Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| TSN |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Stream Identifier S | Stream Sequence Number n |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Payload Protocol Identifier = 53 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Subprotocol payload ID |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Metadata type | Metadata attributes |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

\ \

/ Metadata Payload Data (seq n of Stream S) /

\ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 5 Generic data channel payload format for the timed metadata

The 32-bits *Subprotocol payload ID* indicates the subprotocol or specifications used for the metadata payload data format, such as OpenXR.

The 16-bits *Metadata type* field may indicate the metadata type specified in the payload subprotocol. [Editor’s note: the data length may be extended if 16-bits is not sufficient to accommodate the potential metadata type indication such as URN]

The 16-bit *Metadata attributes* field may indicate the metadata attributes such as time synchronization and reliability.

A WebRTC data channel may be optionally mapped by means of an SDP offer/answer [7] or DCEP [8] procedure to a “subprotocol” parameter indicating which protocol the client expects to exchange data via the channel. This “subprotocol” parameter value is registered for WebRTC data channels as per [8] Section 9.1, or equivalently [8], Section 5.1. If the “subprotocol” parameter is not present, then its value defaults to an empty string.

For the interaction metadata specified in MeCAR, the subprotocol could be defined as “3gpp-timedmetadata”, and the message format would be binary. The specific syntax and semantics of the interaction metadata are indicated by the 32-bits *subprotocol payload ID* field in the User Data of the SCTP packet.

**Figure 6** is an example of an SDP offer that adds 1 timed metadata data channel stream (stream id=110).

|  |
| --- |
| **SDP offer** |
| m=application 52718 UDP/DTLS/SCTP webrtc-datachannel  b=AS:500  a=sctp-port:5002  a=max-message-size:1024  a=fingerprint:SHA-1 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB  a=tls-id: abc3de65cddef001be82  a=dcmap:110 subprotocol="3gpp-timedmetadata" |

**Figure 6 Example SDP offer with timed metadata signalling**

Figure 7 is an example considering further down the syntax and semantics of the generic candidate fields introduced in the Figure 5 for the case of an OpenXR XR\_EXT\_hand\_tracking metadata:

* 32-bits *subprotocol payload ID* is set to an *OpenXR ID value*;
* 16-bits metadata type field is set to 52 which is the registered extension number for XR\_EXT\_hand\_tracking in OpenXR;
* 16-bits metadata attributes field may indicate the metadata attributes such as time synchronization and reliability.
  + OpenXR defines a 64-bit signed integer representing nanoseconds (XrTime) and associates it to the APIs. Yet, SCTP does not carry timestamp information. For the payload protocol that does not define timestamp, a timestamp field may be added to the chunk user data section when a metadata timestamp attribute bit, T, is set to 1 as below.
* OpenXR (XrTime) 64-bit signed timestamp in nanoseconds.

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

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Type = 0 | Reserved|U|B|E| Length |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| TSN |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Stream Identifier S = 110 | Stream Sequence Number n |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Payload Protocol Identifier = 53 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| subprotocol payload ID = OpenXR ID value |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| Metadata type = 52 | Reserved |T|

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| |

| timestamp |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

/ XrSystemHandTrackingPropertiesEXT /

\ \

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 7 Example of OpenXR XR\_EXT\_hand\_tracking metadata payload format

## Security considerations

The interaction class real-time metadata can contain sensitive information tracking the interactions of an end user, e.g., elements of pose, tracking information of palm, hand, or face, as well as controller inputs. Therefore, the integrity and confidentiality of metadata in transit is in some scenarios, depending on the application requirements, necessary.

The transport of the real-time interaction class metadata over the WebRTC data channel and in particular over SCTP as of clause 2.2 ensures by default, [4], integrity and confidentiality given the underlying DTLS over UDP secure transport.

# Proposal

It is requested to include clause 2 of this document in the permanent document and take them into account in the discussion of IRTCW and related topics.

# References

1. 3GPP TSG SA WG4 S4-221150, “MeCAR Permanent Document v3.0”, August 2022
2. 3GPP TSG SA WG4 S4-221209, “5G\_RTP Permanent Document v0.0.2”, August 2022
3. IETF RFC4566, “SDP: Session Description Protocol”, July 2008
4. IETF RFC8831, "WebRTC Data Channels", January 2021
5. IETF RFC4960, “Stream Control Transmission Protocol”, Sept. 2007
6. IETF RFC8837, “Differentiated Services Code Point (DSCP) Packet Markings for WebRTC QoS"
7. IETF RFC 8864, “Negotiation Data Channels Using the Session Description Protocol (SDP)”, 2021-01
8. RFC 8832, “WebRTC Data Channel Establishment Protocol”, 2021-01
9. 3GPP TS26.928, “Extended Reality (XR) in 5G”, 2022-04