**Agenda item:** 10.6

**Source:** Qualcomm Inc.

**Title: [FS\_5G\_RTP\_Ph2] RTP Transport of XR Metadata**

**Document for** Discussion andAgreement

# Introduction

In this contribution, we address the key issue on the RTP transport of XR metadata. We introduce the MPEG activity on the codec-independent carriage of system metadata and propose to collect and register 3GPP-defined metadata with MPEG.

# Issues with RTP-based synchronization

Real-time Transport Protocol (RTP) provides fundamental mechanisms for delivering time-sensitive media over IP networks. However, achieving frame-accurate synchronization between multiple RTP streams presents significant technical challenges that go beyond the protocol's basic capabilities.

Each RTP stream operates with its own timing domain, characterized by independent sequence numbers and timestamps based on local clocks. The complexity is compounded by different timestamp rates for different media types, such as 90kHz for video and 48kHz for audio. This independence introduces several clock-related complications: oscillator frequencies vary between devices, clocks drift over time, timing can be affected by temperature variations, and streams often begin with non-identical offsets.

Network conditions introduce multiple sources of timing uncertainty that significantly impact synchronization accuracy. Variable packet latency creates unpredictable delays, while network jitter affects packet arrival times in ways that are difficult to compensate for. Packet loss may require retransmission or interpolation, further complicating timing relationships. Different streams might traverse different network paths, encountering varying queue delays in network equipment and experiencing different levels of last-mile network congestion.

RTP relies on RTCP Sender Reports (SR) for inter-stream synchronization. In this mechanism, the sender transmits regular RTP media packets interleaved with periodic RTCP SR packets. Each SR packet contains an NTP timestamp providing a wall-clock reference, an RTP timestamp serving as a media clock reference, and packet and octet counts for stream management.

A screenshot of a computer program

Description automatically generated

The temporal resolution of RTCP SR synchronization presents significant challenges. SR packets are typically sent infrequently, often every 1-5 seconds, creating long intervals between timing references. This infrequency necessitates interpolation between SR packets and makes it impossible to capture rapid timing variations. Additionally, timestamp precision is constrained by several factors: Although 64-bit NTP timestamps allow for picosecond accuracy, NTP timestamps rarely exceed microsecond precision. In addition, conversion between timing domains introduces rounding errors, and there are often non-integer relationships between different media clock rates.

Extended Reality (XR) applications demand frame accurate synchronization between the metadata and its corresponding media data. This is very challenging to achieve when the two are sent over separate RTP streams. The embedded delivery of the metadata with its media data becomes essential.

# MPEG Codec-Independent System Metadata

Note: the MPEG codec-independent system metadata is still in exploration stage. No specification is available yet or expected in the near future.

The codec-independent carriage of system metadata activity in MPEG addresses the necessity of embedding metadata alongside media streams in a manner that retains its association with corresponding media samples. System metadata, which is distinct from the codec-dependent data of video or audio content, provides critical contextual information that can enhance media processing, adaptation, and playback experiences across different platforms and devices. This approach ensures that metadata remains accessible without reliance on specific codecs, allowing greater flexibility for cross-platform interoperability and media stream adaptation.

In a codec-independent metadata carriage system, the metadata is associated with specific media samples during the encoding process. This association persists through the decoding process, where the metadata is extracted and passed along with the corresponding decoded media sample. Figure 1 below illustrates the end-to-end process from embedding to extraction.

A diagram of a system

Description automatically generated

In this architecture, the application attaches metadata to a media sample, which is then fed into a media encoder. The encoder integrates the metadata into the encoded bitstream, embedding it alongside the media content in a manner that remains codec-agnostic. When decoding, the media decoder detects the embedded metadata, extracting it along with the decoded sample. This metadata is subsequently delivered to the application layer for further use.

The carriage of system metadata is expected to adhere to the following requirements:

1. **Standardized Metadata Structure**: A universally understood format for metadata is essential to maintain consistency across platforms. The metadata structure generally includes an identifier for each metadata type, followed by a payload whose structure is governed by the metadata identifier. Commonly supported structures include integer-based, URN-based, and UUID-based payloads, each supporting unique use cases.
2. **Elementary Stream Support**: Metadata is carried either as ITU T-35 messages within an elementary stream, or through some other means. For ITU T-35 messages, which contain three parts (country code, terminal provider code, and terminal provider-oriented code), specific values are defined and registered to ensure uniformity.
3. **Metadata Processing APIs**: To enable effective metadata handling, APIs are needed for both insertion and extraction. These APIs allow applications to interact with metadata at the elementary stream level, ensuring that metadata is seamlessly inserted into, and extracted from, media streams without disrupting the media content flow.

A centralized metadata registry, managed by the MPEG Systems group, ensures consistent and interoperable metadata definitions. This registry publishes the unique identifiers of the system metadata, with appropriate pointers to the external specifications that define them.

# Proposal

We propose to document the content of section 2 and 3 in the PD or TR and to initiate a dialogue with MPEG to ensure that the codec-independent system metadata carriage is a suitable solution for the XR services and enablers defined by SA4. We should also collect relevant system metadata, in addition to the rendered pose for split rendering, and share these with MPEG. Eventually, this metadata should be registered once an MPEG registry is established.