

MOQ  
Internet-Draft  
Intended status: Standards Track  
Expires: 1 September 2025

X. de Foy  
InterDigital  
R. Krishna

T. Jiang  
China Mobile  
28 February 2025

MoQ relays for Support of High-Throughput Low-Latency Traffic in 5G  
draft-defoy-moq-relay-network-handling-03

## Abstract

This document describes a mechanism to convey information about media frames. The information is used for specific handling in functions such as error recovery and congestion handling. These functions can be critical to improve energy efficiency and network capacity in some (especially wireless) networks. Due to end-to-end encryption, MoQ relays are expected to extract the metadata required by these functions. This document aims to enable a level of wireless network support for MoQ that is equivalent to what is possible for RTP.

## 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 1 September 2025.

## Copyright Notice

Copyright (c) 2025 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 . . . . .	2
1.1. Techniques used by Wireless Networks for XR Traffic Handling . . . . .	3
1.2. Identifying XR Metadata in RTP Media Flows . . . . .	3
1.3. Identifying XR Metadata in MOQ flows . . . . .	5
1.4. Terms and Definitions . . . . .	5
1.5. Notational Conventions . . . . .	5
2. XR Metadata in MOQ Transport . . . . .	6
2.1. Signalling of XR Metadata Support . . . . .	6
2.2. XR Metadata in MOQT Datagrams or Streams . . . . .	7
3. Endpoint Behavior for Communicating XR Metadata . . . . .	8
3.1. Endpoint Behavior . . . . .	8
3.2. MoQ relay Behavior . . . . .	8
4. IANA considerations . . . . .	9
5. Security Considerations . . . . .	9
6. Acknowledgments . . . . .	9
7. References . . . . .	9
7.1. Normative References . . . . .	9
7.2. Informative References . . . . .	10
Appendix A. XR RTP Header Extension for XR Traffic Handling in 5G Networks . . . . .	10
A.1. Release 18 XR Metadata . . . . .	10
A.2. Release 19 XR Metadata . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

Wireless networks can be a challenging environment for applications with high-throughput and low latency requirements, such as video conferencing and Extended Reality (XR, presented for example in [RFC9699]). Wireless networks implement techniques to improve network capacity and energy efficiency, as well as reduce the impact of packet losses on users' quality of experience (Section 1.1). An extension to the RTP protocol [TS26.522] has been defined, which enables metadata associated with application data units to be identified at the ingress point of the wireless network (Section 1.2). To enable a similar operation with the MoQ protocol [I-D.draft-ietf-moq-transport], this document describes how a MoQ relay can be used at the ingress point of the wireless network (Section 1.3).

The rest of this document is structured as follows:

- \* Section 2 describes XR metadata for MoQ.
- \* Section 3 describes the behavior of the MoQ relay and of MOQ endpoints.
- \* Section 4 describes IANA registrations.
- \* Section 5 describes applicable security considerations.

### 1.1. Techniques used by Wireless Networks for XR Traffic Handling

The network can handle groups of packets based on how critical they are to the user's experience. Some groups of data packets hold a unit of information generated at the application level, which we will designate as an `_application data unit_`, and which can be for example a video frame, or video slice. Application data units are typically handled (e.g., decoded) together by the application. 3GPP defines the term `_PDU set_` to identify these groups of data packets [TS23.501], which can correspond to the data packets of an application layer data unit. The handling of application data units by the application can depend on other application data units (e.g., in the case of decoding dependency). The wireless network performs differentiated handling of groups of data packets. For example, it prioritizes some groups over others in case of congestion. In congestion situations, the network can also selectively drop data packets that depend on already lost data packets. Furthermore, the network can limit the amount of time that the radio needs to stay awake to transmit and receive data. To achieve this this, the scheduler can use information on the size and periodicity of traffic, as well as delay budget and maximum tolerable jitter specific to the application.

### 1.2. Identifying XR Metadata in RTP Media Flows

To perform differentiated handling of PDU sets, a network node (i.e., a User Plane Function, UPF) at the ingress point of a wireless network identifies PDU set metadata from a downlink media flow. 3GPP has developed an RTP header extension for XR traffic for this purpose (see Appendix A for details). When receiving a downlink packet, the UPF reads the XR metadata fields from the RTP header extension and transmits them to the RAN node in the GTP header that encapsulates the packet. The RAN node uses the XR metadata information to implement the differentiated handling described in Section 1.1.

3GPP defines metadata used for XR traffic handling when using RTP:

- \* In the 3GPP release 18 (RTP header extension for PDU set marking, [TS26.522])
  - The `_PDU Set Importance_` (`*PSI*`, 4 bits) is the importance of this PDU Set compared to other PDU Sets within the same Multimedia Session.
  - The `_end PDU of PDU set_` (`*E*`, 1 bit) indicates the last PDU of the PDU set.
  - The `_end of data burst_` (`*D*`, 1 bit) indicates the last PDU of a data burst.
  - The `_PDU set sequence number_` (`*PSSN*`, 10 bits) identifies the PDU set. The PSSN is incremented monotonically by 1 for each subsequent PDU set.
  - The `_PDU Sequence Number within a PDU Set_` (`*PSN*`, 6 bits) identifies a PDU with the PDU set, starting from 0 and incremented monotonically for every PDU in the PDU set in the order of transmission from the sender.
  - The `_PDU set size_` (`*PSSize*`, 24 bits) is an optional field which indicates the total size, in bytes, of all PDUs of the PDU Set (including IP header and IP payload).
  - The `_number of PDUs in the PDU Set_` (`*NPDS*`, 16 bits) is an optional field which indicates the number of PDUs in the same PDU set.
- \* In the 3GPP release 19 (work in progress, documented in [TS23.501])
  - The `_Expedited Transfer Indication_` (`*ETI*`) indicates whether this PDU should be forwarded with an expedited QoS level.
  - The `_Burst Size_` (`*BSize*`) describes the size of a burst of traffic, which includes one or more PDU sets. [23.501] defines a data burst as a set of multiple PDUs generated and sent by the application in a short period of time.
  - The `_Time to Next Burst_` (`*TTNB*`) describes the time between the end of the present burst and the beginning of the next burst.

The optional header extension fields are included subjects to an SDP signaling offer/answer negotiation.

### 1.3. Identifying XR Metadata in MOQ flows

For XR media traffic transported over the MoQ protocol, the UPF cannot access XR metadata unless it is exposed to the UPF in some fashion. This document describes how the UPF can act as, or communicate with, a MoQ relay to obtain XR metadata associated with media data. To enable this behavior, it is also necessary for the media sender to identify application data units that correspond to different desired traffic handling (e.g., different layers within a media frame), and provide associated metadata. Figure 1 describes a UPF with MoQ relay functionality, identifying XR metadata and transmitting it to access nodes, for example, for a media stream sent by B towards A and C. For privacy and security, it is desirable that the MoQ relay, which can be operated by a network or service operator, does not have access to media data. For interoperability, it is also desirable for these mechanisms to not be codec specific.

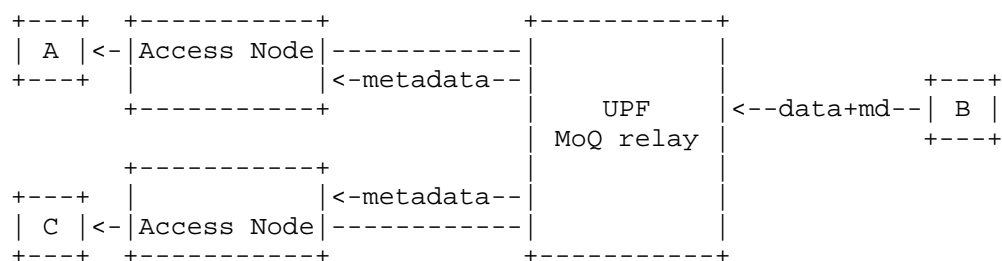


Figure 1: XR Traffic Handling by Access Networks using a MoO relay.

### 1.4. Terms and Definitions

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.

### 1.5. Notational Conventions

This document uses the conventions detailed in ([RFC9000], Section 1.3) when describing the binary encoding. See [I-D.draft-ietf-moq-transport], section 1.3 for a non normative summary of the syntax.

## 2. XR Metadata in MOQ Transport

In MOQ Transport (MOQT), XR metadata is transmitted in object headers, using the extension header mechanism described in [I-D.draft-ietf-moq-transport]. This document describes additional metadata in the MOQT protocol, corresponding to XR metadata identified in Release 18 and 19 of 3GPP.

### 2.1. Signalling of XR Metadata Support

This document registers with IANA two new integer MoQ Extension Header types named `_Release 18 XR Metadata Extension_`, and `_Release 19 XR Metadata Extension_`. These MoQ Extension Header types are used to transmit XR metadata in object headers.

This document registers with IANA a new MOQT setup parameter `_EXT-XR-METADATA_`. The `EXT-XR-METADATA` parameter can optionally be exchanged by endpoints to indicate their support for specific sets of XR metadata. Alternatively, the endpoints may determine whether to communicate XR metadata, and which metadata to communicate, based on an out-of-band agreement.

```
EXT-XR-METADATA {  
    Type (i) = <TBD value registered for EXT-XR-METADATA>,  
    Extension-List (i),  
}
```

Figure 2: `EXT-XR-METADATA` MOQT setup parameter

`_Extension-List_` is a variable-length integer (as defined in [RFC9000]) containing a bit field, with zero or more bits being set to 1 among:

- \* 0x01, which indicates the Release 18 XR Metadata Extension,
- \* 0x02, which indicates the use of optional PSSize field in the Release 18 XR Metadata Extension,
- \* 0x04, which indicates the use of optional NPDS field in the Release 18 XR Metadata Extension,
- \* 0x08, which indicates the Release 19 XR Metadata Extension,
- \* 0x10, which indicates the use of optional Burst Size field in the Release 19 XR Metadata Extension,
- \* 0x20, which indicates the use of optional Time-to-Next-Burst field in the Release 19 XR Metadata Extension,

The client can include an EXT-XR-METADATA parameter in CLIENT\_SETUP, to indicate that it supports receiving certain extensions and options.

The server can include an EXT-XR-METADATA parameter in SERVER\_SETUP, to indicate that it supports receiving certain extensions and options.

The EXT-XR-METADATA parameter is advisory in nature, and its purpose is both to inform the media sender that including per-object XR metadata can be useful, and to enable the media sender to save processing time by only including the fields that are supported by the receiver.

For example, a client may indicate that it supports receiving any metadata and optional metadata with an EXT-XR-METADATA Extension-List value of (0x3f), and the server may reply with a SERVER\_SETUP that does not include EXT-XR-METADATA, to indicate that it does not intend to use any per-object XR metadata in objects it receives from the client (or, e.g., because the connection is intended for unidirectional streams from server to client only).

## 2.2. XR Metadata in MOQT Datagrams or Streams

When sending MOQ objects, an endpoint can include one of the extension headers described in this section. When receiving MOQ objects, an endpoint can ignore any header extension or optional field that it does not support or does not wish to process.

A media sender MAY include a \_Release 18 Metadata Extension Header\_ in an object. If the sender includes the optional PSSize field, it MUST set PSSize\_present to 1, else it sets PSSize\_present to 0. If the sender includes the optional NPDS field, it MUST set NPDS\_present to 1, else it sets NPDS\_present to 0.

```
Release 18 XR Metadata Extension Header {
  Header Type (i) = extension type value TBD,
  Header Length (i),
  E (1),
  D (1),
  PSSize_present (1),
  NPDS_present (1),
  PSI (4),
  PSSN (10),
  PSN (6),
  [PSSize (i)],
  [NPDS (i),]
}
```

Figure 3: XR Metadata MOQT Extension Header

A media sender MAY include a `_Release 19 Metadata Extension Header_` in an object. If the sender includes the optional `BSize` field, it MUST set `BSize_present` to 1, else it sets `BSize_present` to 0. If the sender includes the optional `TTNB` field, it MUST set `TTNB_present` to 1, else it sets `TTNB_present` to 0.

```
Release 19 XR Metadata Extension Header {  
    Header Type (i) = extension type value TBD,  
    Header Length (i),  
    ETI (1),  
    BSize_present (1),  
    TTNB_present (1),  
    Reserved (5),  
    [BSize (i)],  
    [TTNB (i),]  
}
```

Figure 4: XR Metadata MOQT Extension Header

### 3. Endpoint Behavior for Communicating XR Metadata

#### 3.1. Endpoint Behavior

The endpoints can either have an out-of-band or in-band agreement to use per-object XR metadata in some tracks in a MOQT connection. To establish an in-band agreement, the endpoint can exchange the setup parameter `EXT-XR-METADATA` including a value indicating the content of the XR metadata extensions and optional fields that the endpoint support receiving, as described in Section 2.1.

An endpoint can send objects including XR metadata extension header(s) in the object header, which the receiver can parse to obtain XR metadata associated with the object, as described in Section 2.2.

#### 3.2. MoQ relay Behavior

To handle high-throughput low-latency traffic, a MoQ client (e.g., running on a mobile device) should set up a MOQT connection through a MoQ relay interworking with a wireless network and providing this functionality. Discovery of such relay is out of scope of this document.

The MoQ relay establishes a connection with a MoQ server and may send the setup parameter `EXT-XR-METADATA` in `CLIENT_SETUP`, as described in Section 2.1.



The MoQ relay, during the media delivery session, parses XR metadata associated with the downlink media objects, as described in Section 2.2.

The MoQ relay transmits XR metadata along with IP packets containing the MOQT objects, to the radio access network, which uses the metadata to apply XR traffic handling to the IP packets, and conduct corresponding RAN resource scheduling and mobile device control.

#### 4. IANA considerations

This document will register 2 odd-numbered MOQT header extensions: a Release-18 XR Metadata Extension, and a Release-19 XR Metadata Extension.

This document will register a MOQT setup parameter: EXT-XR-PARAMETER.

#### 5. Security Considerations

To enable support for the feature described in this document, the application exposes metadata to a MoQ relay under the control of a network or service operator. End-to-end encrypted media is not exposed to the MoQ relay, so this is not seen as a high-risk exposure.

#### 6. Acknowledgments

Thanks to Srinivas Gudumasu, who was a contributor to the first revision of this draft. Thanks to Jaya Rao, Ghyslain Pelletier, John Kaippallimalil, Sri Gundavelli, Hang Shi, Mike Starsinic and Hyunsik Yang for discussions and comments about this draft.

#### 7. References

##### 7.1. Normative References

- [I-D.draft-ietf-moq-transport]  
Curley, L., Pugin, K., Nandakumar, S., Vasiliev, V., and I. Swett, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-08, 12 February 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-08>>.
- [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>>.

- [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>>.

## 7.2. Informative References

- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/rfc/rfc9000>>.
- [RFC9699] Krishna, R. and A. Rahman, "Use Case for an Extended Reality Application on Edge Computing Infrastructure", RFC 9699, DOI 10.17487/RFC9699, December 2024, <<https://www.rfc-editor.org/rfc/rfc9699>>.
- [TS23.501] 3GPP, "System architecture for the 5G System", 3GPP, 2024, <[www.3gpp.org/dynareport/23501.htm](http://www.3gpp.org/dynareport/23501.htm)>.
- [TS26.522] 3GPP, "5G Real-time Media Transport Protocol Configurations", 3GPP, 2024, <[www.3gpp.org/dynareport/26522.htm](http://www.3gpp.org/dynareport/26522.htm)>.

## Appendix A. XR RTP Header Extension for XR Traffic Handling in 5G Networks

3GPP defined an RTP header extensions for PDU set marking, in [TS26.522], which enables a media sender to indicate media related information in each RTP packet.

### A.1. Release 18 XR Metadata

The fields defined in the Release 18 of 3GPP are included in this appendix, for the reader's convenience (text copied from [TS26.522] V18.1.0 with minor adaptations and omissions of some notes for readability):

- \* \*End PDU of the PDU Set [E]\* (1 bit): This field is a flag that shall be set to 1 for the last PDU of the PDU Set and set to 0 for all other PDUs of the PDU Set.
- \* \*End of Data Burst [D]\* (1 bit): This field is a flag that shall be set to 1 for the last PDU of a Data Burst. It shall be set to 0 for all other PDUs. A Data Burst may consist of one or more PDU Sets.

- \* \*PDU Set Importance [PSI]\* (4 bits): The PDU Set Importance field indicates the importance of this PDU Set compared to other PDU Sets within the same Multimedia Session. This information may help RAN to discard PDUs, when needed. Lower values shall indicate a higher importance PDU Set, with the highest importance PDU Set indicated by 1 and the lowest importance PDU Set indicated by 15. When the RTP sender cannot define an importance, it shall set the value to 0.
- \* \*PDU Set Sequence Number [PSSN]\* (10 bits): The sequence number of the PDU Set to which the current PDU belongs, acting as a 10-bit numerical identifier for the PDU Set. The PSSN shall be incremented monotonically by 1 for each subsequent PDU Set.

NOTE: This value wraps around at 1023, however, using the 16-bit RTP packet sequence number and PSSN pair, a receiver may uniquely distinguish between any PDU Sets.

- \* \*PDU Sequence Number within a PDU Set [PSN]\* (6 bits): The sequence number of the current PDU within the PDU Set. The PSN shall be set to 0 for the first PDU in the PDU Set and incremented monotonically for every PDU in the PDU Set in the order of transmission from the sender.

NOTE: A receiver may use the RTP packet sequence number together with the PSN to distinguish between PDUs within a PDU Set that contains more than 64 PDUs.

- \* \*PDU Set Size [PSSize]\* (24 bits): The PDU Set Size indicates the total size of all PDUs of the PDU Set to which this PDU belongs. This field is optional and subject to an SDP signaling offer/answer negotiation, where the RTP sender shall indicate whether it will provide the size of the PDU Set for that RTP stream. If not enabled, the field shall not be present within the RTP HE. If enabled, but the RTP sender is not able to determine the PDU Set Size for a particular PDU Set, it shall set the value to 0 in all PDUs of that PDU Set. The PSSize shall indicate the size of a PDU Set including RTP/UDP/IP header encapsulation overhead of its corresponding PDUs. The PSSize shall be expressed in bytes. It is recommended to add the PDU Set Size field when the Number of PDUs in the PDU Set field is present.

NOTE: This field may be optionally present given the signaling of the "pdu-set-size" extension attribute in the SDP offer/answer negotiation.

- \* \*Number of PDUs in the PDU Set [NPDS]\* (16 bits): The number of PDUs within the PDU Set indicates the total number of PDUs belonging to the same PDU Set. This field is optional and subject to an SDP signaling offer/answer negotiation, where the RTP sender may indicate whether it will provide the number of PDUs within the PDU Set for that RTP stream. If enabled, but the RTP sender is not able to determine the Number of PDUs in the PDU Set, it shall set the value to 0 in all PDUs of that PDU Set. It is recommended to add the Number of PDUs in the PDU Set field when the PDU Set Size field is present.

NOTE: This field may be optionally present given the signaling of the "num-pdus-in-pdu-set" extension attribute in the SDP offer/answer negotiation.

#### A.2. Release 19 XR Metadata

Additional fields are added to media related information in the Release 19 of 3GPP (text based on [TS23.501]):

- \* \*Expedited Transfer Indication [ETI]\*: this field indicates whether this object should be forwarded with an expedited QoS level.
- \* \*Burst size [BSize]\*: this field describes the size of a burst of traffic, which includes one or more PDU sets.
- \* \*Time-to-next-burst [TTNB]\*: this field describes the time between the end of the present burst and the beginning of the next burst.

#### Authors' Addresses

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

Renan Krishna  
United Kingdom  
Email: renan.krishna@gmail.com

Tianji Jiang  
China Mobile  
China  
Email: tianjijiang@chinamobile.com