



IEEE Standard for Smart Media Transport

IEEE Computer Society

Developed by the
Data Compression Standards Committee

IEEE Std 1857.12™-2022

STANDARDS

IEEE Standard for Smart Media Transport

Developed by the

Data Compression Standards Committee
of the
IEEE Computer Society

Approved 08 November 2022

IEEE SA Standards Board

Abstract: An IP-based media protocol over both broadcasting networks and broadband networks, including data modeling, data transmission methods, signaling messages, and media presentation mechanisms is defined in this standard.

Keywords: adaptive FEC, common encapsulation, data modeling, IEEE 1857.12™, media presentation, smart media transport

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 4 April 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9296-6 STD25852
Print: ISBN 978-1-5044-9297-3 STDPD25852

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this IEEE standard was completed, the Audio Video Coding Working Group had the following membership:

Siwei Ma, *Chair*
Cliff Reader, *Vice Chair*

The officers of 1857.12 Sub-Working Group and the participants for this standard are as follows:

Yiling Xu, *Chair*
Luntian Mou, *Vice Chair*
Cheng Huang, *Vice Chair*
Xiaoxu Luan, *Secretary*

Yongjian Chang
Gao Chen
Hao Chen
Yongqin Chen
Wen Gao
Zongming Guo
Lizhi Hou
Puyue Hou

Tiejun Huang
Wei Huang
Ying Hu
Zhiqian Jiang
Li Liu
Shiqiang Long
Yue Wu
Shaowei Xie

Jianwei Xu
Zhiyao Yang
Haiwu Zhao
Haiyin Zhao
Weimin Zhang
Wenjun Zhang
Xinggong Zhang
Wenjie Zhu

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Oscar Au
William Byrd
Hao Chen
Werner Hoelzl
Puyue Hou
Ying Hu
Cheng Huang

Piotr Karocki
Fan Liang
Dong Liu
Xiaoxu Luan
Luntian Mou
Bo Sun
Yaowei Wang
Lisa Ward

Shaowei Xie
Yiling Xu
Lu Yu
Yu Yuan
Oren Yuen
Haiwu Zhao
Wenjie Zhu

When the IEEE SA Standards Board approved this standard on 08 November 2022, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Ramy Ahmed Fathy
J. Travis Griffith
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick

Johnny Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Andrew Myles
Damir Novosel
Annette D. Reilly
Robby Robson
Jon Walter Rosdahl

Mark Siira
Dorothy V. Stanley
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Philip B. Winston
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1857.12-2022, IEEE Standard for Smart Media Transport.

The smart media transport (SMT) protocol is a packet-switched oriented application layer protocol designed to provide transport services for media data including timed media data (e.g., video, audio), non-timed media data (e.g., text, images), and user feedback data (e.g., real-time commands) over heterogeneous networks. Heterogeneous networks are hybrid networks consisting of unidirectional physical networks (e.g., digital broadcast networks) and bidirectional physical networks (e.g., Internet Protocol [IP] networks). The SMT protocol is based on the IP protocol and can dynamically adapt to different network conditions in heterogeneous networks.

The SMT protocol can be logically divided into three logical functional areas; namely, the encapsulation functional area, the transmission functional area, and the signaling functional area, see Figure 1.

This document provides the following parts about the usage of SMT:

- The logical data model of SMT protocol for flexible organization and dynamic configuration of services
- The application layer transport protocol for packetized delivery and SMT payload format
- The signaling message formats for delivery and consumption of SMP packets
- Media presentation model based on SMT protocol
- An application of SMT in HTTP/2 live streaming
- Adaptive Forward Error Correction (FEC) coding mechanism for reliable delivery under SMT protocol.

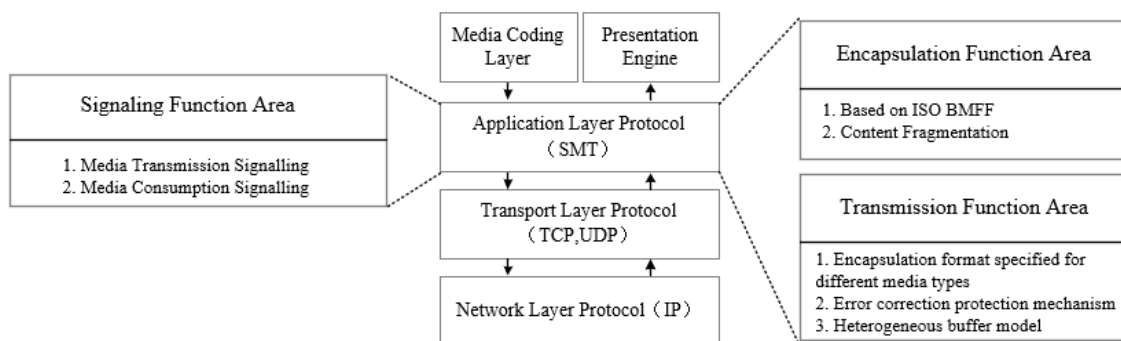


Figure 1—Architecture of SMT

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Word usage	11
2. Normative references.....	12
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations	13
4. Conventions.....	14
4.1 Overview	14
4.2 Arithmetic operators	14
4.3 Logical operators	14
4.4 Relational operators	15
4.5 Bit-wise operators	15
4.6 Assignment operators	16
4.7 Mathematical functions	16
5. SMT data model	17
5.1 General	17
5.2 Package.....	18
5.3 Asset	18
5.4 Common Encapsulation Unit.....	19
6. Data transport	24
6.1 General	24
6.2 Transport model.....	25
6.3 Transport protocol	26
6.4 SMTP payload	31
6.5 Payload structure-based transport operations	35
7. Signaling.....	37
7.1 General	37
7.2 Signaling message format.....	38
7.3 Signaling messages for Package consumption	39
7.4 Asset descriptors.....	48
7.5 Syntax element group	53
7.6 ID value	59
7.7 Resource Request/Response Message	61
7.8 Interaction feedback message	63
7.9 Session control message	65
7.10 Device capability information	67
7.11 Clock relationship information messages	70
7.12 Synchronization request messages.....	72
7.13 Synchronization response messages	73
7.14 Asset delivery characteristic messages	74
8. Media Presentation	78
8.1 General	78

8.2 Presentation Model.....	78
8.3 Media presentation signaling.....	79
9. HTTP/2 Live Streaming	86
9.1 General	86
9.2 System architecture.....	87
9.3 Data type.....	88
9.4 Push strategy.....	91
9.5 HTTP/2 protocol binding.....	93
9.6 Message definition.....	94
10. Adaptive FEC coding	97
10.1 General	97
10.2 FEC coding mechanism.....	98
10.3 Encoding symbol block format.....	103
10.4 FEC source and repair packet format	105
Annex A (normative) Hypothetical receiver buffer model.....	109
A.1 General.....	109
A.2 FEC decoding buffer.....	109
A.3 De-jitter buffer	110
A.4 SMTP packet decapsulation buffer.....	110
A.5 Usage of hypothetical receiver buffer model.....	111
A.6 Estimation of end-to-end delay and buffer requirement	111
A.7 Syntax	111
A.8 Semantics.....	112
Annex B (normative) FEC code	113
B.1 FEC coding algorithm.....	113
B.2 Adaptive FEC code	113

IEEE Standard for Smart Media Transport

1. Overview

1.1 Scope

This standard defines an Internet Protocol (IP)-based media protocol over both broadcasting networks and broadband networks, including data modeling, data transmission methods, signaling messages, and media presentation mechanisms.

1.2 Purpose

The purpose of this standard is to define smart multimedia transport protocols for use in heterogeneous networks (including live streaming, video on demand over broadcast networks, broadband networks, and mobile networks), enabling media to be efficiently and seamlessly transported between entities in the content development and distribution industries.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{1,2}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

¹ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

² The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1003.1™-2008, IEEE Standard for Information Technology - Portable Operating System Interface (POSIX®).^{3,4}

IETF RFC 1738, Uniform Resource Locators (URL).⁵

IETF RFC 2141, Uniform Resource Names (URN) Syntax.

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax.

IETF RFC 4122, A Universally Unique Identifier (UUID) URN Namespace.

IETF RFC 6570, Uniform Resource Identifier (URI): Template.

IETF RFC 7230, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing.

IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content.

ISO/IEC 14496-12, Information technology – Coding of audio-visual objects – Part 12: ISO base media file format (ISO BMFF).^{6,7}

ISO/IEC 23009-1, Dynamic adaptive streaming over HTTP (DASH).

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁸

access unit: Smallest media data entity to which timing information can be attributed.

asset: Any multimedia data entity that is associated with a unique identifier and that is used for building a multimedia presentation.

³ IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., (<http://standards.ieee.org/>).

⁴ The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

⁵ IETF publications are available from the Internet Engineering Task Force (<https://www.ietf.org/>)

⁶ ISO publications are available from the International Organization for Standardization (<https://www.iso.org/>) and the American National Standards Institute (<https://www.ansi.org/>).

⁷ IEC publications are available from the International Electrotechnical Commission (<https://www.iec.ch>) and the American National Standards Institute (<https://www.ansi.org/>).

⁸ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

common encapsulation unit: Generic container for decodable timing or non-timing data that is independent of the media codec.

media content: Information such as audio, video, and subtitles with the same time reference contained in the media data.

non-timed data: Media data that does not have an inherent timeline for the decoding and/or presenting of its media content.

package: Logical collection of media data, delivered using SMT.

packet: Formatted unit of the media data, delivered using SMT.

packet flow: Sequence of packets with same sending entity and receiving entity.

payload: Formatted unit of media data to carry Packages and/or signaling messages using either the SMT protocol, or an Internet application layer transport protocol.

receiving entity: Entity that receives and consumes media data.

sending entity: Entity that sends media data to one or more receiving entities.

Smart Media Transport protocol: Application layer transport protocol used to transport payloads over IP networks.

timed data: Data that has inherent timeline information for the decoding and/or presentation of its media contents.

3.2 Acronyms and abbreviations

AL-FEC	Application Layer Forward Error Correction
AU	Access Unit
CEU	Common Encapsulation Unit
CRI	Clock Relation Information
DASH	Dynamic Adaptive Streaming over HTTP
DCI	Device Capability Information
FEC	Forward Error Correction
HRBM	Hypothetical Receiver Buffer Model
HTTP	Hypertext Transfer Protocol
ISI	Inter Symbol Interference
ISO BMFF	ISO Base Media File Format
MIME	Multipurpose Internet Mail Extensions
MFU	Media Fragment Unit
MP	Media Presentation
MPD	Media Presentation Description
MPT	Media Presentation Table
NTP	Network Time Protocol
PA	Package Access
PID	Packet Identifier
QoS	Quality of Service
RAP	Random Access Point
RTP	Real-time Transport Protocol
SMT	Smart Media Transport
SMTP	SMT Protocol

TS	Transport Stream
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
XML	Extensible Mark-up Language

4. Conventions

4.1 Overview

The mathematical operators used in this document are similar to those used in the C programming language. However, integer division and arithmetic shift operations are specifically defined. Unless otherwise stated, numbering and counting conventions generally begin from zero.

4.2 Arithmetic operators

Arithmetic operators are defined in Table 1.

Table 1—Definition of arithmetic operators

Arithmetic operator	Definition
+	Addition
−	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	Multiplication, including matrix multiplication
ab	Exponentiation. Specifies a to the power of b . In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	Integer division with truncation of the result toward zero. For example, $7/4$ and $-7/-4$ are truncated to 1, and $-7/4$ and $7/-4$ are truncated to -1 .
÷	Division in mathematical equations where no truncation or rounding is intended.
$\frac{a}{b}$	Division in mathematical equations where no truncation or rounding is intended.
$\sum_{i=a}^b f(i)$	Summation of $f(i)$, with i taking all integer values from a up to and including b .
$a \% b$	Modulus. Remainder of a divided by b , defined only for integers a and b with $a \geq 0$ and $b > 0$.

4.3 Logical operators

Logical operators are defined in Table 2.

Table 2—Definition of logical operators

Logical operator	Definition
$a \&\& b$	Boolean logical “and” of a and b.
$a \parallel b$	Boolean logical “or” of a and b.
!	Boolean logical “not”.
$a ? b : c$	If a is TRUE or not equal to 0, evaluates to the value of b; otherwise, evaluates to the value of c.

4.4 Relational operators

Relational operators are defined in Table 3.

Table 3—Definition of relational operators

Relation operator	Definition
$>$	Greater than.
$\geq$	Greater than or equal to.
$<$	Less than.
$\leq$	Less than or equal to.
$=$	Equal to.
$\neq$	Not equal to.

4.5 Bit-wise operators

Bit-wise operators are defined in Table 4.

Table 4—Definition of bit-wise operators

Bit-wise operator	Definition
~	bit-wise “not”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
&	bit-wise “and”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
	bit-wise “or”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
^	bit-wise “exclusive or”. When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.
a >> b	arithmetic right shift of a two's complement integer representation of a by b binary digits. This function is defined only for non-negative integer values of b. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of a prior to the shift operation.
a << b	arithmetic left shift of a two's complement integer representation of a by b binary digits. This function is defined only for non-negative integer values of b. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

4.6 Assignment operators

Assignment operators are defined in Table 5.

Table 5—Definition of assignment operators

Assignment operator	Definition
=	assignment operator.
++	increment, i.e., x++ is equivalent to x = x + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation.
--	decrement, i.e., x-- is equivalent to x = x - 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation.
+=	increment by amount specified, i.e., x += 3 is equivalent to x = x + 3, and x += (-3) is equivalent to x = x + (-3).
-=	decrement by amount specified, i.e., x -= 3 is equivalent to x = x - 3, and x -= (-3) is equivalent to x = x - (-3).

4.7 Mathematical functions

The definitions of mathematical functions are shown as follows.

$$\text{Abs}(x) = \begin{cases} x & ; \ x \geq 0 \\ -x & ; \ x < 0 \end{cases}$$

Ceil(x) the smallest integer greater than or equal to x.

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases}$$

Floor(x) the largest integer less than or equal to x .

Log2(x) the base-2 logarithm of x .

Log10(x) the base-10 logarithm of x .

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

Mean(A) the arithmetic average value of the input array A .

Median(A) the median value of the input array A .

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

Round(x) = Sign(x) * Floor(Abs(x) + 0.5)

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x = 0 \\ -1 & ; \quad x < 0 \end{cases}$$

Sort(A) the sorted version of the input array A in increasing order.

Sqrt(x) the square root of x .

5. SMT data model

5.1 General

This clause introduces the logical data model of the smart media transport (SMT) protocol. The SMT protocol supports both streaming delivery and download delivery of coded media data. The SMT protocol supports protected data model during delivery with signaling messages. To meet the requirements of flexible organization of SMT content, the association relationship of content in SMT services is specified by the description file. Only updating the description file is needed when the service changes, instead of changing data stream layers. In order to adapt to different network environments, separate media transmission management information is required. In heterogeneous networks, transmission management information can be dynamically updated according to changes in the network environment. Separating signaling messages from media data helps ensure dynamic flexibility in content self-organization and provides better cross-layer optimization.

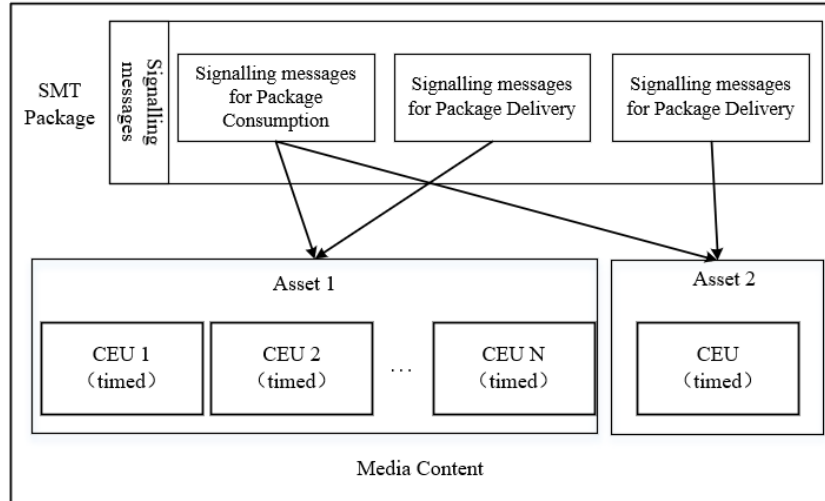


Figure 2—Data model

5.2 Package

As shown in Figure 2, a Package is a logical entity. The SMT package is a logical entity and can be considered as a service, which mainly consists of signaling description file (see 8 for details) and media content. The signaling file includes forward signaling and feedback signaling, which can be divided into two types: signaling messages for package consumption and signaling messages for package delivery. The signaling messages for consumption mainly contains the description information of the service, such as the composition, storage location, type, and presentation policy of the media content. The signaling messages for transmission mainly contains the management information in the transmission process, such as quality of service (QoS) parameters and buffer setting information, etc. Media content can also be divided into two types: one is timed media, such as video and audio content; the other is non-timed media, such as text and picture information. In order to support the effective transmission of media content under heterogeneous network conditions and the dynamic configuration of content during transmission, a generic encapsulation unit for SMT media content is designed, which shall be fragmented, self-contained, and unified, thus supporting the needs of dynamic organization of content and dynamic adaptation of transmission.

5.3 Asset

An asset is any multimedia data to be used for building a multimedia presentation. An Asset is a logical grouping of content fragments named common encapsulation units (CEUs) that share the same Asset ID for carrying encoded media data. Encoded media data of an Asset can be either timed data or non-timed data. Timed data are encoded media data that has an inherent timeline and shall require synchronized decoding and presentation of the data units at a designated time. By contrast, non-timed data does not have an inherent timeline for decoding and presenting of its media content. That is, the decoding time and presentation time of each item in non-timed data shall not necessarily be interdependent with the other items of same non-timed data.

Two CEUs of the same Asset carrying timed media data shall have no overlap in their presentation time.

The media data type of a media Asset can be audio, video, or web pages, etc. The Edit_list describes the mapping of multimedia files to different versions of associated content and indicates the identification number of the media data unit contained in the Edit_list, as shown in Figure 3. Different Edit_list levels of the same media Asset can contain different media data units, or they can contain the same media data unit.

Edit_list requires signaling support, see the MUR_descriptor in the media presentation (MP) table for the relevant signaling.

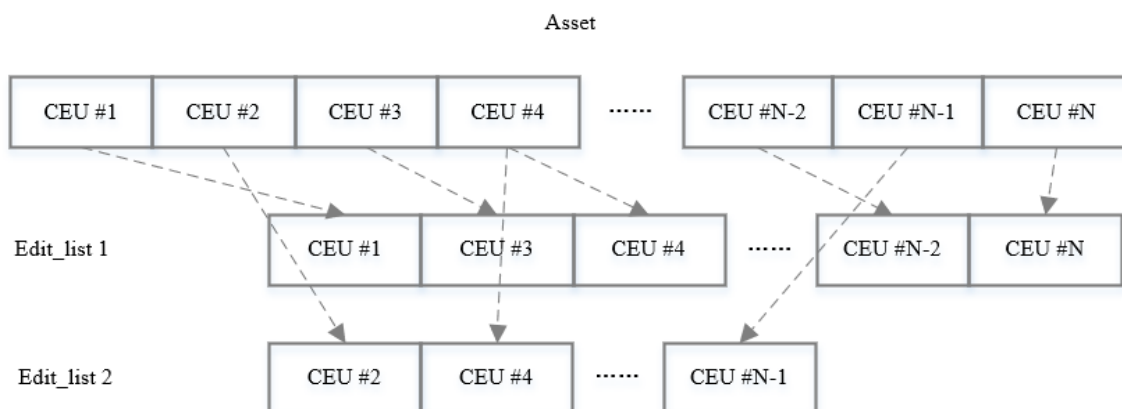


Figure 3—Data composition of media Assets

5.4 Common Encapsulation Unit

5.4.1 General

A Common Encapsulation Unit is a conformant ISO base media file format (ISO BMFF) file generated according to rules in 5.4.2. Asset ID, the sequence number of the CEU and related information are provided in the 'cceu' box to uniquely identify the media data encapsulated in the CEU file. The 'moov' box contains all codec configuration information for decoding and presentation of the media data.

Timed media data are stored as track of the ISO BMFF (a single media tracks is allowed in CEU). Non-timed media are stored as part of the metadata in an ISO BMFF. Figure 4 depicts two examples of SMT encapsulation, one for timed and the other for non-timed media. For packetized delivery of CEU, an SMT hint track provides the information to convert encapsulated CEUs to SMT Protocol (SMTP) payloads and SMTP packets.

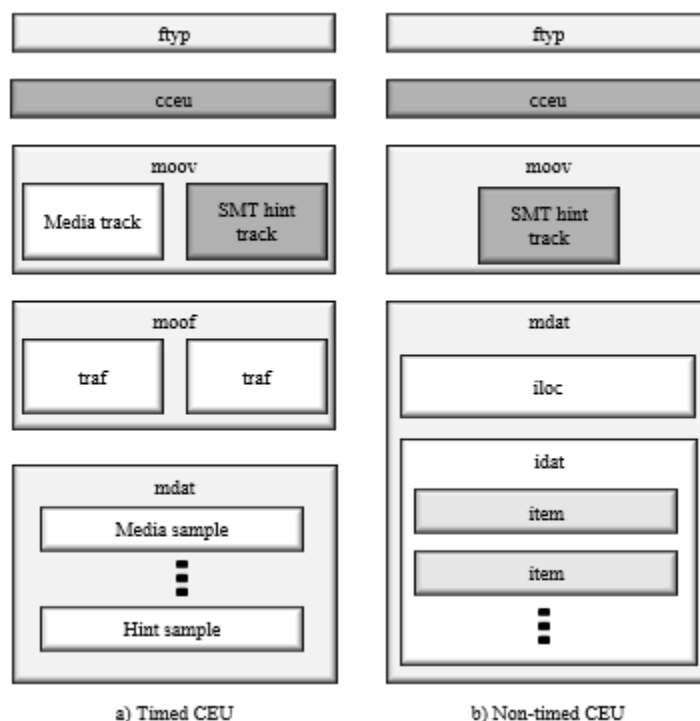


Figure 4—CEU package structure

5.4.2 CEU brand definition

The brand 'ceuf' (CEU file) defined in this part identifies files that conform to the encapsulation rules for CEU. The 'ceuf' brand requires the support of the 'isom' brand. Support to the other brands, such as 'dash' brand (defined in ISO/IEC 23009-1⁹), can also be indicated separately.

A CEU file is composed of a set of metadata boxes that enables the CEU to be self-contained. A CEU file shall contain an 'ftyp', 'cceu', 'moov' box, and shall optionally contain a 'sidx' box, all of which are part of the CEU metadata. Other boxes are allowed but will be ignored if the parser does not recognize them.

The 'moov' box shall contain at most one media track and shall contain SMT hint tracks that identify the smallest media fragment units in the transport format. The tracks in the 'moov' box shall contain no samples to help ensure small overhead (i.e., the entry_count in the 'stts', 'stsc' and 'stco' boxes shall be set to 0). The 'mvex' box shall be contained in the 'moov' box for the file storing a CEU with timed media data to indicate that the movie fragment structure is used. The 'mvex' box also sets the default values for the tracks and samples of the following 'moof' boxes.

Additionally, a 'cceu' box shall occur at the file level and the following rules shall be applied, including orders of the boxes:

- If present, the 'cceu' box shall be placed right after the 'ftyp' box.
- For timed media data, zero or more 'sidx' boxes shall be present in the file. If present, they shall index the 'moof' boxes that build the current CEU.

In addition to the box orders, the following restrictions shall also be applied to the 'ceuf' brand.

⁹ Information on references can be found in Clause 2.

- a) The maximum number of independent (e.g., an empty 'tref' box) media tracks in this file shall be one. Additionally, tracks with non-empty 'tref' box (e.g., hint tracks) shall be available.
- b) For timed media data, the file shall have at least one 'moof' box.
- c) For non-timed media data, one 'meta' box shall be present at the file level and shall contain non-timed media items of the CEU.
- d) If present, an Edit List Box ('elst') shall only provide an initial offset.
- e) Runs of sample data shall be placed in the 'mdat' box, in decoding order and without any other data between them.
- f) Any sample auxiliary data, as described by 'saio' and 'saiz', shall be placed at the beginning of the 'mdat' boxes, before any sample data.
- g) Any hint data shall be placed after sample data in the 'mdat' (or in another mdat placed after the sample data) so that the sample offsets are not changed before and after transmission.

The 'tfdt' box shall be present inside the 'traf' box of each 'moof' box to provide the decode time of the first sample of the movie fragment in decoding order.

If any 'elst' box is available, the indicated offset shall be applied to the composition time of the first sample in presentation order of the CEU in addition to the presentation time provided by any presentation information.

Timed media data are stored as a track of the ISO BMFF and indexed by the 'moov' and 'moof' boxes in the fully backward-compatible manner. An SMT hint track guides the SMT sending entity in converting the encapsulated CEUs into a packetized media stream to be delivered using a transport protocol such as the SMT protocol.

Non-timed media data are stored as metadata items that are described by a 'meta' box. The 'meta' box shall appear at the file level. Each file of the non-timed media data shall be stored as a separate item in the 'meta' box. The entry point to the non-timed media shall be marked as the primary item of the 'meta' box (see ISO/IEC 14496-12).

5.4.3 CEU box

5.4.3.1 General

- Box type: 'ccue'
- Container: File
- Mandatory: Yes
- Number: One or more

The CEU ('ccue') box contains the Asset identifier of the Asset to which the current CEU belongs and other attribute information for the current CEU. The Asset identifier is used to uniquely identify the Asset globally. The CEU information includes the sequence number of the CEU in the Asset and the related attribute information.

5.4.3.2 Syntax

```
aligned(8) class CEUBox
extends FullBox('cceu', version, 0){
    unsigned int(1) is_complete;
    unsigned int(7) reserved;
    unsigned int(32) ceu_sequence_number;
    AssetIdentifierBox();
}

aligned(8) class AssetIdentifierBox {
    unsigned int(32) asset_id_scheme;
    unsigned int(32) asset_id_length;
    unsigned int(8) asset_id_value[asset_id_length];
}
```

5.4.3.3 Semantics

is_complete: indicates whether the CEU contains all the Media Fragment Units (MFUs) described in the MFU structure or not.

ceu_sequence_number: the sequence number of the current CEU. For the first CEU in a media asset (Asset), the sequence number shall be '0', and it is incremented by '1' for each following CEU. This sequence number is unique within an Asset.

asset_id_scheme: identifies the scheme of Asset ID and determines the type of **asset_id_value**. Valid schemes are listed in Table 6.

Table 6—Values of asset_id_scheme

Value	Description
"UUID"	UUID (Uniform Unique Identifier), see IETF RFC 4122
"URI"	URI (Uniform Resource Identifier), see IETF RFC 3986

asset_id_length: the length of **asset_id_value**.

asset_id_value: the identifier of the Asset. Format of the value depends on the type of the **asset_id_scheme** field.

5.4.4 SMT hint track

5.4.4.1 General

For transportation purposes, an SMT hint track provides an SMT sending entity with hints for the fragmentation of a CEU into the smallest fragment units of the media in transport format. The smallest fragment unit is media-aware and is used to build the SMTP payload, i.e., media data in the CEU is extracted and put on an SMTP payload by the SMT sending entity at transport time.

A SMT hint track provides an SMT sending entity with hints for fragmenting CEU into the smallest media fragment units in transport format. CEU metadata shall contain 'ftyp', 'sidc', 'cceu', and 'moov' boxes.

5.4.4.2 Hint track sample properties

5.4.4.2.1 Syntax

```
aligned(8) class SMTHintSampleEntry() extends
SampleEntry('smth') {
    unsigned int(16) hinttrackversion = 1;
    unsigned int(16) highestcompatibleversion = 1;
    unsigned int(16) packet_id;
    unsigned int(1) is_fragment;
    unsigned int(1) is_timed;
    unsigned int(6) reserved;
}
```

5.4.4.2.2 Semantics

`packet_id`: is a unique identifier for the Asset for which this hint track applies.

`is_fragment`: is a flag indicating whether the CEUs are fragmented into MFUs or not. If this flag is set to '0', the hint track is applied to the complete CEU, i.e., each track fragment corresponds to one media fragment; otherwise, each hint sample is applied to a single MFU.

`is_timed`: indicates whether the media data hinted by this track is timed data or non-timed data.

5.4.4.3 Hint track sample format

5.4.4.3.1 Definition

Each media sample is assigned to one or more MFUs. Each sample of the SMT hint track generates one or more MFUs.

5.4.4.3.2 General

```
aligned(8) class SMTHSample {
    unsigned int(32) sequence_number;
    if (is_timed) {
        signed int(8) trackrefindex;
        unsigned int(32) movie_fragment_sequence_number;
        unsigned int(32) samplenummer;
        unsigned int(8) priority;
        unsigned int(8) dependency_counter;
        unsigned int(32) offset;
        unsigned int(32) length;
    }
    else{
        unsigned int(16) item_ID;
    }
}
```

5.4.4.3.3 Semantic

`sequence_number`: an integer number indicating the sequence order of this MFU within the CEU. Discontinuity of sequence numbers in a CEU is allowed to indicate that certain MFUs (whose sequence number was not in the sequence) were not processed after packetization of CEU. Examples of MFU processing are delivery and caching by underlying network entity.

`trackrefindex`: the ID of the media track from which the MFU data is extracted.
`movie_fragment_sequence_number`: the sequence number of the movie fragment to which the media data of this MFU belongs.

`samplenum`: the number of the sample from which this MFU is extracted. Sample number n points to the n -th sample from accumulated samples of the current movie fragment. The sample number of the first sample in the movie sample is set to '1'.

`priority`: indicates the priority of the MFU relative to other MFUs within a CEU.

`dependency_counter`: indicates the number of MFUs whose decoding is dependent on this MFU. The value of this field is equal to the number of subsequent MFUs in the order of `sequence_number` that shall not be correctly decoded. For example, if the value of this field is equal to n , then n subsequent MFUs that shall not be correctly decoded without this MFU.

`offset`: the offset of the media data contained in this MFU. The offset base is the beginning of the containing 'mdat' box. MFU shall be placed at the position that offset indicates.

`length`: the length of the data corresponding to this MFU in bytes.

`item_ID`: for non-timed media data, this is the ID of the item that is contained in this MFU.

6. Data transport

6.1 General

This clause specifies an application layer transport protocol for packetized delivery of Packages and defines the SMT payload format. The SMT payload format is defined as a generic payload format for the packetization of media data components of a Package. It is agnostic to media codecs used for encoded media data, so that any type of media data that are encapsulated as a CEU can be packetized into SMT, which supports the application layer transport protocol for media data streaming. SMT payloads can be used for Real-time Transport Protocol (RTP), SMT protocol, and other transport protocols. SMT payload can also be used to packetize signaling messages. The SMT protocol is an application layer transport protocol supporting delivery of Packages over heterogeneous packet-switched delivery networks, including IP-based network environments. The SMT protocol provides the necessary features for delivery of Packages, such as protocol-level multiplexing that enables different Assets to be delivered over a single SMT packet flow, and delivery timing model independent of presentation time to adapt to a wide range of network jitters.

6.2 Transport model

The SMT transport model can solve various problems, such as how to help ensure reliable data transmission under network channels with different QoS parameters, how to design the terminal buffer model to synchronize the presentation of multi-source media data under heterogeneous network conditions, and how to provide differentiated protection mechanisms based on data importance levels when network conditions are bad.

To support SMT transport requirements, the transport model in Figure 5 is proposed. SMT signaling files and transport packets are organized according to media service logical packets, and media organization and transmission are related to media services, which is conducive to the adaptation of media content organization and transmission according to the differentiation of different service types.

SMT logical Packages can be serialized as SMT files to support media file-style storage, transmission, and download; they can also be packaged as SMT transport Packages to support streaming of media. Due to the high correlation in content between file format and transport packet format, SMT supports easy conversion of both for relay service, while the transport model can efficiently respond to dynamic configuration of service content. The signaling files such as service description and transmission control can be transmitted in different modes from the media content, which can meet the out-of-band transmission requirements and provide a higher level of protection measures for the signaling files.

The fragmented, self-contained feature of the media CEU also determines that the SMT transport model can cope with packet error problems during transmission, achieve resilient error tolerance and error hiding, and support fast error recovery at the receiving entity. Each CEU and its MFUs are uniquely identified, and the header information is combined with signaling messages to specify different levels of parameter settings such as data range, priority, QoS requirements, protection methods, and presentation time, which are important references for hierarchical, heterogeneous, multi-channel, and reliable transmission of media content.

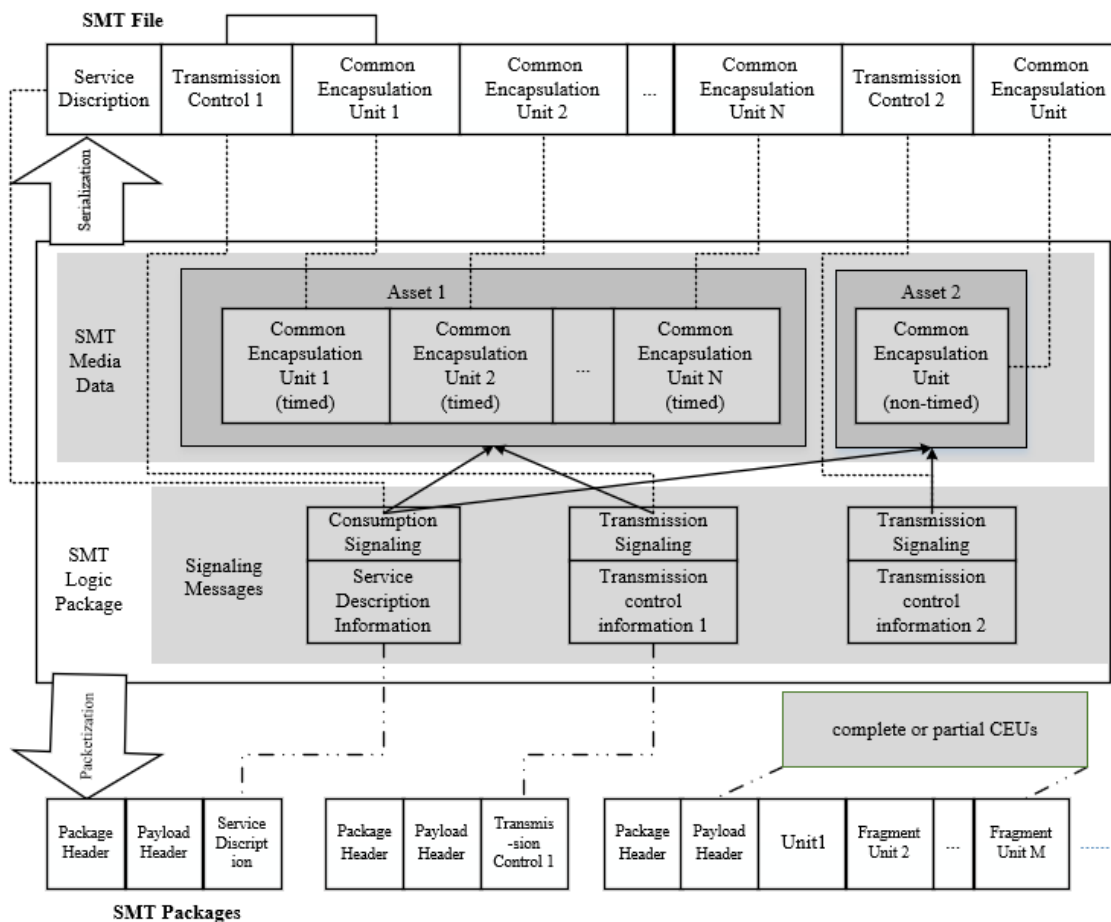


Figure 5—Transport model

6.3 Transport protocol

6.3.1 Minimum fragmentation unit based on encapsulation structure

The structural relationship between MFU and CEU is shown in Figure 6 and Figure 7. An MFU containing timed media is a media sample or subsample, and each movie fragment consisting of a 'moof' box and a 'mdat' box can contain one or more media samples. An MFU containing non-timed media is an item.

Each MFU is composed of a header and the associated media data. The MFU header shall be a copy of the MFU hint sample, and the media data shall be a copy of the media data that is referenced by that MFU hint sample. The extraction and reconstruction of an MFU is instructed by the SMT hint track, see 5.4.4.

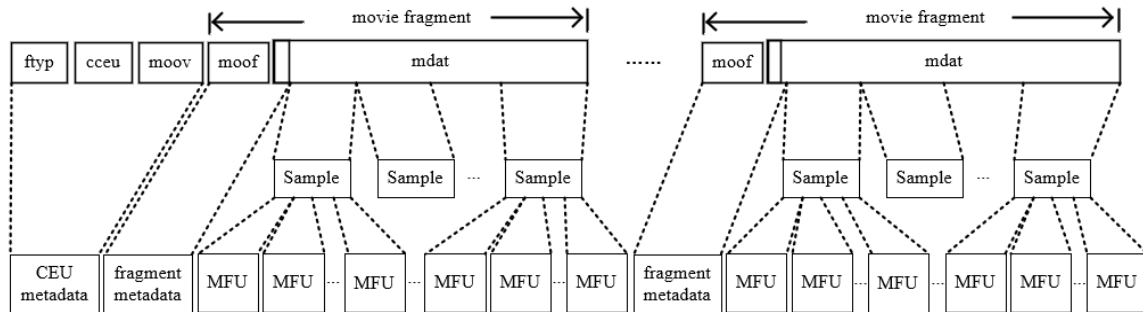


Figure 6—Relationship between CEU and MFU in timed media

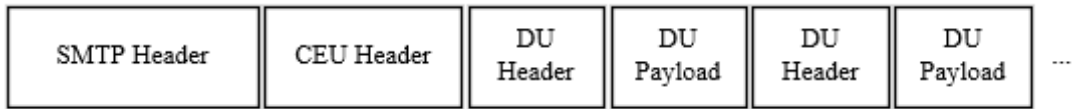


Figure 7—Relationship between CEU and MFU in non-timed media

6.3.2 SMT package

6.3.2.1 General

The SMT protocol is an application layer transport protocol that is designed to efficiently and reliably transport Packages. SMT protocol can be used for both timed and non-timed media data. The protocol supports several enhanced features, such as media multiplexing and network jitter calculation. These features are designed to deliver content composed of various types of encoded media data more efficiently. The SMT protocol can run on top of existing network protocols, such as user datagram protocol (UDP) and IP.

The SMT protocol supports the multiplexing of different media data such as multiple CEUs from various Assets over a single SMTP packet flow. It delivers multiple types of data in the order of consumption to the receiving entity to help synchronization between different types of media data without introducing a large delay or requiring large buffer. The SMT protocol also supports the multiplexing of media data and signaling messages within a single packet flow. A single SMTP payload shall be carried in only one SMTP packet. Fragmentation and aggregation are only provided by the payload format and not by SMTP itself.

SMT protocol also provides means to calculate and remove jitter introduced by the underlying delivery network, so that constant end-to-end packet delivery delay can be achieved. By using the timestamp field in the packet header, jitter can be accurately calculated without requiring any additional signaling information and protocols.

6.3.2.2 SMT package structure

Figure 8 illustrates the structure of the SMTP packet at V=0.

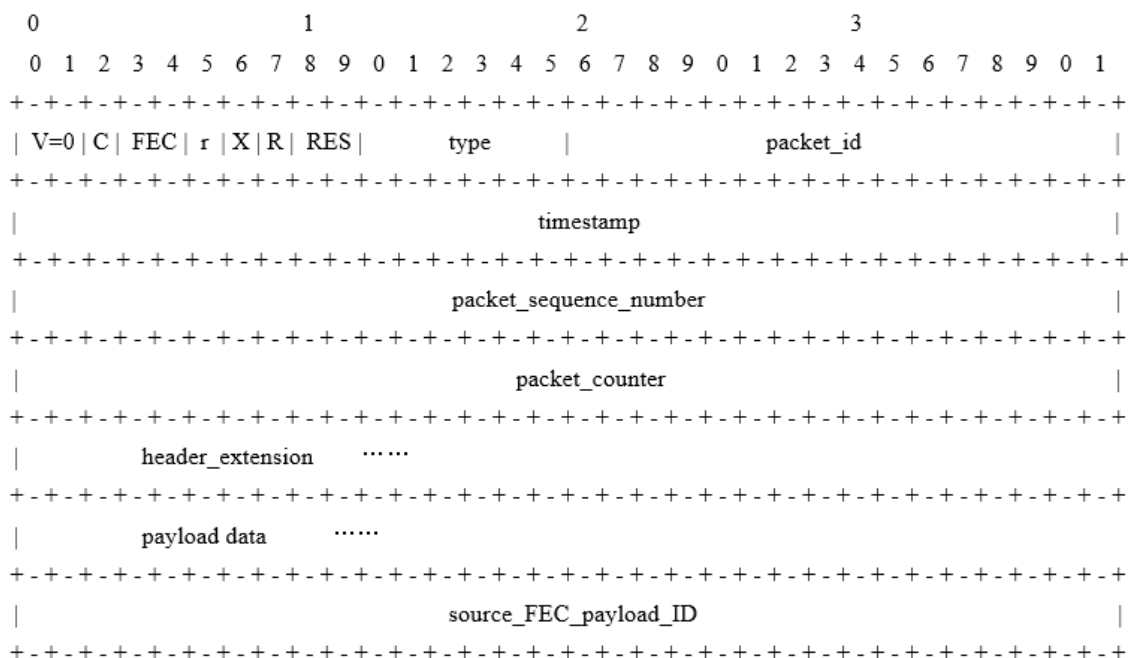


Figure 8—Package structure (V=0)

Figure 9 illustrates the structure of SMTP packets at V=1. This version defines a simplified SMTP packet structure for flexible and diverse message interactions.

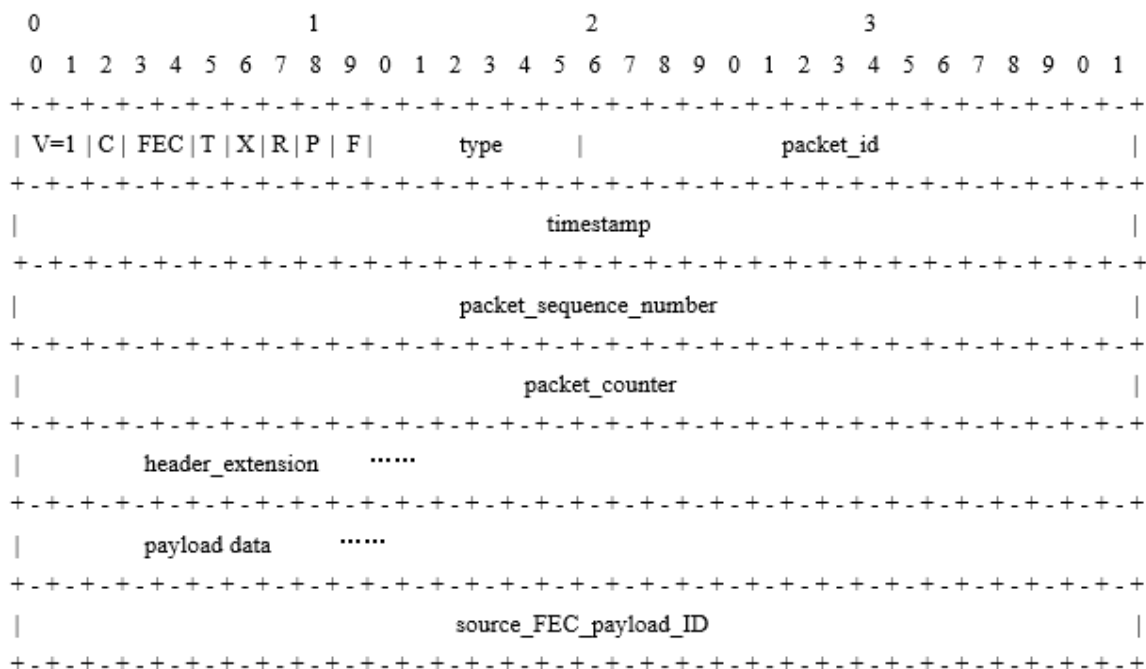


Figure 9—Package structure (V=1)

6.3.2.3 Semantics

version (V: 2 bits): indicates the version number of the SMT protocol.

`packet_counter_flag` (C: 1 bit): “1” in this field indicates that the `packet_counter` field is present.

`FEC_type` (FEC: 2 bits): indicates the type of the Forward Error Correction (FEC) scheme used for error protection of SMTP packets. Valid values of this field are listed in Table 7.

Table 7—FEC_type

Value	Description
0	SMTP Packages without Application Layer Forward Error Correction (AL-FEC) protection
1	SMTP packets with AL-FEC protection (FEC packets)
2	SMTP package with repair symbol (FEC repair package)
3	Reserved for future use

`reserved` (v=0, r: 1 bit): reserved for future use.

`timestamp_flag` (v=1, T: 1 bit): when set to “1”, the `timestamp` field is present.

`extension_flag` (X: 1 bit): when set to “1”, this flag indicates that the `header_extension` field is present.

`RAP_flag` (R: 1 bit): When this field is set to '1', it indicates that the payload contains a random access point (RAP) to the data stream of that data type. The exact semantics of this flag are defined by the data type itself. The `RAP_flag` field needs to be set to '1' when the data unit type is CEU metadata or media fragment metadata. The `RAP_flag` field needs to be set to '1' when the data unit type is an MFU containing synchronization information in a timed CEU or a major item containing synchronization information in a non-timed CEU.

`reserved` (v=0, RES:2 bit): reserved for future use.

`packet_id_flag` (v=1, P:1 bit): when set to “1”, the `packet_id` field is present.

`fragmentation_flag` (v=1, F:1 bit): when set to “1”, the `packet_sequence_number` field is present.

`type` (6bits): identifies the payload data type. The values of the payload type are shown in Table 8.

Table 8—Data type and data unit definition

Value	Data Type	Data unit definition
0x00	CEU	Media-aware fragment of the CEU
0x01	Signaling messages	One or more signaling messages or fragments of a signaling messages
0x02~0x03	Reserved for other data types	
0x04 ~ 0x3F	Private Use Reservation	

packet_id (16bits): this field is an integer value used to distinguish one Asset from another. The value of this field is derived from the **asset_id** field of the Asset to which the package belongs. The mapping between **packet_id** and **asset_id** is identified by the MP table in a signaling message. Separate value is assigned to signaling messages and FEC repair flows. The **packet_id** value is unique throughout the lifetime of the delivery session and for all SMT flows delivered by the same SMT sending entity. For AL-FEC, the mapping between **packet_id** and FEC repair flows is provided in the AL-FEC message.

timestamp (32bits): specifies the time instance of SMTP packet delivery based on Coordinated Universal Time (UTC). The format is the “short-format” as defined in IETF RFC 5905, Network Time Protocol (NTP) version 4, Clause 6.¹⁰ This timestamp specifies the sending time at the first byte of SMTP packet. It is required that an SMT sending entity shall provide accurate time information that are synchronized with UTC.

packet_sequence_number (32bits): an integer value used to distinguish packets that have the same **packet_id**. The value of this field starts from arbitrary value and is incremented by one for each SMTP packet received. It wraps around to “0” after the maximum value is reached.

packet_counter (32bits): an integer value for counting SMTP packets. It is incremented by 1 when an SMTP packet is sent regardless of its **packet_id** value. This field starts from arbitrary value and wraps around to “0” after its maximum value is reached.

header_extension: The header extension mechanism is provided to allow for proprietary extensions to the payload format to enable applications and media types that require additional information to be carried in the payload format header. The header extension mechanism is designed in such a way that it shall be discarded without impacting the correct processing of the SMTP payload. The header extension shall have the format as shown in Figure 10. The protocol does not specify any particular packet header extension.

Source_FEC_payload_ID (32bits): this field shall be used only when the 'FEC_type' field is set to '1'. When 'FEC_type' field is set to '1', SMTP packets are protected with AL-FEC, and this field shall be added to SMTP packets after using AL-FEC protection.

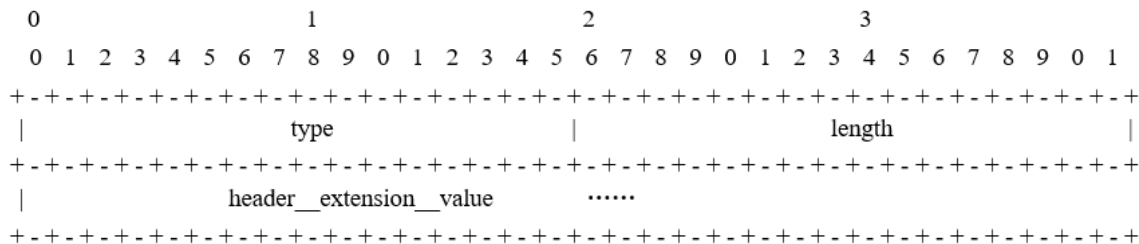


Figure 10—Header extension structure

type (16bits): unique identification of the following header extension.

length (16bits): indicates the length of the **header_extension_value** field (in byte).

header_extension_value: provides the extension information. The protocol does not define the format of this field.

¹⁰ IETF publications are available from the Internet Engineering Task Force., (<https://www.ietf.org/standards/>), (<https://datatracker.ietf.org/>)

6.3.3 SMTP session description information

SMTP session description information shall be delivered to the receiving entity in different ways to accommodate different deployment environments. Before a receiver is able to join an SMTP session, the receiver needs to obtain the following information:

- a) The destination information. In an IP environment, the destination IP address and port number
- b) An indication that the session is an SMTP session
- c) The version number of the SMT protocol used in the SMTP session

6.4 SMTP payload

6.4.1 General

SMTP payload is a generic payload to packetize and carry SMT media data. SMTP payload can be one or more complete CEUs or CEU fragments, or signaling messages, etc. For each payload type, a single data unit for delivery as well as a type specific payload header are defined. For example, fragment of a CEU is considered as a single data unit when SMTP payload carries CEU fragments. The SMT protocol shall aggregate multiple data units with the same data type into a single SMTP payload. It can also fragment a single data unit into multiple SMTP packets.

6.4.2 CEU mode

6.4.2.1 General

The delivery of CEUs using the SMT protocol requires a packetization and depacketization procedure to take place at the SMT sending entity and SMT receiving entity, respectively. The packetization procedure transforms a CEU into a set of SMTP payloads that are then carried in SMTP packets. The SMTP payload format allows for fragmentation of the SMTP payload to enable the delivery of large payloads. It also allows for the aggregation of multiple SMTP payload data units into a single SMTP payload to cater for smaller data units. At the receiving entity, depacketization is performed to recover the original CEU data.

6.4.2.2 SMTP payload header for CEU mode

The payload type specific header is provided in Figure 11.

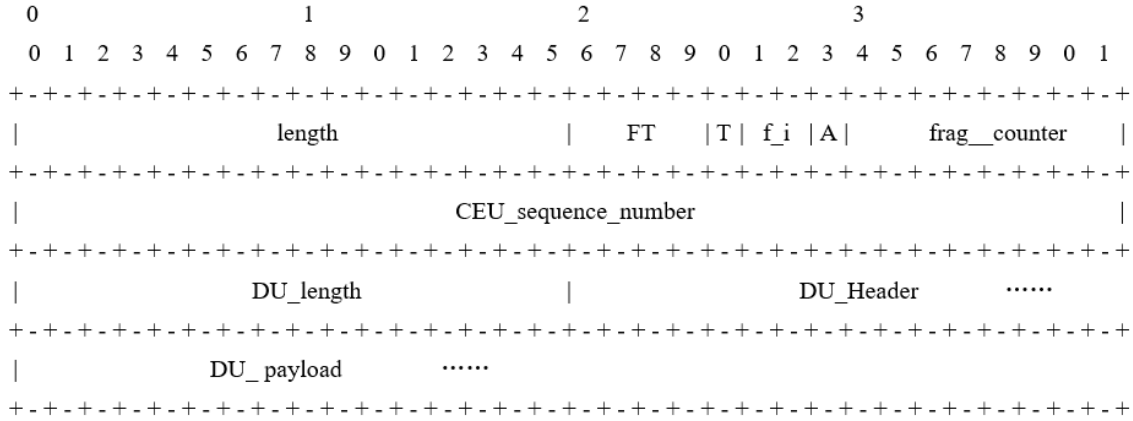


Figure 11—SMTP payload header structure in CEU mode

For payloads that carry MFUs, the data unit header is specified depending on the value of the “T” indicating whether the carried data are timed or non-timed media. For timed media, the structure is shown in Figure 12. For non-timed media, the structure of the header is shown in Figure 13.

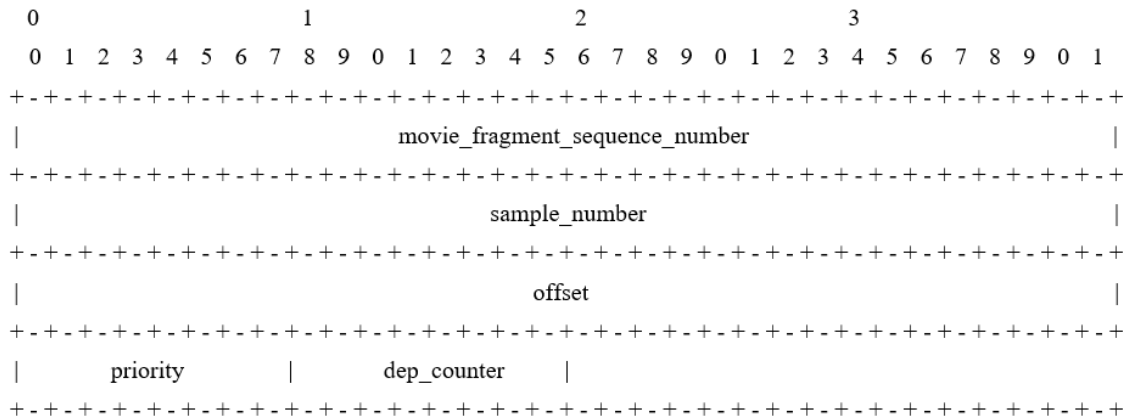


Figure 12—Timed media data unit header

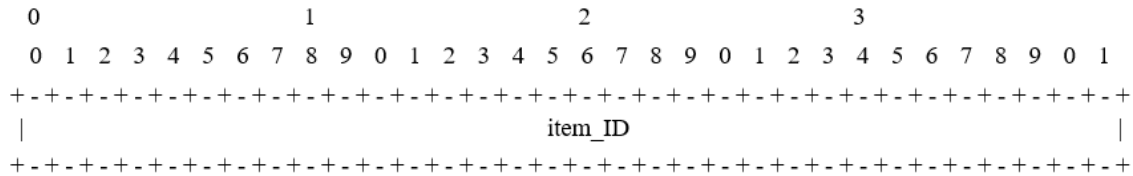


Figure 13—Non-timed media data unit header

6.4.2.3 Semantics

length (16bits): this field indicates the length of the payload excluding this field (in byte).

CEU Fragment Type (FT: 4bits): this field indicates the fragment type, and its valid values are shown in Table 9.

Table 9—Data type and data unit definition

FT	Description	Content
0	CEU metadata	contains the ftyp, cceu, moov, and other metadata boxes that appear in between.
1	Movie fragment metadata	contains the moof box and the mdat box, excluding all media data inside the mdat box, but including any chunks of auxiliary sample information.
2	MFU	contains a sample or subsample of timed media data, or an item of non-timed media data.
3 ~ 15	Private use reservation	reserved

Timed_Flag (T: 1byte): this flag indicates if the fragment is from a CEU that carries timed (value 1) or non-timed media (value 0).

Fragmentation_Indicator (f_i: 2bits): this field indicates that the fragmentation indicator contains information about fragmentation of data unit in the payload. Valid values are listed in Table 10. If the value is set to “00”, the **aggregation_flag** can be presented.

Table 10—Values of data unit indicator

Value	Description
00	The payload contains one or more complete data units.
01	The payload contains the first fragment of data unit.
10	The payload contains intermediate fragments of data unit.
11	The payload contains the last fragment of data unit.

aggregation_flag (A: 1byte): when set to “1” indicates that more than 1 data unit is present in the payload, i.e., multiple data units are aggregated.

fragment_counter (fragment_count: 8bits): this field specifies the number of SMTP payloads containing fragments of same data unit succeeding this SMTP payload. This field shall be '0' when the **aggregation_flag** field is set to '1'.

CEU_sequenece_number (32bits): the sequence number of the CEU to which the CEU fragment belongs.

DU_length (16bits): this field indicates the length of the data following this field. When **aggregation_flag** is set to '0', this field shall not be presented. When **aggregation_flag** is set to '1', this field shall appear as many times as the number of data units aggregated in the payload and preceding each aggregated data unit.

DU_header: the header of the data unit, exists only when FT is 2, i.e., MFU is valid, and the semantics corresponding to timed and non-timed media are different.

DU_payload: this field indicates data unit payload.

moive_fragment_sequence_number (32bits): the sequence number of the media fragment to which the MFU media data belongs.

`sample_number` (32bits): the sample number of the sample to which the media data of the MFU belongs.

offset (32bits): offset of the MFU media data inside the referenced sample.

priority (priority: 8bits): provides the priority of the media data carried by this MFU compared with other media data of the same CEU. The value of priority is in range of '0' to '255', with higher values indicating higher priority.

`dependency_counter` (`dep_counter`: 8bits): indicates the number of data units that depend on the MFU media data for media processing.

Item ID (32bits): the identifier of the item that is carried as part of this MFU.

6.4.3 Signaling message mode

6.4.3.1 General

The signaling message mode of SMTP is defined for the delivery of signaling messages. Signaling messages shall be encoded in one of different formats, such as binary format or Extensible Mark-up Language (XML) format. It is therefore important to enable quick access and filtering of signaling messages at the transport layer and thus to avoid parsing the signaling message itself for filtering purposes.

The signaling message payload format provides fragmentation and aggregation functionality to support efficient packetization.

6.4.3.2 SMTP payload header for signaling message mode

The structure of SMTP payload in signaling message mode is shown in Figure 14.

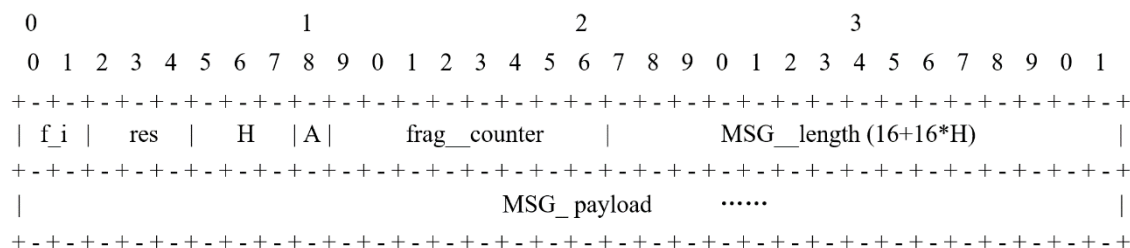


Figure 14—SMTP payload header structure in signaling message mode

6.4.3.3 Semantics

Fragmentation_Indicator (f_i: 2bits): contains information about fragmentation of a signaling message in the SMTP payload. The valid values of this field are shown in Table 11.

Table 11—Values of fragmentation indicator

Value	Description
00	The payload contains one or more complete signaling messages.
01	The payload contains the first fragment of the signaling message.
10	The payload contains neither the first, nor the last fragment of the signaling message.
11	The payload contains the last fragment of the signaling message.

RES (res: 4bits): this field contains bits that are reserved for future use and shall be set to “0”.

H (1bit): this flag indicates if an additional 16 bit is used to indicate the signaling message length.

aggregation_flag (A: 1byte): when set to “1”, indicates that more than one signaling message is present in the payload.

fragmentation_counter (frag_counter: 8bits): this field specifies the number of SMTP payloads containing fragments of the same signaling message following the current fragment. This field shall be '0' when the aggregation_flag field is set to '1'.

MSG_length (16+16*Hbits): This field indicates the length of the signaling message following this field. When aggregation_flag is set to '1', the number of times this field shall appear as many times as the number of the signaling messages aggregated in the payload and preceding each aggregated signaling message.

MSG_payload: the signaling message payload.

6.5 Payload structure-based transport operations

6.5.1 General operations

An SMTP session consists of one SMTP transport flow. SMTP transport flow is defined as all packet flows that are delivered to the same destination and which shall originate from multiple SMT sending entities. In the IP case, the destination is the IP address and port number.

A single packet shall require one or more SMTP transport flows.

A single SMTP transport flow shall deliver data from multiple Packages.

An SMTP transport flow shall carry multiple Assets. Each Asset is associated with a unique packet_id within the scope of the SMTP session. SMTP provides a streaming-optimized mode (CEU mode). The Asset is delivered as a set of related objects denoted as an object flow. Object shall either be a CEU, file or signaling message.

The SMTP packet sub-flow is the subset of the packets of an SMTP packet flow that share the same packet_id. The object flow is transported as an SMTP packet sub-flow.

CEU mode supports the packetized streaming of CEUs.

SMTP is suitable for unicast and multicast media distribution. To help ensure scalability in a multicast/broadcast environment, SMTP relies mainly on FEC rather than retransmitting packets.

Before joining the SMTP session, the SMT receiving entity needs to obtain sufficient information to enable reception of the delivered data. See 6.3.3 for the minimum required information.

SMTP requires the SMT receiving entity to be able to uniquely identify and de-multiplex SMTP packets belonging to a specific object flow.

6.5.2 Delivering CEUs

6.5.2.1 General

The CEU mode is used to transport CEUs sent by a sending entity to a receiving entity.

6.5.2.2 SMT sending entity operation

6.5.2.2.1 Timed media data

The packetization procedure takes into account the boundaries of different types of data in CEU to generate packets. The resulting packets shall carry delivery data units of either CEU metadata, movie fragment metadata, or MFU. The resulting packet shall not carry more than two different types of delivery data units. CEU metadata consists of 'ftyp' boxes, 'cceu' boxes, 'moov' boxes, and other boxes that are applied to the CEUs. The media fragment metadata consists of the 'moof' box and the 'mdat' box header (in addition to the media data). The FT field of the SMTP load carrying the media fragment metadata is set to 0x01. The media data in the 'mdat' box is divided into one or more MFUs. The FT field of the SMTP payload carrying the MFUs is set to 0x02.

Figure 15 depicts the payload generation for timed media.

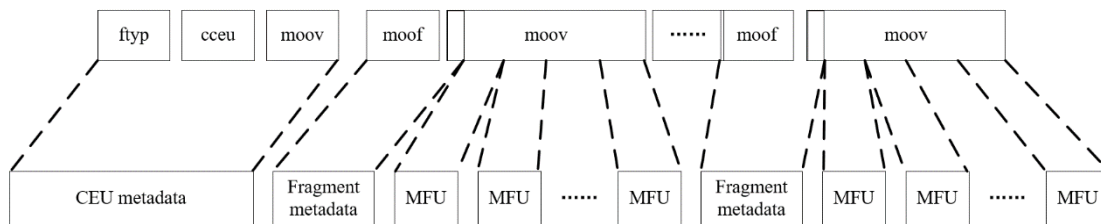


Figure 15—Payload generation for timed media

6.5.2.2.2 Non-timed media data

For non-timed media data, the packets' structure shall be similar to the packets for timed media data. The FT field of the SMTP payload carrying the CEU metadata is set to 0x01. Each MFU data unit consists of one item of non-timed media. The FT field of the SMTP payload carrying the MFU is set to 0x02. Each item of the non-timed data is then used to build an MFU. Each MFU consists of an MFU header and the item's data.