**3GPP TSG- S4 Meeting #118e *S4-220471***

 **Electronic Meeting, 2022**

|  |
| --- |
| *CR-Form-v12.2* |
| **Pseudo CHANGE REQUEST** |
|  |
|  | **26.517** | **CR** |  | **rev** | **-** | **Current version:** | **1.0.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |
| --- |
|  |
| ***Title:***  | [5MBP3]: Stage 3 Proposal for Clause 6 (Object Distribution Method) |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | 5MBP3 |  | ***Date:*** | 31.3.2022 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** |  |
|  |  |
| ***Summary of change:*** | This pCR starts proposing Stage 3 text for the Object Distribution Method, based on existing text in TS 26.346 |
|  |  |
| ***Consequences if not approved:*** |  |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  |  |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  |  |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  |  |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

\*\*\*\* First Change \*\*\*\*

# 6 Object Distribution Method

Editor’s Note:

* Specify the stage 3 protocols for the MBS distribution methods (between MBSTF andMBS Client) based on existing MBMS delivery methods.
	+ Object distribution method, based on or reference to clause 7 of TS 26.346.
* Agreements per S4-220023
* Object delivery Method that includes:
	+ Download delivery method, File Delivery as defined in TS 26.346, clause 7.
	+ DASH/HLS over MBMS as defined in TS 26.346, clause 5.6 and 5.7.
* For the object delivery method, it is proposed to differentiate two different cases.
	+ Non-real-time file delivery including Carouselling
		- Selected properties of this mode include
			* Scheduled delivery
			* File repair
			* Carousel (for example supporting functionalities defined in DSM-CC)
			* Post-delivery reporting
			* File delivery QoS
			* Usage of FEC for file delivery
			* Support of single large file distribution
		- On stage-3 it is expected that we use FLUTE as defined in TS 26.346 with the following proposal:
			* Upgrade to the latest version of ALC, FLUTE and LCT
			* Keep a legacy version
			* Profile/remove any non-used functionalities based on MBMS Download Profile in TS 26.346, Annex L.4
	+ Object Streaming addressing DASH/HLS
		- Selected properties of this mode include
			* Timed delivery
			* Object deadline that is relevant for proper application operation.
			* Concurrent metrics reporting
			* Usage of FEC for object delivery
			* Sequence of multiple objects
			* Possibly multiple flows
			* Limited size
			* Partial objects
		- Enhancements are needed beyond the existing FLUTE.
			* Resolve and address object timing model (stage-3).

## 6.1 Introduction

The Object Distribution Method uses the FLUTE protocol (RFC 3926 [x1]) when delivering content over MBS Sessions.

FLUTE is built on top of the Asynchronous Layered Coding (ALC) protocol instantiation (RFC 3450 [x2]). ALC combines the Layered Coding Transport (LCT) building block [x3], a congestion control building block and the Forward Error Correction (FEC) building block ([x4]) to provide congestion controlled reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender. As mentioned in (RFC 3450 [x2]), congestion control is not appropriate in the type of environment that Object Distribution Method is provided, and thus congestion control is not used for Object Distribution. See figure 6.1-1 for an illustration of FLUTE building block structure. FLUTE is carried over UDP/IP, and is independent of the IP version and the underlying link layers used.



Figure 6.1-1: Building block structure of FLUTE

ALC uses the LCT building block to provide in-band session management functionality. The LCT building block has several specified and under-specified fields that are inherited and further specified by ALC. ALC uses the FEC building block to provide reliability. The FEC building block allows the choice of an appropriate FEC code to be used within ALC, including using the no-code FEC code that simply sends the original data using no FEC coding. ALC is under-specified and generally transports binary objects of finite or indeterminate length. FLUTE is a fully-specified protocol to transport files (any kind of discrete binary object), and uses special purpose objects - the File Description Table (FDT) Instances - to provide a running index of files and their essential reception parameters in-band of a FLUTE session.

## 6.2 Session description

### 6.2.1 Introduction

RFC 3926 [x] describes required and optional parameters for FLUTE session and media descriptors. This clause specifies SDP for FLUTE session that is used for the MBS Object Distribution and service announcement sessions. The formal specification of the parameters is given in ABNF ([x]).

### 6.2.2 SDP Parameters for Object Distribution session

#### 6.2.2.1 General

The semantics of a Session Description of an Object Distribution session includes the following parameters:

- The sender IP address.

- The number of channels in the session.

- The destination IP address and port number for each channel in the session per media.

- The Transport Session Identifier (TSI) of the session.

- The start time and end time of the session.

- The protocol ID (i.e. FLUTE/UDP).

- Media type(s) and fmt-list.

- Data rate using existing SDP bandwidth modifiers.

- MBS Service Type of MBS Session.

- FEC capabilities and related parameters.

- Service-language(s) per media.

- QoE Metrics (FFS)

- Alternative TMGI

This list includes the parameters required by FLUTE - RFC 3926 [x]

These shall be expressed in SDP ( [x] and [x]) syntax according to the following clauses.

#### 6.2.2.2 Sender IP address

There shall be exactly one IP sender address per MBS Object Distribution session, and thus there shall be exactly one IP source address per complete Object Distribution session SDP description. The IP source address shall be defined according to the source-filter attribute ("a=source-filter:") ([x] and [x]) for both IPv4 and IPv6 sources, with the following exceptions:

1. Exactly one source address may be specified by this attribute such that exclusive-mode shall not be used and inclusive-mode shall use exactly one source address in the <src-list>.

2. There shall be exactly one source-filter attribute per complete Object Distribution session SDP description, and this shall be in the session part of the session description (i.e. not per media).

3. The \* value shall be used for the <dest-address> subfield, even when the Object Distribution session employs only a single LCT (multicast) channel.

#### 6.2.2.3 Number of channels

Only one FLUTE channel is allowed per FLUTE session in the present document and thus there is no further need for a descriptor of the number of channels.

#### 6.2.2.4 Destination IP address and port number for channels

The FLUTE channel shall be described by the media-level channel descriptor. These channel parameters shall be per channel:

- IP destination address.

- Destination port number.

The IP destination address shall be defined according to the "connection data" field ("c=") of SDP ( [x]). The destination port number shall be defined according to the <port> sub-field of the media announcement field ("m=") of SDP ( [x]).

The presence of a FLUTE session on a certain channel shall be indicated by using the "*m-*line" in the SDP description as shown in the following example:

- m=application 12345 FLUTE/UDP 0

- c=IN IP6 FF1E:03AD::7F2E:172A:1E24/1

In the above SDP attributes, the *m*-line indicates the media used and the *c*-line indicates the corresponding channel. Thus, in the above example, the *m*-line indicates that the media is transported on a channel that uses FLUTE over UDP. Further, the *c*-line indicates the channel address, which, in this case, is an IPv6 address.

#### 6.2.2.5 Transport Session Identifier (TSI) of the session

The combination of the TSI and the IP source address identifies the FLUTE session. Each TSI shall uniquely identify a FLUTE session for a given IP source address during the time that the session is active, and also for a large time before and after the active session time (this is also an LCT requirement - RFC 3451 [x]).

The TSI shall be defined according to the SDP descriptor given below. There shall be exactly one occurrence of this descriptor in a complete FLUTE SDP session description and it shall appear at session level.

The syntax in ABNF is given below:

- flute-tsi-line = "a=flute-tsi:" tsi CRLF

- tsi = 1\*15DIGIT

#### 6.2.2.6 Multiple objects transport indication

RFC 3626 [x] requires the use of the Transport Object Identifier (TOI) header field (with one exception for packets with no payload when the A flag is used). The transport of a single FLUTE file requires that multiple TOIs are used (TOI 0 for FDT Instances). Thus, there is no further need to indicate to receivers that the session carries packets for more than one object and no SDP attribute (or other FLUTE out of band information) is needed for this.

#### 6.2.2.7 Session Timing Parameters

A MBS Object Distribution session start and end times shall be defined according to the SDP timing field ("t=") ( [x]).

#### 6.2.2.8 MBS service type of MBS Session

A new MBS service type declaration attribute is defined which results in, e.g.:

- a=mbs-servicetype:broadcast 123869108302929

- OR

- a=mbs-servicetype:multicast 123869108302929

The MBS service type declaration attribute shall be used in session descriptions using one or more MBS broadcast session media or multicast session media. If all media declarations use MBS broadcast session or multicast session, then the SDP attribute may be declared at session level. In that case the session level attribute applies to all media without a media level occurrence of the "mbs-servicetype" attribute.

- mbs-service-type-declaration-line = "a=mbs-servicetype:" ("broadcast"/"multicast" SP tmgi) CRLF

- tmgi = 1\*15DIGIT

Note: Please find below an example of the building of the TMGI:

UK MCC = 234 (MCC Digit 1 = 2; MCC Digit 2 = 3 and MCC Digit 3 = 4)

Vodafone UK MNC = 15

Therefore, with padding, Vodafone UK MNC = 15F (MNC Digit 1 = 1; MNC Digit 2 = 5 and MNC Digit 3 = F)

MBS Service ID = 70A886

Therefore, TMGI = 70A886 32F451 (Hex)

Therefore, TMGI = 123869108302929 (Decimal)

The Temporary Mobile Group Identity (tmgi) information element is defined in TS 24.008 [x] including the coding of the fields. Octets 3 to 8 (MBS Service ID, MCC and MNC) shall be placed in the tmgi attribute of the MBS service type declaration line, and are encoded as a decimal number. Octet 3 is the most significant octet. As this is encoded as a decimal number, leading zeros of the MBS Service ID field may be omitted.

If the MBS service type declaration attribute is applied at the session level, there shall be exacly one instance of MBS service type declaration attribute in the Session Description.

#### 6.2.2.9 FEC capabilities and related parameters

A new FEC-declaration attribute is defined which results in, e.g.:

- a=FEC-declaration:0 encoding-id=1

This attribute may be used on both session-level and media-level. Multiple instances are allowed to specify several different FEC declarations. The attribute is used on session level to define FEC declarations used by multiple media components. On media level it is used to define FEC declarations which are only valid for a single media component. If FEC declarations on both session and media level use the same reference number (fec-ref) then the media level declaration takes precedence for that media component. Each media component references one FEC declaration using the "a=FEC" attribute.

This attribute is optional to use for the Object Distribution Method as the information will be available elsewhere (e.g. FLUTE FDT Instances). If this attribute is not used, and no other FEC-OTI information is signalled to the MBS Client by other means, it may assume that support for FEC id 0 is sufficient capability to enter the session.

A new FEC-declaration reference attribute is defined which results in, e.g.:

- a=FEC:0

This is a media-level only attribute, used as a short hand to reference one of one or more FEC-declarations.

The syntax for the attributes in ABNF [23] is:

- fec-declaration-line = "a=FEC-declaration:" fec-ref SP fec-enc-id [";" SP fec-inst-id] CRLF

- fec-ref = 1\*3DIGIT ; value is the SDP-internal identifier for FEC-declaration.

- fec-enc-id = "encoding-id=" enc-id

- enc-id = 1\*DIGIT ; value is the FEC encoding ID used

- fec-inst-id = "instance-id=" inst-id

- inst-id = 1\*DIGIT ; value is the FEC Instance ID used.

- fec-line = "a=FEC:" fec-ref CRLF

#### 6.2.2.10 Service-language(s) per media

The existing SDP attribute "a=lang" is used to label the language of any language-specific media. The values are taken from [73] which in turn takes language and (optionally) country tags from ISO 639 [x] and ISO 3166 [x] (e.g. "a=lang:EN-US"). These are the same tags used in the User Service Description XML.

#### 6.2.2.11 Bandwidth Specification

The maximum bit-rate required by this FLUTE session shall be specified using the "AS" bandwidth modifier [x] on media level. The Application Specific (AS) bandwidth for a FLUTE session shall be the largest sum of the sizes of all packets transmitted during any one second long period of the session, expressed as kilobits. The size of the packet shall be the complete packet, i.e. IP, UDP and FLUTE headers, and the data payload.

#### 6.2.2.12 FEC Redundancy Level

The "FEC-redundancy-level" declaration attribute is defined in the form:

- a=FEC-redundancy-level:<fec-ref> <fec-redun-lev>,

This attribute is associated with the FEC-declaration attribute defined in sub-clause 6.2.2.9, with the same <fec-ref> field value. It may be used at the session or media level, and declares the redundant level of FEC protection, as a percentage, applied to the media component(s) carried on the associated MBS Object Distribution session. For example, a FEC redundancy level of 40% means that for an FEC-encoded block of *K* symbols, 1.4\**K* symbols are broadcast over the air. The applicability of the FEC redundancy level parameter, at the session or media level, mirrors the session- or media-level use of the corresponding FEC-declaration attribute with the same <fec-ref> value. The FEC-redundancy-level attribute is optional to use as a FEC declaration.

The syntax for this attribute, in ABNF [x], is as follows:

*- <*fec-ref*>*  is as defined in sub-clause 6.2.2.9,

*- <*fec-redun-lev*>* = "redundancy level=" <redun-lev>, and

- <redun-lev> *=* 1\*3DIGIT; represents the redundant amount of FEC protection applied to the object, expressed as an integer percentage value.

In the event that both the FDT extension attribute "FEC-Redundancy-Level" as defined in sub-clause 6.3.9, and the SDP FEC redundancy level indication are present, the declaration in the FDT shall take precedence from the MBS Client processing perspective.

#### 6.2.2.13 Alternative TMGI

An alternative tmgi declaration attribute is defined at the session level with the following ABNF [x] syntax:

- "a=alternative-tmgi:" tmgi-list CRLF

- tmgi-list = tmgi \*("," tmgi)

- tmgi = 1\*15DIGIT

The content(s) of an MBS User Service may be delivered simultaneously in multiple PLMN areas, over different MBS Session service instances (each identified by a unique TMGI). In this case, the alternative-tmgi attribute shall be present at the session level and lists all alternative values to the TMGI contained in the session-level MBS Service type declaration attribute, used for the broadcast of the FLUTE session data.

When this attribute is present, the MBS Client shall determine that the service is available at its current location, upon detecting a match between the TMGI derived from the PLMN-ID representing its current location, with one of the TMGIs from the following list:

- The set of TMGI values comprising the default TMGI in the MBS Service type declaration attribute and

- the TMGIs contained in the alternative-tmgi attribute.

Absence of a match shall be an indication to the MBS Client that the service not available at its current location.

The alternative tmgi declaration attribute is optional. It is not a replacement for the MBS Service type declaration attribute as defined in clause 6.2.2.8. In addition to the MBS Service type declaration attribute (which is the default TMGI), at most a single instance of the alternative tmgi declaration attribute shall be present in the Session Description. The same definition of the Temporary Mobile Group Identity (tmgi) as used in clause 6.2.2.8 shall be applied.

#### 6.2.2.14 Transport protocol identification

For the MBS Object Distribution Method, the <proto> field of the media descriptions ("m=") line of the SDP shall be set to 'FLUTE/UDP'.

#### 6.2.2.15 Media type and fmt-list

For the MBS Object Distribution Method, the media type and format list information shall be set in the "m=" line of the SDP as follows. The <media> field shall be set to 'application' and the <fmt> field shall be set to '0'.

### 6.2.3 SDP Examples for FLUTE Session

Here is a full example of SDP description describing a FLUTE session:

*v=0*

*o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24*

*s=Object Distribution session example*

*i=More information*

*t=2873397496 2873404696*

*a=mbs-servicetype:broadcast* 123869108302929

*a=FEC-declaration:0 encoding-id=1*

*a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9*

*a=flute-tsi:3*

*m=application 12345 FLUTE/UDP 0*

*c=IN IP6 FF1E:03AD::7F2E:172A:1E24/1*

*b=1000*

*a=lang:EN*

*a=FEC:0*

Below is a second example of an SDP description describing a FLUTE session and which indicates that 25% redundant FEC protection is applied to the FEC encoding of the video Segments of the associated DASH-formatted content:

*v=0*

*o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24*

*s=Object Distribution session carrying 2-hour DASH-encoded program*

*i=More information*

*t=3615124600 3615131800*

*a=mbs-servicetype:broadcast* 123869108302929

*a=FEC-declaration:0 encoding-id=1*

*a=FEC-redundancy-level:0 redundancy-level=25*

*a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9*

*a=flute-tsi:5*

*m=video 10111 FLUTE/UDP 0*

*c=IN IP6 FF1E:03AD::7F2E:172A:1E24/1*

*b=2048*

*a=lang:EN*

Below is a third example of an SDP description describing a FLUTE session with three TMGIs: one associated with the MBS Service type declaration attribute, and two others that are carried in the "alternative-tmgi" attribute:

*v=0*

*o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24*

*s=*Object Distribution *session carrying 2-hour DASH-encoded program*

*i=More information*

*t=3615124600 3615131800*

*a=mbs-servicetype:broadcast* 123869108302929

*a=FEC-declaration:0 encoding-id=1*

*a=FEC-redundancy-level:0 redundancy-level=25*

*a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9*

*a=flute-tsi:5*

*a=alternative-tmgi:*123869108302899,123869108302915

*m=video 10111 FLUTE/UDP 0*

*c=IN IP6 FF1E:03AD::7F2E:172A:1E24/1*

*b=2048*

*a=lang:EN*

## 6.3 FLUTE usage for Object Distribution

### 6.3.1 General

The purpose of Object Distribution is to deliver content like files or messages. In the context of Object Distribution, an object contains any type of data (e.g. 3GPP file (Audio/Video), Binary data, Still images, Text, Service Announcement metadata).

In the present document the term "object" is used for all objects carried by FLUTE (with the exception of the FDT Instances).

The interaction of Object Distribution and the caching directives is defined in sub-clause 6.3.12.

MBS Clients and servers (i.e. MBSTFs) supporting MBS Object Distribution Method shall implement the FLUTE specification (RFC 3926 [x]), as well as ALC (RFC 3450 [x]) and LCT (RFC 3451 [x]) features that FLUTE inherits. In addition, several optional and extended aspects of FLUTE, as described in the following clauses, shall be supported.

One FDT instance is typically bound to one MBS transmission session. It is therefore recommended, that each MBS transmission session should contain one or more repetitions of the same FDT instance.

### 6.3.2 Fragmentation of objects

Fragmentation of objects shall be provided by a blocking algorithm (which calculates source blocks from source objects) and a symbol encoding algorithm (which calculates encoding symbols from source blocks).

### 6.3.3 Symbol Encoding Algorithm

The "Compact No-Code FEC scheme" - [12] (FEC Encoding ID 0, also known as "Null-FEC") shall be supported.

The Raptor FEC scheme is described in sub-clause 6.3.11.

An MBS Client that supports MBS User Services shall support a decoder for the Raptor FEC scheme.

If an MBS Client that supports MBS User Services receives a mathematically sufficient set of encoding symbols generated according to the encoder specification in [x] for reconstruction of a source block then the decoder shall recover the entire source block. Note that the example decoder described in Raptor Forward Error Correction Scheme for Object Delivery (RFC 5350[x]) clause 5.5 fulfils this requirement.

### 6.3.4 Blocking Algorithm

In the case of the Compact No-Code FEC scheme [x] (FEC Encoding ID 0), then the "Algorithm for Computing Source Block Structure" described within the FLUTE specification (RFC 3926 [x]) shall be used.

In the case of Raptor forward error correction, then the algorithm defined in [x] shall be used.

The values of *N*, *Z*, *T* and *A* shall be set such that the sub-block size is less than 256KB.

### 6.3.5 Congestion Control

For simplicity of congestion control, FLUTE channelization shall be provided by a single FLUTE channel with single rate transport.

### 6.3.6 Content Encoding of Objects for Transport

Files may be content encoded for transport, as described in [x], in the MBS Object Distribution Method using the generic GZip algorithm as specified in RFC 1952 [x]. MBS Clients shall support GZip content decoding of FLUTE files (GZIP RFC 1952 [x], clause 9).

### 6.3.7 Signalling of Parameters with Basic ALC/FLUTE Headers

FLUTE and ALC mandatory header fields shall be as specified in [9, 10] with the following additional specializations:

- The length of the CCI (Congestion Control Identifier) field shall be 32 bits and it is assigned a value of zero (C=0).

- The Transmission Session Identifier (TSI) field shall be of length 16 bits (S=0, H=1, 16 bits).

- The Transport Object Identifier (TOI) field should be of length 16 bits (O=0, H=1).

- Only Transport Object Identifier (TOI) 0 (zero) shall be used for FDT Instances.

- The following features may be used for signalling the end of session and end of object transmission to the receiver:

- The Close Session flag (A) for indicating the end of a session.

- The Close Object flag (B) for indicating the end of an object.

In FLUTE the following applies:

- The Sender Current Time present flag (T) shall be set to zero.

- The Expected Residual Time present flag (R) shall be set to zero.

- The LCT header length (HDR\_LEN) shall be set to the total length of the LCT header in units of 32-bit words.

- For "Compact No-Code FEC scheme" [12], the FEC Payload ID shall be set according to RFC 3695 [13] such that a 16 bit SBN (Source Block Number) and then the 16 bit ESI (Encoding Symbol ID) are given.

- For "MBS FEC scheme", the FEC Payload ID shall be set according to Clause 6.3.11.

- For "EXT\_TIME" LCT Header [119], the sender may include it in all or some of the LCT packets for a object transmission. If EXT\_TIME is included, it shall contain the ERT time value set according to [119].

### 6.3.8 Signalling of Parameters with FLUTE Extension Headers

The FLUTE sender shall use FLUTE extension header fields EXT\_FDT, EXT\_FTI , EXT\_CENC [x] as follows:

- EXT\_FTI shall be included in every FLUTE packet carrying symbols belonging to any FDT Instance.

- FLUTE packets carrying symbols of files (not FDT Instances) shall not include an EXT\_FTI.

- FDT Instances shall not be content encoded and therefore EXT\_CENC shall not be used.

According to FLUTE [x] the following rules apply for a FLUTE sender:

- EXT\_FDT is in every FLUTE packet carrying symbols belonging to any FDT Instance.

- FLUTE packets carrying symbols of files (not FDT instances) do not include the EXT\_FDT.

Note: As an MBS Client conforms to a FLUTE receiver the receiver side treatment of LCT extension headers is covered by RFC3451 and RFC3926. The actions when receiving EXT\_FDT and EXT\_FTI are defined in RFC3926. The default action for unrecognized header extensions is to ignore them.

### 6.3.9 Signalling of Parameters with FDT Instances

The extended FLUTE FDT instance schema defined in Annex X (based on the one in RFC 3926 [x]) shall be used. In addition, the following applies to both the session level information and all files of a FLUTE session.

The inclusion of these FDT Instance data elements is mandatory according to the FLUTE specification:

- Content-Location (URI of a file).

- TOI (Transport Object Identifier of a object instance).

- Expires (expiry data for the FDT Instance).

For MBS operation, the MBS Client shall not use a received FDT Instance to interpret packets received beyond the expiration time of the FDT Instance.

Additionally, the inclusion of these FDT Instance data elements is mandatory. Note the following elements are optional in the FDT schema to stay aligned with the IETF RFC defined schema:

- Content-Length (source object length in bytes).

- Content-Type (content MIME type).

- FEC Encoding ID.

Other FEC Object Transmission Information specified by the FEC scheme in use:

NOTE 1: The FEC Object Transmission Information elements used are dependent on the FEC scheme, as indicated by the FEC Encoding ID.

- FEC-OTI-Maximum-Source-Block-Length.

- FEC-OTI-Encoding-Symbol-Length.

- FEC-OTI-Max-Number-of-Encoding-Symbols.

- FEC-OTI-Scheme-Specific-Info.

NOTE 2: RFC 3926 [x] describes which part or parts of an FDT Instance may be used to provide these data elements.

These optional FDT Instance data elements may or may not be included for FLUTE in MBS:

- Complete (the signalling that an FDT Instance provides a complete, and subsequently unmodifiable, set of object parameters for a FLUTE session may or may not be performed according to this method).

- Content-Encoding.

- Content-MD5: represents a digest of the transport object. The object server should indicate the MD5 hash value whenever multiple versions of the object are anticipated for the Object Distribution session.

- FEC-Redundancy-Level: indicate the FEC redundancy level for the object. For example, if the FEC-Redundancy-Level is set to 20, it means MBSTF will add 20% extra redunancy for this object during MBS Obbject Distribution.

- File-ETag: represents the value of the ETag, or entity-tag as defined in RFC 2616 [x] which mays also serve as the version identifier of the object described by the FDT Instance.

NOTE 3: The values for each of the above data elements are calculated or discovered by the FLUTE sender.

The FEC-OTI-Scheme-Specific-Info FDT Instance data element contains information specific to the FEC scheme indicated by the FEC Encoding ID encoded using base64.

### 6.3.11 FEC Scheme definition

This clause defines an FEC encoding scheme for the MBS forward error correction code defined in RFC 5350 [x] for the MBS Object Distribution Method. This scheme is identified by FEC Encoding ID 1. The FEC Payload ID format and FEC Object Transmission Information format are as defined in RFC 5350 [x], sub-clauses 3.1 and 3.2 respectively.

### 6.3.12 Caching Directives

A Object Distribution service may indicate the caching recommendations for a specific object or set of objects that are delivered using FLUTE.

The caching functionality defines three different caching directives:

**- no-cache**: this directive is used to indicate to the receiver not to cache a specific object(or set of objects). This is probably useful in the case where the object is expected to be highly dynamic (changes to the file occur quite often) or if the object will be used only once by the receiver application.

**- max-stale**: this directive indicates to the FLUTE receiver that a specific object (or set of objects) should be cached for an indefinite period of time, if possible. The object has no expiry date.

**- Expires**: this directive is used by the server to indicate the expected expiry time of a specific object (or set of objects). It indicates a date and time value expressed as the 32 most significant bits of the NTP [78] 64-bit timestamp format. These 32 bits provide an unsigned integer representing the time in seconds relative to 0 hours 1 January 1900.

The syntax of the caching directives is described in section 7.2.10.5.

## 6.4 Object Distribution Profile

The Object Distribution Profile primarily defines the required, expected and permitted usage of FLUTE FDT attributes and elements by the MBSTF, and the corresponding mandatory versus optional support for those FDT parameters by the MBS Client.

### 6.4.1 Common FDT-Instance and File Attributes

The following FDT attributes, defined at both the FDT-Instance and File levels, shall be carried in the FDT sent by the FLUTE sender, under either the *File-Instance* or *File* element, and shall be supported by the FLUTE receiver:

- Content-Type

- FEC-OTI-FEC-Encoding-ID

- FEC-OTI-Maximum-Source-Block-Length

- FEC-OTI-Encoding-Symbol-Length

- FEC-OTI-Scheme-Specific-Info

- FEC-OTI-Max-Number-of-Encoding-Symbols

The following FDT parameters, defined at both the FDT-Instance and File levels, shall not be used by the FLUTE sender, in either the *File-Instance* or *File* element:

- Content-Encoding attribute

- FEC-OTI-FEC-Instance-ID attribute

### 6.4.2 FDT-Instance specific Elements and Attributes

The following parameters, defined at theFDT-Instance level, shall be used by the FLUTE sender:

*-* Expires attribute

The following parameters, defined at theFDT-Instance level, shall not be used by the FLUTE sender:

*- Complete* attribute

### 6.4.3 FDT File specific Elements and Attributes

The following attributes, defined at the File level, shall be carried in the FDT sent by the FLUTE sender, and shall be supported by the FLUTE receiver, subject to the qualifications indicated below:

- Content-Location

- TOI

- Content-Length

The following attributes, defined at the File level, may be carried in the FDT sent by the FLUTE sender, and shall be supported by the FLUTE receiver, subject to the qualifications indicated below:

- Content-MD5

- File-ETag

- FEC-Redundancy-Level

The following element may be carried in the FDT sent by the FLUTE sender, and shall be supported by the FLUTE receiver:

- Cache-Control

- SchemaVersion

The following attributes shall only be carried in the in the File element of the FDT sent by the FLUTE sender, for the purpose of replacing or overriding corresponding attributes at the FDT-Instance level.

- Content-Type

- FEC-OTI-FEC-Encoding-ID

- FEC-OTI-Maximum-Source-Block-Length

- FEC-OTI-Encoding-Symbol-Length

- FEC-OTI-Scheme-Specific-Info

- FEC-OTI-Max-Number-of-Encoding-Symbols

The following attributes shall not be carried in the FDT sent by the FLUTE sender:

- Transfer-Length

### 6.4.4 SDP

The following attributes shall be presented the Object Distribution:

- "v=" indicates the version of the Session Description Protocol

- "s=" indicates the textual session name.

- "a=source-filter:" indicates the sender IP address

- "t=" indicates the MBS distribution session start time and stop time

- "a=mbs-servicetype: " indicates the MBS service type and used TMGI

- "a=flute-tsi:" indicates the Transport Session Identifier (TSI) of the MBS Session

- "m=application" indicates media type (application), transport protocol and the transport port for used the MBS Session

- "c=IN" indicates Destination IP address

The following attributes may be presented in the Object Distribution:

- "o=" indicates the originator of the session

- "a=lang" is used to label the language of any language-specific media.

- "a= FEC-declaration: " indicates the FEC encoding id

- "a= FEC-redundancy-level: " indicates the FEC dedundancy level

- "a= alternative-tmgi: " indicates the content(s) of an MBS User Service may be delivered simultaneously in multiple PLMN areas

## 6.4 Segment streaming profile

The segment streaming profile should be same as the Object Distribution except the following items

- Content-MD5 and File-ETag may be not presented

Editor’s Note: It should be clarified, what triggers the MBS Client to make an object or a partial object available.

## 6.5 Object repair

Editor’s Note: currently only HTTP/1.1 is included, support for HTTP/2 and HTTP/3 should be allowed, but are are FFS

The purpose of the Object Repair is to repair the lost or corrupted object blocks from the MBS data transmission. Once missing object data is identified, the MBS Client sends one or more messages to a object repair server requesting transmission of data that allows recovery of missing object data. All object repair requests and repair responses for a particular MBS transmission shall take place in a single TCP session using the HTTP protocol (RFC 2616 [x]). The repair request is routed to the object repair server IP address resolved from the selected object repair server URI.

The timing of the opening of the TCP connection to the server, and the first repair request, of a particular MBS Client is randomized over a time window defined in the service announcement. If there is more than one repair request to be made these are sent immediately after the first.

In object repair message format, the MBS Client uses the conventional HTTP/1.1 GET or partial GET requests as defined in RFC 2616 [x] to request all or a subset of source symbols of the referenced resource, respectively. The MBS Client shall support these message requests formats to allow the object repair requests to be serviced by a standard HTTP/1.1 server.

The MBS Client uses the HTTP GET request when it requires all the source symbols of the resource to be transmitted.

If the MBS Client only requests transmission of a subset of the source symbols or sub-symbols, it shall uses the HTTP partial GET request with the Range request header as defined in 14.35.2 of RFC 2616 [x]. The MBS Client shall indicate the specific source symbols or sub-symbols as a byte-range-spec as defined in 14.35.1 of RFC 2616 [x].

For messaging efficiency, the HTTP GET method allows the MBS Client to include multiple byte range requests within a single partial GET request. If the MBS Client includes multiple byte ranges in a single request the HTTP GET request should not exceed 2048 bytes in length to avoid truncation by the HTTP server.

If the MBS Client determines that it can select among multiple subsets of the source symbols or sub-symbols, then it should request the subset with the lowest ESI values, i.e., choose the missing source symbols or sub-symbols from the beginning of the source block or source sub-block, respectively. This improves the caching efficiency of the HTTP object repair servers.

If more than one object were distributed in a particular MBS Object Distribution session, and, if the MBS Client needs repair data for more than one object received in that session, the MBS Client shall send separate HTTP GET requests for each file.

If "File-ETag" is present in the FDT Instance, its value shall be used as the entity-tag in the "If-Match" or "If-Range" header of a conditional byte-range object request.

If "File-ETag" is not present in the FDT Instance, but "Content-MD5" is, the latter may be used as the entity-tag in the "If-Match" or "If-Range" header of a conditional byte-range object request, or the MBS Client may choose to send an HTTP GET request containing the "Range" header for the requested byte range(s), without the "If-Match" or "If-Range" header.

For the MBS Client, the nominal objective of using the "If-Match" header is to receive the requested range(s) of the object associated with the entity-tag, or no repair data if the request cannot be satisfied by the repair server. The nominal objective of using the "If-Range" header is to receive the latest version of the entire object in case the version associated with the entity-tag is no longer available on the repair server. To reduce the impact to capacity, the MBS Client should not use the "If-Range" header if it can request the range(s) from other repair servers.

If the "Content-Encoding" element is included in the FDT Instance for the object and is set to "gzip", then the MBS Client shall make the request to a modified URL, that is, the original object URL with the ".gz" extension added to the full path name but prior to the query part of the URL, if any. The MBS Client shall only use this request if a) the "File-ETag" attribute is present in the FDT Instance of that object, for use as the entity-tag in the request, or b) the "Content-MD5" attribute is present in the FDT Instance for that object, for use as the entity-tag in the request. Otherwise, the MBS Client should rather request the complete object instead of using byte range requests.

As an example, a FLUTE receiver partially receives the transport object with URL "<http://www.example.com/service1/document.pdf>",Content-Encoding set to "gzip", and with the Content-MD5 set to "B2B359591E961C6B0F468FE536BCD920=" while the "File-ETag" attribute is absent in the FDT Instance. It issues a repair request to the host server to fetch the missing bytes. The request is as follows:

**GET** /service1/document.pdf.gz  **HTTP/1.1
If-Match:** "B2B359591E961C6B0F468FE536BCD920="
**Range:** bytes=5018640-5042399

**Host:** [www.example.com](http://www.example.com)

The conditional request is used by the repair server to ensure that the byte range it will serve to the client is from the exact same compressed file. The conditional repair procedure is described earlier in this section.

In case the version identifier, indicated by the "Content-MD5" value as the entity-tag in the ‘If-Match’ header cannot be matched, the server will reply with a 412 "Precondition Failed" reply. Otherwise, the server will satisfy the request and reply with a 206 "Partial Content" if the request would be successful without the ‘If-Match’ header.

The following is an example of a response from the repair server:

HTTP/1.1 412 Precondition Failed
Content-Range:bytes=5018640-5042399,19037040-19050239
ETag: "B2B359591E961C6B0F468FE536BCD920="
Content-Length: 0

In this example, the version identifier of the file, represented by the value of the FDT Instance’s "File-ETag" and used as the entity-tag in the ‘If-Match’ header, matches the object version at the byte-range repair server. The server will send a 206 "Partial Content" response, providing the requested byte ranges in the payload:

HTTP/1.1 206 Partial Content

Date: Wed, 15 Nov 2015 06:25:24 GMT

ETag: "10690a1-4f2-40d45ae1"

Content-Length: 18001

Content-Type: multipart/byteranges; boundary=SEPARATION\_STRING

--SEPARATION\_STRING

Content-Type: application/pdf

Content-Range: bytes 5000-7999

...<*the first range>*...

-- SEPARATION\_STRING

Content-type: application/pdf

Content-range: bytes 25500-40500

*...<the second range>…*

-- SEPARATION\_STRING

The response message to the object repair follows the format and procedures in RFC 2616 [x] for responding to byte range requests.

When the HTTP message includes the content of a single byte range the repair server can provide the HTTP response with a "206 Partial content" status, include the Content-Range header, and use the content-range-spec to indicate the byte range of the repair data as specified in 14.16 of RFC 2616 [x].

When the repair server receives a request for multiple byte ranges it should attempt to transmit all the requested ranges in a single HTTP response. When an HTTP message includes multiple byte ranges, these are transmitted as a multipart message using the "multipart/byteranges" media type as defined in appendix 19.2 of RFC 2616 [x].

\*\*\*\* Next Change \*\*\*\*

# Annex X MBS Object Distribution session FDT Schema

## X.1 Extended FLUTE FDT Schema

Editor’s Note: A new schema is defined, removing some of the un-used features and making the FDT file smaller.

This specification defines two XML Schema elements necessary for the UE and the network side to maintain forward and backward compatibility: *schemaVersion* and *delimiter*.

In this version of the specification the network shall set the *schemaVersion* element, defined as a child of *FDT-Instance* element, to 1.

The schema *version* attribute (part of the schema instruction) shall be included in the UE schema and the network schema.

Note: The value of the *schemaVersion* element and *version* attribute is intended to be increased by 1 in every future releases where new element(s) or attribute(s) are added.

When an MBS Client receives an instantiation of an FDT compliant to this schema, it shall determine the schema version required to parse the instantiation as follows:

- If the MBS Client supports one or more versions of the FDT schema with the schema *version* attribute, then it shall use the schema that has the highest schema *version* attribute value that is equal to or less than the value in the received *schemaVersion* element;

- The *delimiter* element shall be set by the network to a value of 0, and the element content shall be ignored by the MBS Client.

The following is the FDT schema, and name as FLUTE-FDT-3GPP-Main.xsd

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema

 xmlns="urn:IETF:metadata:2022:FLUTE:FDT"

 xmlns:fl="urn:IETF:metadata:2022:FLUTE:FDT"

 xmlns:xs="http://www.w3.org/2001/XMLSchema"

 targetNamespace="urn:IETF:metadata:2022:FLUTE:FDT"

 elementFormDefault="qualified"

 version="1">

 <xs:element name="FDT-Instance" type="FDT-InstanceType"/>

 <xs:complexType name="FDT-InstanceType">

 <xs:sequence>

 <xs:element name="File" type="FileType" maxOccurs="unbounded"/>

 <xs:element ref="schemaVersion"/>

 <xs:element ref="delimiter"/>

 <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute name="Expires" type="xs:string" use="required"/>

 <xs:attribute name="Complete" type="xs:boolean" use="optional"/>

 <xs:attribute name="Content-Type" type="xs:string" use="optional"/>

 <xs:attribute name="Content-Encoding" type="xs:string" use="optional"/>

 <xs:attribute name="FEC-OTI-FEC-Encoding-ID" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-FEC-Instance-ID" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Maximum-Source-Block-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Encoding-Symbol-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Max-Number-of-Encoding-Symbols" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Scheme-Specific-Info" type="xs:base64Binary" use="optional"/>

 <xs:anyAttribute processContents="skip"/>

 </xs:complexType>

 <xs:complexType name="FileType">

 <xs:sequence>

 <xs:element ref="Cache-Control" minOccurs="0"/>

 <xs:element ref="delimiter"/>

 <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute name="Content-Location" type="xs:anyURI" use="required"/>

 <xs:attribute name="TOI" type="xs:positiveInteger" use="required"/>

 <xs:attribute name="Content-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="Transfer-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="Content-Type" type="xs:string" use="optional"/>

 <xs:attribute name="Content-Encoding" type="xs:string" use="optional"/>

 <xs:attribute name="Content-MD5" type="xs:base64Binary" use="optional"/>

 <xs:attribute name="FEC-OTI-FEC-Encoding-ID" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-FEC-Instance-ID" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Maximum-Source-Block-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Encoding-Symbol-Length" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Max-Number-of-Encoding-Symbols" type="xs:unsignedLong" use="optional"/>

 <xs:attribute name="FEC-OTI-Scheme-Specific-Info" type="xs:base64Binary" use="optional"/>

 <xs:attribute name="FEC-Redundancy-Level" type="xs:unsignedInt" use="optional"/>

 <xs:attribute name="File-ETag" type="xs:string" use="optional"/>

 <xs:anyAttribute processContents="skip"/>

 </xs:complexType>

 <xs:element name="Cache-Control">

 <xs:complexType>

 <xs:choice>

 <xs:element name="no-cache" type="xs:boolean" fixed="true"/>

 <xs:element name="max-stale" type="xs:boolean" fixed="true"/>

 <xs:element name="Expires" type="xs:unsignedInt"/>

 </xs:choice>

 <xs:anyAttribute processContents="skip"/>

 </xs:complexType>

 <xs:element name="schemaVersion" type="xs:unsignedInt"/>

 <xs:element name="delimiter" type="xs:byte"/>

</xs:schema>

## X.2 Example of FDT

[new example]

<?xml version="1.0" encoding="UTF-8"?>

<FDT-Instance

 xmlns="urn:IETF:metadata:2022:FLUTE:FDT"

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

 xsi:schemaLocation="urn:IETF:metadata:2022:FLUTE:FDT FLUTE-FDT-3GPP-Main.xsd"

 Expires="331129600">

 <File

 Content-Type="application/sdp"

 Content-Length="7543"

 TOI="2"

 FEC-OTI-FEC-Encoding-ID="1"

 FEC-OTI-Maximum-Source-Block-Length="8192"

 FEC-OTI-Encoding-Symbol-Length="16"

 FEC-OTI-Scheme-Specific-Info="AAECCA=="

 Content-Location=<http://www.example.com/fancy-session/main.sdp>>

 <Cache-Control>

 <Expires>331129630</Expires>

 </Cache-Control>

 <delimiter>0</delimiter>

 </File>

 <File

 Content-Type="String"

 Content-Length="161934"

 TOI="3"

 FEC-OTI-FEC-Encoding-ID="1"

 FEC-OTI-Maximum-Source-Block-Length="8192"

 FEC-OTI-Encoding-Symbol-Length="200"

 FEC-OTI-Scheme-Specific-Info="AAECCA=="

 Content-Location="http://www.example.com/fancy-session/trailer.3gp">

 <delimiter>0</delimiter>

 </File>

 <schemaVersion>1</schemaVersion>

 <delimiter>0</delimiter>

</FDT-Instance>

\*\*\*\* Last Change \*\*\*\*